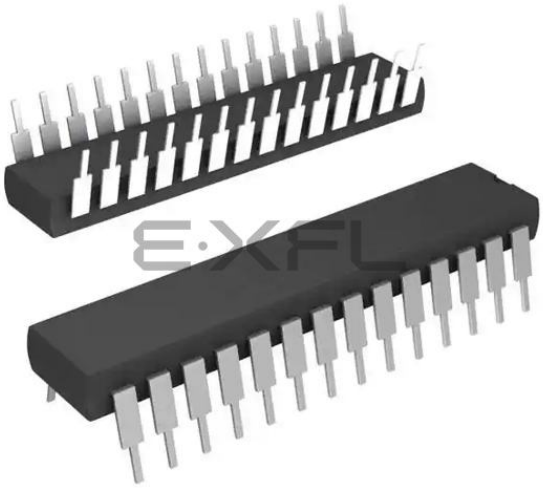
Infineon Technologies - CY8C24423-24PI Datasheet



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	M8C
Core Size	8-Bit
Speed	24MHz
Connectivity	I²C, SPI, UART/USART
Peripherals	LVD, POR, PWM, WDT
Number of I/O	24
Program Memory Size	4KB (4K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	256 x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 5.25V
Data Converters	A/D 2x14b; D/A 2x9b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Through Hole
Package / Case	28-DIP (0.300", 7.62mm)
Supplier Device Package	28-PDIP
Purchase URL	https://www.e-xfl.com/product-detail/infineon-technologies/cy8c24423-24pi

3. CPU Core (M8C)

This chapter explains the CPU Core, called M8C, and its associated registers. It covers the internal M8C registers, address spaces, instruction formats, and addressing modes. For additional information concerning the M8C instruction set, reference the *Assembly Language User Guide* available at the Cypress.com web site.

Table 3-1. M8C Registers

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
M8C Register										
x,F7h	CPU_F				XOI		Carry	Zero	GIE	RL : 00
Related Registers										
1,E0h	OSC_CR0	32k Select	PLL Mode	No Buzz	Sleep[1:0]		CPU Speed[2:0]			RW : 00
x,FF	CPU_SCR0	GIES		WDRS	PORS	Sleep			STOP	RW : 17

LEGEND
L: The AND, OR, and XOR flag instructions can be used to modify this register.
x: An "x" before the comma in the address field indicates that this register can be accessed or written to no matter what bank is used.

The M8C is a four MIPS 8-bit Harvard architecture microprocessor. Code selectable processor clock speeds from 93.7 kHz to 24 MHz allow the M8C to be tuned to a particular application's performance and power requirements. The M8C supports a rich instruction set which allows for efficient low-level language support.

3.1 Internal Registers

The M8C has five internal registers that are used in program execution. The following is a list of these registers.

- Accumulator (A)
- Index (X)
- Program Counter (PC) – internal use only
- Stack Pointer (SP)
- Flags (F)

All of the internal M8C registers are eight bits in width except for the PC which is 16 bits wide. Upon reset, A, X, PC, and SP are reset to `00h`. The Flag register (F) is reset to `02h`, indicating that the Z flag is set.

With each stack operation, the SP is automatically incremented or decremented so that it always points to the next stack byte in RAM. If the last byte in the stack is at address `FFh` the Stack Pointer will wrap to RAM address `00h`. It is the firmware developer's responsibility to ensure that the stack does not overlap with user-defined variables in RAM.

With the exception of the F register, the M8C internal registers are not accessible via an explicit register address. The internal M8C registers are accessed using instructions such as:

- `MOV A, expr`
- `MOV X, expr`
- `SWAP A, SP`
- `OR F, expr`
- `JMP LABEL`

The F register may be read by using address `F7h` in either register bank.

3.2 Address Spaces

The M8C has three address spaces: ROM, RAM, and registers. The ROM address space includes the supervisory ROM (SROM) and the Flash. The ROM address space is accessed via its own address and data bus. Figure 3-1 illustrates the arrangement of the PSoC microcontroller address spaces.

The ROM address space is composed of the Supervisory ROM and the on-chip Flash program store. Flash is organized into 64-byte blocks. The user need not be concerned with program store page boundaries, as the M8C automatically increments the 16-bit PC on every instruction making the block boundaries invisible to user code. Instructions occurring on a 256-byte Flash page boundary (with the

3.4 Instruction Format

The M8C has a total of seven instruction formats which use instruction lengths of one, two, and three bytes. All instruction bytes are fetched from the program memory (Flash) using an address and data bus that are independent from the address and data buses used for register and RAM access.

While examples of instructions will be given in this section, refer to the PSoC Designer Assembly Language User Guide for detailed information on individual instructions.

3.4.1 One-Byte Instructions

Many instructions, such as some of the `MOV` instructions, have single-byte forms because they do not use an address or data as an operand. As shown in Table 3-3, one-byte instructions use an 8-bit opcode. The set of one-byte instructions can be divided into four categories according to where their results are stored.

Table 3-3. One-Byte Instruction Format

Byte 0
8-bit opcode

The first category of one-byte instructions are those that do not update any registers or RAM. Only the one-byte `NOP` and `SSC` instructions fit this category. While the Program Counter is incremented as these instructions execute they do not cause any other internal M8C registers to be updated nor do these instructions directly affect the register space or the RAM address space. The `SSC` instruction will cause SROM code to run which will modify RAM and M8C internal registers.

The second category has only the two `PUSH` instructions in it. The `PUSH` instructions are unique because they are the only one-byte instructions that cause a RAM address to be modified. These instructions automatically increment the SP.

The third category has only the `HALT` instruction in it. The `HALT` instruction is unique because it is the only single-byte instruction that causes a user register to be modified. The `HALT` instruction modifies user register space address `FFh` (`CPU_SCR`).

The final category for single-byte instructions are those that cause internal M8C registers to be updated. This category holds the largest number of instructions: `ASL`, `ASR`, `CPL`, `DEC`, `INC`, `MOV`, `POP`, `RET`, `RETI`, `RLC`, `ROMX`, `RRC`, SWAP. These instructions can cause the A, X, and SP registers or SRAM to be updated.

