PSoC® Creator™
Project Datasheet for SPI_slave

Creation Time: 11/12/2014 12:39:51
User: MICKKIRKEGA1470\mickkirkegaardnielse
Project: SPI_slave

Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709
Phone (USA): 800.858.1810
Phone (Intl): 408.943.2600
http://www.cypress.com

Copyrights

PSoC and CapSense are registered trademarks of Cypress Semiconductor Corporation. PSoC Designer is a trademark of Cypress Semiconductor Corporation. All other trademarks or registered trademarks referenced herein are the property of their respective owners.

Purchase of I2C components from Cypress or one of its sublicensed Associated Companies conveys a license under the Philips I2C Patent Rights to use these components in an I2C system, provided that the system conforms to the I2C Standard Specification as defined by Philips. As from October 1st, 2006 Philips Semiconductors has a new trade name, NXP Semiconductors.

The information in this document is subject to change without notice and should not be construed as a commitment by Cypress. While reasonable precautions have been taken, Cypress assumes no responsibility for any errors that may appear in this document. No part of this document may be copied, or reproduced for commercial use, in any form or by any means without the prior written consent of Cypress.

Disclaimer

CYPRESS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Cypress reserves the right to make changes without further notice to the materials described herein. Cypress does not assume any liability arising out of the application or use of any product or circuit described herein. Cypress does not authorize its products for use as critical components in life-support systems where a malfunction or failure may reasonably be expected to result in significant injury to the user. The inclusion of Cypress' product in a life-support systems application implies that the manufacturer assumes all risk of such use and in doing so indemnifies Cypress against all charges.

Flash Code Protection

Cypress products meet the specifications contained in their particular Cypress PSoC Datasheets. Cypress believes that its family of PSoC products is one of the most secure families of its kind on the market today, regardless of how they are used. There may be methods, unknown to Cypress, that can breach the code protection features. Any of these methods, to our knowledge, would be dishonest and possibly illegal. Neither Cypress nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as "unbreakable."

Cypress is willing to work with the customer who is concerned about the integrity of their code. Code protection is constantly evolving. We at Cypress are committed to continuously improving the code protection features of our products.

Table of Contents

1 Overview

The Cypress PSoC 4 is a family of 32-bit devices with the following characteristics:

- High-performance 32-bit ARM Cortex-M0 core with a nested vectored interrupt controller (NVIC)
- Digital system that includes configurable Universal Digital Blocks (UDBs) and specific function peripherals, such as UART, SPI and I2C
- Analog subsystem that includes 12-bit SAR ADC, PWMs, comparators, op amps, CapSense, LCD drive and more
- Several types of memory elements, including SRAM and flash
- Programming and debug system through Serial Wire Debug (SWD)
- Flexible routing to all pins

Figure 1 shows the major components of a typical PSoC 4200 family member PSoC 4 device. For details on all the systems listed above, please refer to the PSoC 4 Technical Reference Manual .