3.4.2 Two-Byte Instructions

The majority of M8C instructions are two bytes in length. While these instructions can be divided into categories identical to the one-byte instructions this would not provide a useful distinction between the three two-byte instruction formats that the M8C uses.

Table 3-4. Two-Byte Instruction Formats

Byte 0	Byte 1
4-bit opcode	12-bit relative address
8-bit opcode	8-bit data
8-bit opcode	8-bit address

The first two-byte instruction format shown in Table 3-4 is used by short jumps and calls: `CALL`, `JMP`, `JACC`, `INDEX`, `JC`, `JNC`, `JNZ`, `JZ`. This instruction format uses only 4-bits for the instruction opcode leaving 12-bits to store the relative destination address in a two's-complement form.These instructions can change program execution to an address relative to the current address by -2048 or +2047.

The second two-byte instruction format (Table 3-4) is used by instructions that employ the Source Immediate addressing mode ("Source Immediate" on page 41). The destination for these instructions is an internal M8C register while the source is a constant value. An example of this type of instruction would be `ADD A, 7`.

The third two-byte instruction format is used by a wide range of instructions and addressing modes. The following is a list of the addressing modes that use this third two-byte instruction format:

- Source Direct (`ADD A, [7]`)
- Source Indexed (`ADD A, [X+7]`)
- Destination Direct (`ADD [7], A`)
- Destination Indexed (`ADD [X+7], A`)
- Source Indirect Post Increment (`MVI A, [7]`)
- Destination Indirect Post Increment (`MVI [7], A`)

For more information on addressing modes see "Addressing Modes" on page 41.

Regardless of the CPU speed bit's setting, if the actual CPU speed is greater than 12 MHz, the 24 MHz operating requirements apply. An example of this scenario is a device that is configured to use an external clock, which is supplying a frequency of 20 MHz. If the CPU speed register's value is 0b011, the CPU clock will be 20 MHz. Therefore, the supply voltage requirements for the device are the same as if the part was operating at 24 MHz off of the internal main oscillator. The operating voltage requirements are not relaxed until the CPU speed is at 12.0 MHz or less.

Table 10-4. OSC_CR0[2:0] Bits: CPU Speed

Bits	Internal Main Oscillator	External Clock
000b	3 MHz	EXTCLK/ 8
001b	6 MHz	EXTCLK/ 4
010b	12 MHz	EXTCLK/ 2
011b	24 MHz	EXTCLK/ 1
100b	1.5 MHz	EXTCLK/ 16
101b	750 kHz	EXTCLK/ 32
110b	187.5 kHz	EXTCLK/ 128
111b	93.7 kHz	EXTCLK/ 256

For additional information, reference the OSC_CR0 register on page 176.

10.2.2 ECO_TR Register

The External Crystal Oscillator Trim register (ECO_TR) sets the adjustment for the External Crystal Oscillator. The device specific value placed in this register at boot time is based on factory testing. This register does not adjust the frequency of the External Crystal Oscillator. It is recommended that the user not alter the bits in this register.

Bits 7 and 6: PSSDC[1:0]. These bits are used to set the sleep duty cycle.

Bits 5 to 0: Reserved.

For additional information, reference the ECO_TR register on page 184.

10.2.3 CPU_SCR1 Register

The CPU_SCR1 register is used to convey status and control of events related to internal resets and watchdog reset.

Bits 7 to 1: Reserved.

Bit 0: IRAMDIS. The Initialize RAM Disable bit is a control bit that is readable and writeable. The default value for this bit is 0, which indicates that the maximum amount of SRAM should be initialized on reset to a value of 00h. When the bit is set, the minimum amount of SRAM is initialized after a watchdog reset. For more information on this bit, see the "SROM Function Descriptions" on page 48.

For additional information, reference the CPU_SCR1 register on page 153.

13.1.7 DxBxxDR2

Digital Basic/Communication Type B Block Data Register 2

Individual Register Names and Addresses

DBB00DR2 : 0,22h	DBB01DR2 : 0,26h	DCB02DR2 : 0,2Ah	DCB03DR2 : 0,2Eh

	7	6	5	4	3	2	1	0
Access : POR	RW : 00							
Bit Name	Data[7:0]							

The function of this register is dependant on the function its block has been configured for (selected in the FN[2:0] bits of the DxBxxFN register on page 159. For additional information, reference the “Register Definitions” on page 208 in the Digital Blocks chapter.

Bit	Name	Description
[7:0]	**Data[7:0]**	Data for selected function.

Block Function	**Register Function**
Timer	Capture/Compare
Counter	Compare
Dead Band	Buffer
CRCPRS	Seed/Residue
SPIM	RX Buffer
SPIS	RX Buffer
TXUART	Not applicable
RXUART	RX Buffer

13.1.12 DCBxxCR0

(SPIM Control)

Digital Communication Type B Block Control Register 0

Individual Register Names and Addresses

DCB02CR0: 0,2Bh DCB03CR0: 0,2Fh

	7	6	5	4	3	2	1	0
Access : POR	RW : 0	R : 0	R : 0	R : 1	R : 0	RW : 0	RW : 0	RW : 0
Bit Name	LSB First	Overrun	SPI Complete	TX Reg Empty	RX Reg Full	Clock Phase	Clock Polarity	Enable

The LSB First, Clock Phase, and Clock Polarity bit are configuration bits and should never be changed once the block is enabled. They can be set at the same time that the block is enabled. For additional information, reference the “Register Definitions” on page 208 in the Digital Blocks chapter.

Bit	Name	Description
[7]	LSB First	This bit should not be changed during an SPI transfer. 0 Data is shifted out MSB first. 1 Data is shifted out LSB first.
[6]	Overrun	0 No overrun has occurred. 1 Overrun has occurred. Indicates that a new byte has been received and loaded into the RX Buffer before the previous one could be read. Cleared on read of this (CR0) register.
[5]	SPI Complete	0 Indicates that a byte may still be in the process of shifting out, or no transmission is active. 1 Indicates that a byte has been shifted out and all associated clocks have been generated. Cleared on read of this (CR0) register. Optional interrupt.
[4]	TX Reg Empty	The reset state and the state when the block is disabled is ‘1’. 0 Indicates that a byte is currently buffered in the TX register. 1 Indicates that a byte can be written to the TX register. Cleared on write of the TX Buffer (DR1) register. Default interrupt. This status will initially be asserted on block enable; however, the TX Reg Empty interrupt will occur only after the first data byte is written and transferred into the shifter.
[3]	RX Reg Full	0 RX register is empty. 1 A byte has been received and loaded into the RX register. Cleared on read of the RX Buffer (DR2) register.
[2]	Clock Phase	0 Data is latched on the leading edge of the clock. Data changes on the trailing edge (Modes 0,1). 1 Data changes on the leading edge of the clock. Data is latched on the trailing edge (Modes 2,3).
[1]	Clock Polarity	0 Non-inverted, clock idles low (Modes 0,2). 1 Inverted, clock idles high (Modes 1,3).
[0]	Enable	0 SPI Master is not enabled. 1 SPI Master is enabled.

13.1.22 ACBxxCR0

Analog Continuous Time Type B Block Control Register 0

Individual Register Names and Addresses

ACB00CR0 : x,71h ACB01CR0 : x,75h

	7	6	5	4	3	2	1	0
Access : POR	RW : 0				RW : 0	RW : 0	RW : 0	
Bit Name	RTapMux[3:0]				Gain	RTopMux	RBotMux[1:0]	

For additional information, reference the “Register Definitions” on page 257 in the Continuous Time Block chapter.

Bit	Name	Description
[7:4]	**RTapMux[3:0]**	Encoding for selecting one of 18 resistor taps. The four bits of RTapMux[3:0] allow selection of 16 taps. The two additional tap selections are provided using ACBxxCR3 bit 0, EXGAIN. The EXGAIN bit only affects the RTapMux values 0h and 1h.

RTap	EXGAIN	Rf	Ri	Loss	Gain
0h	1	47	1	0.0208	48.000
1h	1	46	2	0.0417	24.000
0h	0	45	3	0.0625	16.000
1h	0	42	6	0.1250	8.000
2h	0	39	9	0.1875	5.333
3h	0	36	12	0.2500	4.000
4h	0	33	15	0.3125	3.200
5h	0	30	18	0.3750	2.667
6h	0	27	21	0.4375	2.286
7h	0	24	24	0.5000	2.000
8h	0	21	27	0.5625	1.778
9h	0	18	30	0.6250	1.600
Ah	0	15	33	0.6875	1.455
Bh	0	12	36	0.7500	1.333
Ch	0	9	39	0.8125	1.231
Dh	0	6	42	0.8750	1.143
Eh	0	3	45	0.9375	1.067
Fh	0	0	48	1.0000	1.000

[3] **Gain**

Select gain or loss configuration for output tap.

0	Loss
1	Gain

[2] **RTopMux**

Encoding for feedback resistor select.

0	Rtop to Vdd
1	Rtop to opamp’s output

[1:0] **RbotMux[1:0]**

Encoding for feedback resistor select. Bits [1:0] are overridden if bit 1 of ACBxxCR3 is set. In that case, the bottom of the resistor string is connected cross columns. Note that available mux inputs vary by individual PSoC block.

	ACB00	ACB01
00b	ACB01	ACB00
01b	AGND	AGND
10b	Vss	Vss
11b	ASC10	ASD11

13.1.66 CPU_SCR0

System Status and Control Register 0

Individual Register Names and Addresses

CPU_SCR0: x,FFh

	7	6	5	4	3	2	1	0
Access : POR	R : 0		RC : 0	RC : 1	RW : 0			RW : 0
Bit Name	GIES		WDRS	PORS	Sleep			STOP

For additional information, reference the “Register Definitions” on page 77 in the Sleep and Watchdog chapter.

Bit	Name	Description
[7]	**GIES**	Global interrupt enable status. It is recommended that the user read the global interrupt enable Flag bit from the CPU_F register on page 152. This bit is Read Only for GIES. Its use is discouraged, as the Flag register is now readable at address x,F7h (read only).
[6]	**Reserved**	
[5]	**WDRS**	Watchdog Reset Status. This bit may not be set by user code; however, it may be cleared by writing it with a zero (0). 0 No Watchdog Reset has occurred. 1 Watchdog Reset has occurred.
[4]	**PORS**	Power On Reset Status. This bit may not be set by user code; however, it may be cleared by writing it with a zero (0). 0 Power On Reset has not occurred and Watchdog Timer is enabled. 1 Will be set after external reset or Power On Reset.
[3]	**Sleep**	Set by the user to enable the CPU sleep state. CPU will remain in Sleep mode until any interrupt is pending. 0 Normal operation 1 Sleep
[2:1]	**Reserved**	
[0]	**STOP**	0 M8C is free to execute code. 1 M8C is halted. Can only be cleared by POR, XRES, or WDR.

17.1.3 Output De-Multiplexers

Most functions have two outputs: a primary and an auxiliary output. Each of these outputs may be driven onto the row output bus. Each de-multiplexer is implemented with four tri-state drivers. There are two bits to select one of the four and an additional bit to enable the selected driver.

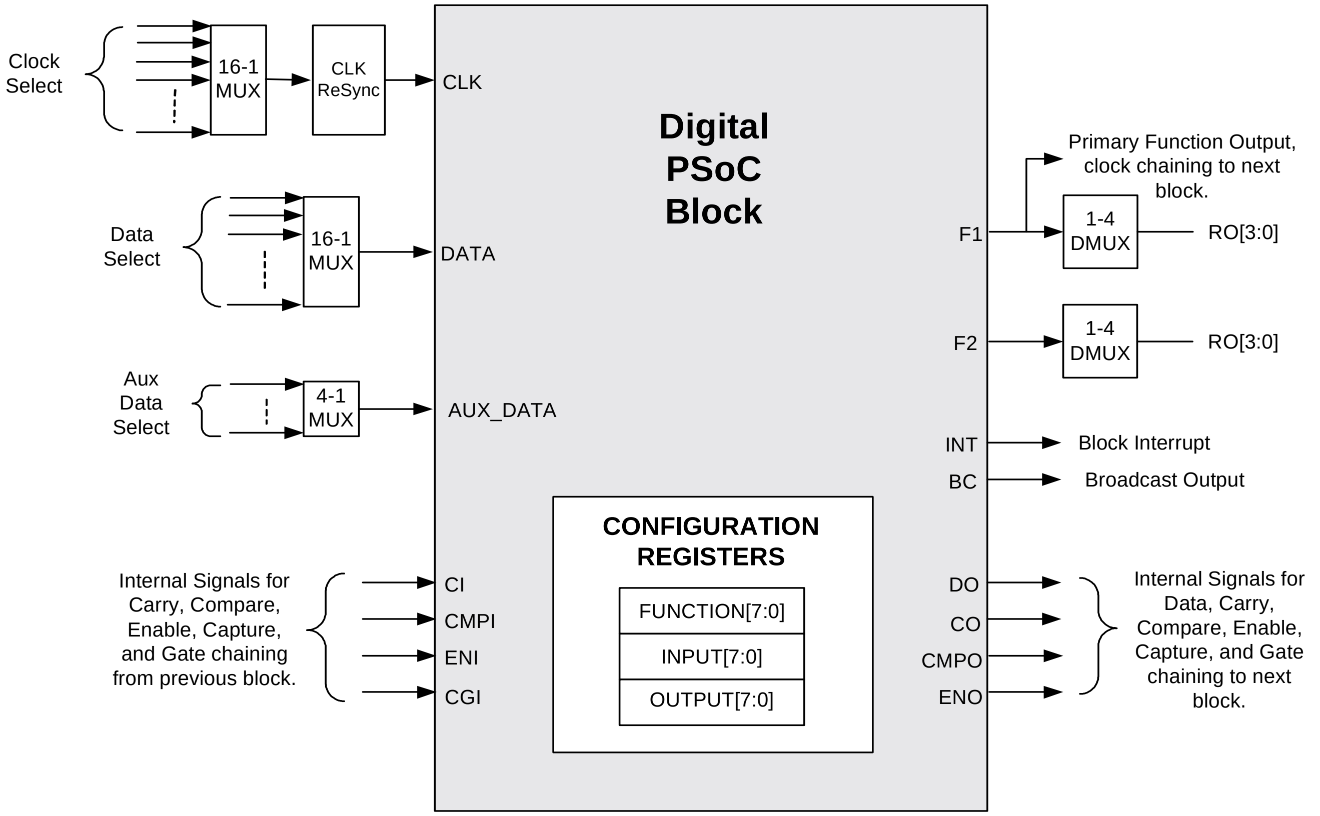


Figure 17-2. Digital Blocks Top-Level Block Diagram

17.1.4 Block Chaining Signals

Each digital block has the capability to be chained and to create functions with bit widths greater than eight. There are signals to propagate information, such as Compare, Carry, Enable, Capture and Gate, from one block to the next to implement higher precision functions. The selection made in the Function register determines which signals are appropriate for the desired function. User Modules that have been designed to implement digital functions, with greater than 8-bit width, will automatically make the proper selections of the chaining signals, to ensure the correct information flow between blocks.

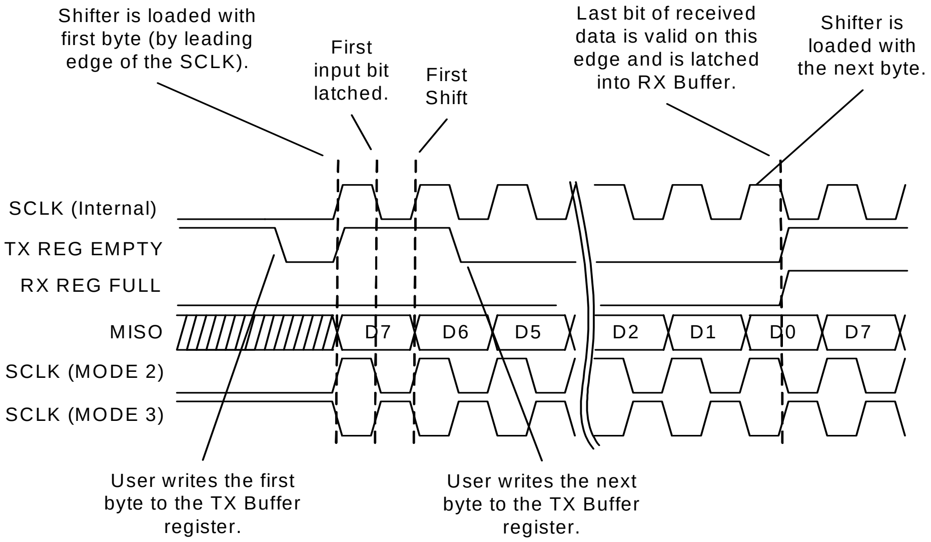

Shifter is loaded with first byte (by leading edge of the SCLK).
First input bit latched.
First Shift
Last bit of received data is valid on this edge and is latched into RX Buffer.
Shifter is loaded with the next byte.
SCLK (Internal)
TX REG EMPTY
RX REG FULL
MISO
SCLK (MODE 2)
SCLK (MODE 3)
D7
D6
D5
D2
D1
D0
D7
User writes the first byte to the TX Buffer register.
User writes the next byte to the TX Buffer register.

Figure 17-20. Typical SPIS Timing in Modes 2 and 3

Slave Select (SS_, active low). Slave Select must be asserted to enable the SPIS for receive and transmit. There are two ways to do this:

1. Drive the auxiliary input from a pin (selected by the Aux IO Select bits in the Output register). This gives the SPI master control of the slave selection in a multi-slave environment.
2. SS_ may be controlled in firmware with register writes to the Output register. When Aux IO Enable = 1, Aux IO Select bit 0 becomes the SS_ input. This allows the user to save an input pin in single slave environments.

When SS_ is negated (whether from an external or internal source), the SPIS state machine is reset, and the MISO output is forced to idle at logic '1'. In addition, the SPIS will ignore any incoming MOSI/SCLK input from the master.

Status Generation and Interrupts. There are four status bits in the SPIS Block: TX Reg Empty, RX Reg Full, SPI Complete, and Overrun. The timing of these status bits are identical to the SPIM, with the exception of TX Reg Empty which is covered in the section on TX data queuing.

Status Clear On Read. Refer to the same subsection in "SPIM Timing" on page 219.

TX Data Queuing. Most SPI applications call for data to be sent back from the slave to the master. Writing firmware to accomplish this requires an understanding of how the shift register is loaded from the TX Buffer register.

All modes use the following mechanism: 1) If there is no transfer in progress, 2) if the shifter is empty, and 3) if data is available in the TX Buffer register, the byte is loaded into the shifter.

The only difference between the modes is that the definition of "transfer in progress" is slightly different between modes 0 and 1 and modes 2 and 3.

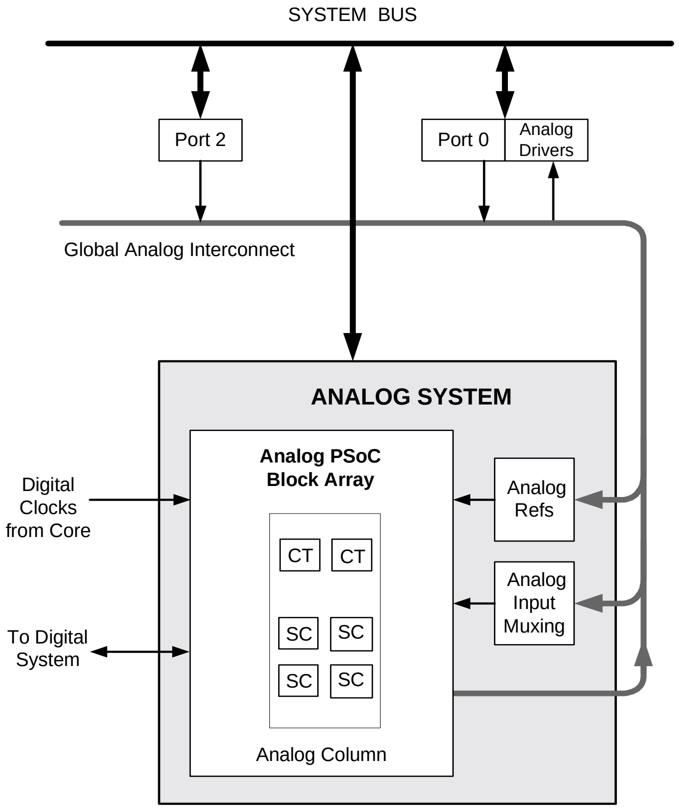
Figure 17-21 illustrates TX data loading in modes 0 and 1. A transfer in progress is defined to be from the falling edge of SS_ to the point at which the RX Buffer register is loaded with the received byte. This means that in order to send a byte in the next transfer, it must be loaded into the TX Buffer register before the falling edge of SS_. This ensures a minimum setup time for the first bit since the leading edge of the first SCLK must latch in the received data. If SS_ is not toggled between each byte or is forced low through the configuration register, the leading edge of SCLK is used to define the start of transfer. However, in this case, the user must provide the required setup time (one-half clock minimum before the leading edge), with a knowledge of system latencies and response times.

SECTION E ANALOG SYSTEM

The Analog System section discusses the analog components of the PSoC device and the registers associated with those components. This section encompasses the following chapters:

Top-Level Analog Architecture

The figure below displays the top-level architecture of the PSoC's analog system. With the exception of Analog Drivers, each component of the figure is discussed at length in this section. Analog drivers are discussed in detail in the Analog Output Drivers chapter on page 63.



PSoC Analog System Block Diagram

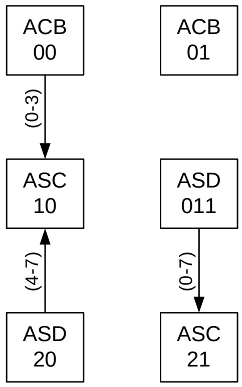


Figure 19-6. CMux Connections

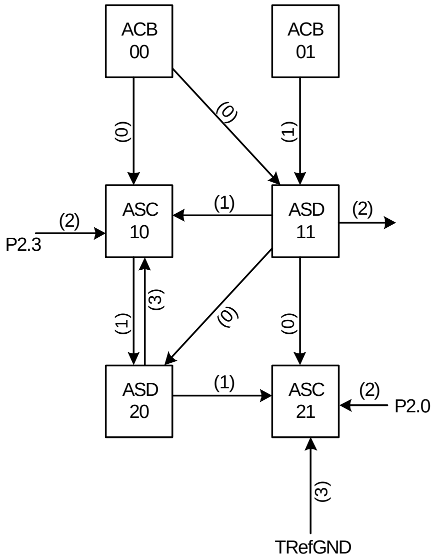


Figure 19-7. BMuxSC/SD Connections

19.1.1 Analog Comparator Bus

Each analog column has a dedicated comparator bus associated with it. Every analog PSoC block has a comparator output that can drive out on this bus. However, the comparator output from only one analog block in a column can be actively driving the comparator bus for that column at any one time. The output on the comparator bus can drive into the digital blocks and is also available to be read in the CMP_CR register.

The comparator bus is latched before it is available to either drive the digital blocks or be read in the Analog Comparator Control Register. The latch for each comparator bus is transparent (the output tracks the input), during the high period of PHI2. During the low period of PHI2, the latch retains the value on the comparator bus during the high to low transition of PHI2.

The output from the analog block that is actively driving the bus may also be latched internally to the analog block itself.

In the Continuous Time (CT) analog blocks, the CPhase and CLatch bits inside the Analog Continuous Time Type B Block xx Control Register 2 determine whether the output signal on the comparator bus is latched inside the block, and if it is, which clock phase it is latched on.

In the SC analog blocks, the output on the comparator bus is always latched. The ClockPhase bit in the Analog Switch-Cap Type B Block xx Control Register 0 or the Analog SwitchCap Type B Block xx Control Register 0 determines the phase on which this data is latched and available.

19.2 Temperature Sensing Capability

A temperature-sensitive voltage, derived from the bandgap sensing on the die, is buffered and available as an analog input into the Analog Switch Cap Type C Block ASC21. Temperature sensing allows protection of device operating ranges for fail-safe applications. Temperature sensing, combined with a long sleep timer interval (to allow the die to approximate ambient temperature), can give an approximate ambient temperature for data acquisition and battery charging applications. The user may also calibrate the internal temperature rise based on a known current consumption.

The temperature sensor input to the ASC21 block is labeled VTemp and its associated ground reference is labeled TRef-GND.

Analog Switch Cap Type C PSoC Block Control Registers

22.3.1 ASCxxCR0 Register

Bit 7: FCap. This bit controls the size of the switched feedback capacitor in the integrator.

Bit 6: ClockPhase. This bit controls the internal clock phasing relative to the input clock phasing. ClockPhase affects the output of the analog column bus, which is controlled by the AnalogBus bit in Control 2 Register (ASC10CR2, ASC21CR2).

Bit[6] is the ClockPhase select that inverts the clock internal to the blocks. During normal operation of an SC block for the amplifier of a column enabled to drive the output bus, the connection is only made for the last half of PHI2 (during PHI1 and for the first half of PHI2, the output bus floats at the last voltage to which it was driven). This forms a sample and hold operation using the output bus and its associated capacitance. This design prevents the output bus from being perturbed by the intermediate states of the SC operation (often a reset state for PHI1 and settling to the valid state during PHI2).

Following are the exceptions: 1) If the ClockPhase bit in CR0 (for the SC block in question) is set to 1, then the output is enabled for the whole of PHI2. 2) If the SHDIS signal is set in bit 6 of the Analog Clock Source Control Register, then sample and hold operation is disabled for all columns and all enabled outputs of SC blocks are connected to their respective output busses for the entire period of their respective PHI2s.

This bit also affects the latching of the comparator output (CBUS). Both clock phases, PHI1 and PHI2, are involved in the output latching mechanism. The capture of the next value to be output from the latch (capture point event) happens during the falling edge of one clock phase, and the rising edge of the other clock phase will cause the value to come out (output point event). This bit determines which clock phase triggers the capture point event, and the other clock will trigger the output point event. The value output to the comparator bus will remain stable between output point events.

Bit 5: ASign. This bit controls the switch phasing of the switches on the bottom plate of the ACap capacitor. The bottom plate samples the input or the reference.

Bits 4 to 0: ACap[4:0]. The ACap bits set the value of the capacitor in the A path.

For additional information, reference the ASCxxCR0 register on page 112.

22.3.2 ASCxxCR1 Register

Bits 7, 6, and 5: ACMUX[2:0]. ACMux controls the input muxing for both the A and C capacitor branches. The high order bit, ACMux[2], selects one of two inputs for the C branch.

Bits 4 to 0: BCap[4:0]. The BCap bits set the value of the capacitor in the B path.

For additional information, reference the ASCxxCR1 register on page 113.

22.3.3 ASCxxCR2 Register

Bit 7: AnalogBus. This bit gates the output to the analog column bus. The output on the analog column bus is affected by the state of the ClockPhase bit in Control 0 Register (ASC10CR0, ASC21CR0). If AnalogBus is set to 0, the output to the analog column bus is tri-stated. If AnalogBus is set to 1, the signal that is output to the analog column bus is selected by the ClockPhase bit. If the ClockPhase bit is 0, the block output is gated by sampling clock on last part of PHI2. If the ClockPhase bit is 1, the block output continuously drives the analog column bus (ABUS).

Bit 6: CompBus. This bit controls the output to the column comparator bus (CBUS). Note that if the comparator bus is not driven by anything in the column, it is pulled low. The comparator output is evaluated on the rising edge of internal PHI1 and is latched so it is available during internal PHI2.

Bit 5: AutoZero. This bit controls the shorting of the output to the inverting input of the opamp. When shorted, the opamp is basically a follower. The output is the opamp offset. By using the feedback capacitor of the integrator, the block can memorize the offset and create an offset cancellation scheme. AutoZero also controls a pair of switches between the A and B branches and the summing node of the opamp. If AutoZero is enabled, then the pair of switches is active. AutoZero also affects the function of the FSW1 bit in Control 3 Register.

Bits 4 to 0: CCap[4:0]. The CCap bits set the value of the capacitor in the C path.

For additional information, reference the ASCxxCR2 register on page 114.

Bit 0: EXGAIN. The continuous time block's resistor tap is specified by the value of ACBxxCR3 EXGAIN combined with the value of ACBxxCR0 RtapMux[3:0]. For RtapMux values from 02h through 15h, the EXGAIN bit has no effect on which tap is selected. See THE ACBxxCR0 register details. The EXGAIN bit allows one additional tap selection for RtapMux = 01h and a second additional tap selection for RtapMux = 00h (see Figure 23-4).

For additional information, reference the ACBxxCR3 register on page 108.

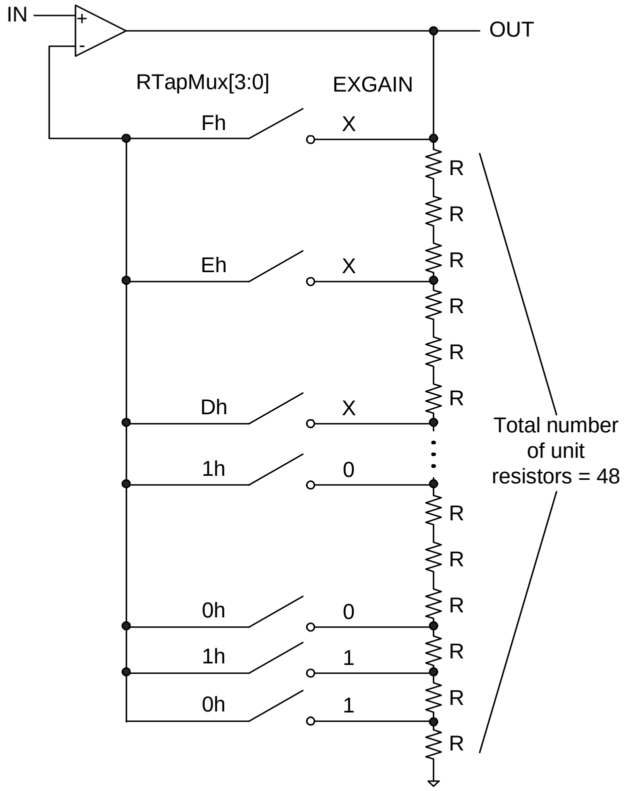


Figure 23-4. Continuous Time Block in Gain Configuration

System Resources Register Summary

The table below lists all the PSoC registers that the system resources of the device and its individual blocks use.

Summary Table of the System Resource Registers

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
DIGITAL CLOCK REGISTERS										
0,DAh	INT_CLR0	VC3	Sleep	GPIO			Analog 1	Analog 0	V Monitor	RW : 00
0,E0h	INT_MSK0	VC3	Sleep	GPIO			Analog 1	Analog 0	V Monitor	RW : 00
1,DEh	OSC_CR4							VC3 Input Select[1:0]		RW : 00
1,DFh	OSC_CR3	VC3 Divider[7:0]								RW : 00
1,E0h	OSC_CR0	32k Select	PLL Mode	No Buzz	Sleep[1:0]		CPU Speed[2:0]			RW : 00
1,E1h	OSC_CR1	VC1 Divider[3:0]				VC2 Divider[3:0]				RW : 00
1,E2h	OSC_CR2	PLLGAIN					EXTCLKEN	IMODIS	SYSCLKX2 DIS	RW : 00
MULTIPLY ACCUMULATE (MAC) REGISTERS										
0,E8h	MUL_X	Data[7:0]								W : XX
0,E9h	MUL_Y	Data[7:0]								W : XX
0,EAh	MUL_DH	Data[7:0]								R : XX
0,EBh	MUL_DL	Data[7:0]								R : XX
0,ECh	MAC_X/ ACC_DR1	Data[7:0]								RW : 00
0,EDh	MAC_Y/ ACC_DR0	Data[7:0]								RW : 00
0,EEh	MAC_CL0/ ACC_DR3	Data[7:0]								RW : 00
0,EFh	MAC_CL1/ ACC_DR2	Data[7:0]								RW : 00
DECIMATOR REGISTERS										
0,E4h	DEC_DH	Data High Byte[7:0]								RC : XX
0,E5h	DEC_DL	Data Low Byte[7:0]								RC : XX
0,E6h	DEC_CR0	IGEN[3:0]				ICLKS0	DCOL[1:0]		DCLKS0	RW : 00
0,E7h	DEC_CR1	ECNT	IDEC			ICLKS1			DCLKS1	RW : 00
I2C REGISTERS										
0,D6h	I2C_CFG		PSelect	Bus Error IE	Stop IE	Clock Rate		Enable Master	Enable Slave	RW : 00
0,D7h	I2C_SCR	Bus Error	Lost Arb	Stop Status	ACK	Address	Transmit	LRB	Byte Complete	R : 00
0,D8h	I2C_DR	Data[7:0]								RW : 00
0,D9h	I2C_MSCR					Bus Busy	Master Mode	Restart Gen	Start Gen	R : 00
POR AND LVD REGISTERS										
1,E3h	VLT_CR	SMP		PORLEV[1:0]		LVDTBEN	VM[2:0]			RW : 00
1,E4h	VLT_CMP						PUMP	LVD	PPOR	R : 00
INTERNAL VOLTAGE REFERENCE REGISTER										
1,EAh	BDG_TR		AGNDBYP	TC[1:0]		V[3:0]				RW:00
SWITCH MODE PUMP (SMP) REGISTER										
1,E3h	VLT_CR	SMP		PORLEV[1:0]		LVDTBEN	VM[2:0]			RW : 00
SYSTEM RESET REGISTERS										
0,FEh	CPU_SCR1								IRAMDIS	RW : 00
0,FFh	CPU_SCR0	GIES		WDRS	PORS	Sleep			STOP	RW : XX

LEGEND
C: Cearable register or bits.
X: The value for power on reset is unknown.

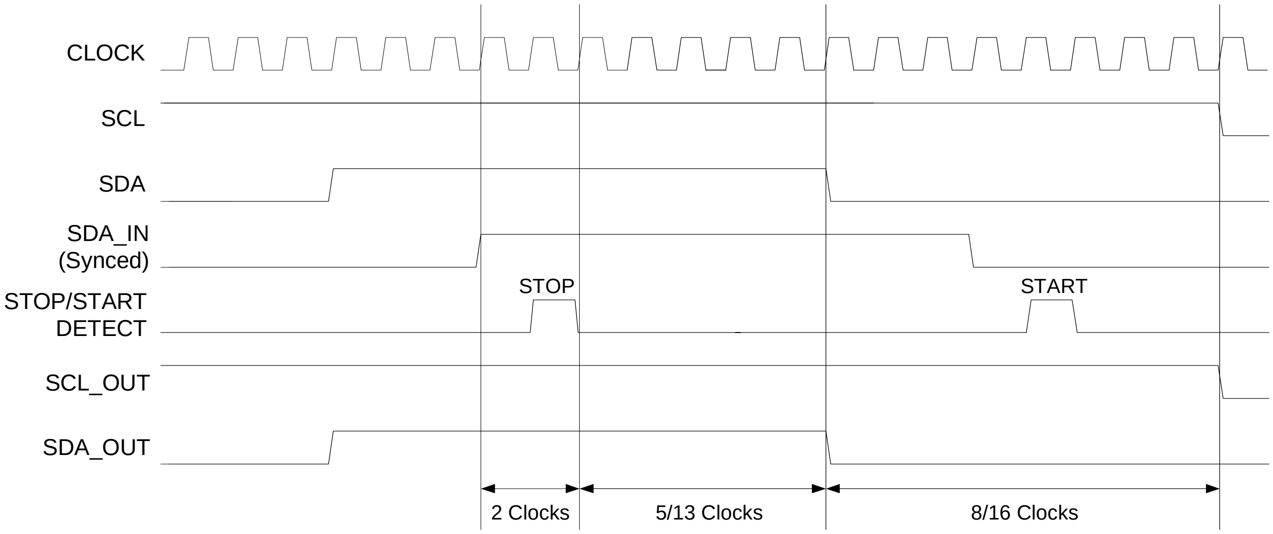


Figure 27-13. Master Stop/Start Chaining

27.4.6 Master Restart Timing

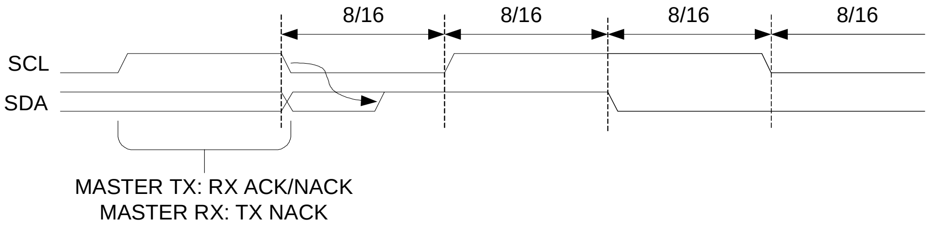


Figure 27-14. Master Restart Timing

27.4.7 Master Stop Timing

Figure 27-15 shows basic Master Stop timing. In order to generate a Stop, the SDA line is first pulled low, in accordance with the basic SDA output timing. Then, after the full low of SCL is completed and the SCL line is pulled high, the SDA line remains low for a full one-half bit time before it is pulled high to signal the Stop.

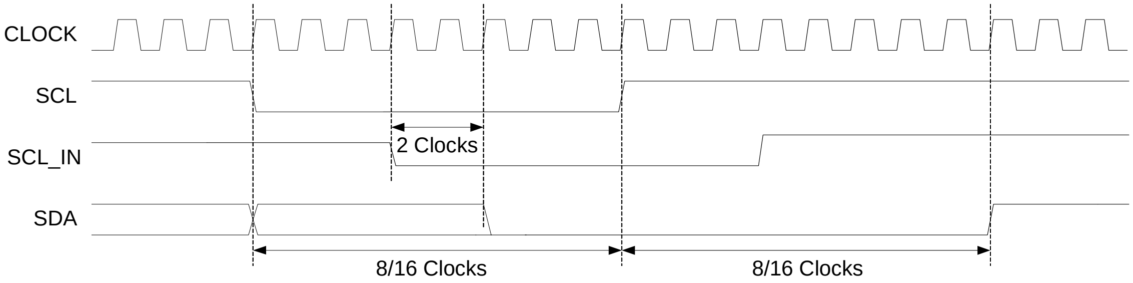


Figure 27-15. Master Stop Timing

27.4.10 Master Clock Synchronization

Figure 27-18 shows the timing associated with Master clock synchronization. Clock synchronization is always operational, even if it is the only Master on the bus. In which case, it is synchronizing to its own clock. In the wired AND bus, an SCL output of ‘0’ will be seen by all Masters. When the hardware asserts a ‘0’ to the output, it is immediately fed back from the chip pin to the input synchronizer for the SCL input. The counter value (depending on the sampling rate) takes into account the worst case latency for input synchronization of three clocks, giving a net period of 8/16 clocks for both high and low time. This results in an overall clocking rate of 16/32 clocks per bit.

In Multi-Master environments, when the hardware outputs a ‘1’ on the SCL output, if any other master is still asserting a ‘0’, the clock counter will hold until the SCL input line matches the ‘1’ on the SCL output line. When matched, the remainder of the high time is counted down. In this way, the Master with the fastest frequency determines the high time of the clock and the Master with the lowest frequency determines the low time of the clock.

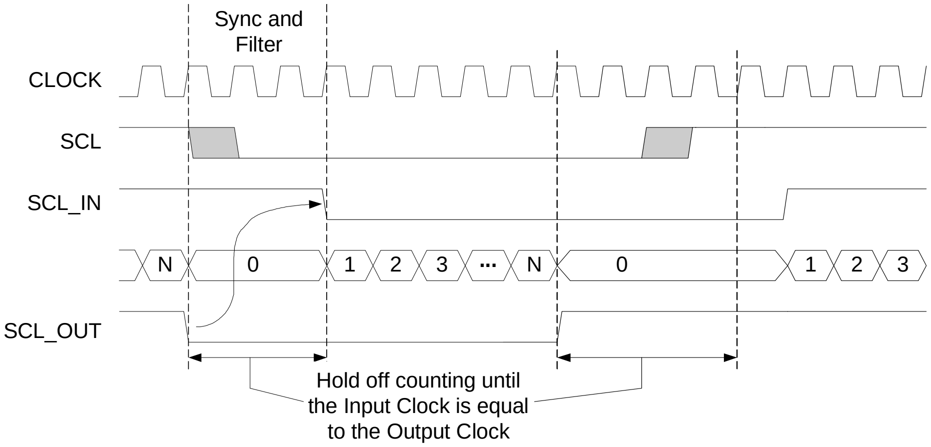

Sync and Filter
CLOCK
SCL
SCL_IN
N
0
1
2
3
...
N
0
1
2
3
SCL_OUT
Hold off counting until the Input Clock is equal to the Output Clock

Figure 27-18. Master Clock Synchronization