Figure 1. PSoC 4200 Device Family Block Diagram

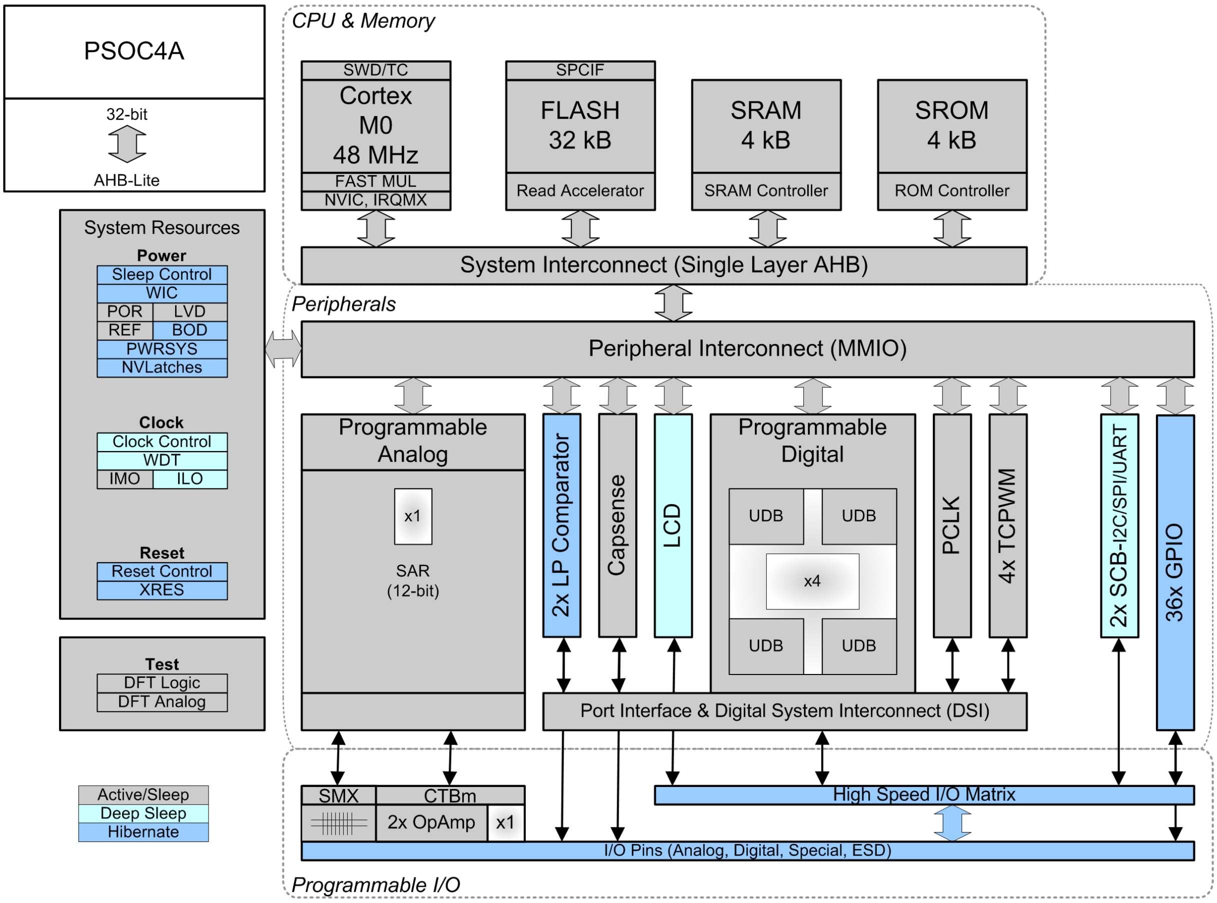


Table 1 lists the key characteristics of this device.

Table 1. Device Characteristics

Name	Value
Architecture	PSoC 4
Family	PSoC 4200
CPU speed (MHz)	48
Flash size (kBytes)	32
SRAM size (kBytes)	4
Vdd range (V)	1.71 to 5.5
Automotive qualified	No (Industrial Grade Only)
Temp range (Celcius)	-40 to 85

NOTE: The CPU speed noted above is the maximum available speed. The CPU is clocked by HFCLK, listed in the System Clocks section below.

Table 2 lists the device resources that this design uses:

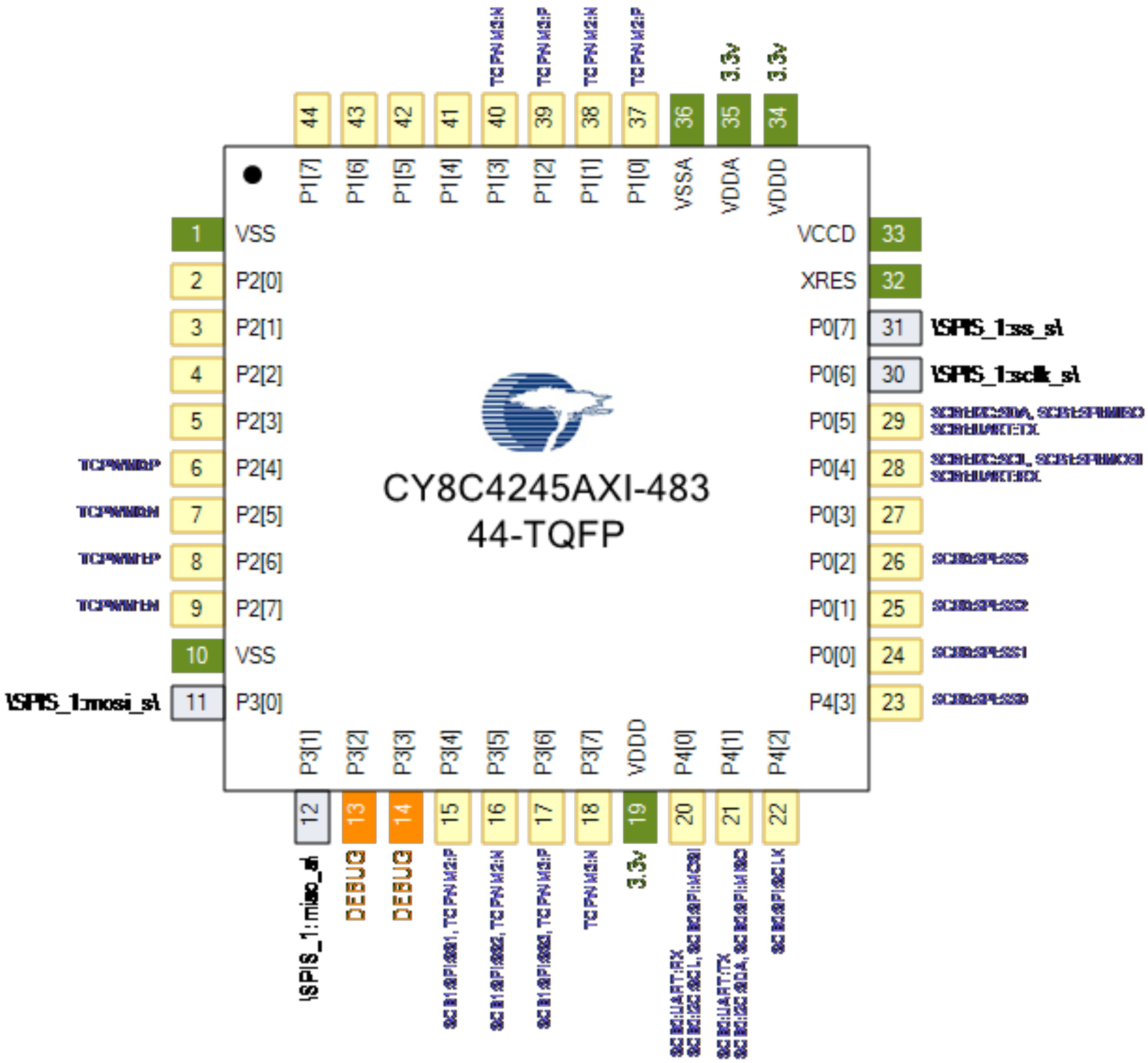
Table 2. Device Resources

Name	Resources in Use	Total Resources Available
Digital clock dividers	0 (0.0%)	4
Pins	6 (16.7%)	36
UDB Macrocells	0 (0.0%)	32
UDB Unique Pterms	0 (0.0%)	64
UDB Datapath Cells	0 (0.0%)	4
UDB Status Cells	0 (0.0%)	4
UDB Control Cells	0 (0.0%)	4
Interrupts	0 (0.0%)	32
Comparator/Opamp Fixed Blocks	0 (0.0%)	2
SAR Fixed Blocks	0 (0.0%)	1
CSD Fixed Blocks	0 (0.0%)	1
CapSense Blocks	0 (0.0%)	1
8-bit CapSense IDACs	0 (0.0%)	1
7-bit CapSense IDACs	0 (0.0%)	1
Temperature Sensor	0 (0.0%)	1
Low Power Comparator	0 (0.0%)	2
TCPWM Blocks	0 (0.0%)	4
Serial Communication Blocks	1 (50.0%)	2
Segment LCD Blocks	0 (0.0%)	1

2 Pins

Figure 2 shows the pin layout of this device.

Figure 2. Device Pin Layout



2.1 Hardware Pins

Table 3 contains information about the pins on this device in device pin order. (No connection ["n/c"] pins have been omitted.)

Table 3. Device Pins

Pin	Port	Name	Type	Drive Mode	Reset State
1	VSS	VSS	Power		
2	P2[0]	GPIO [unused]			HiZ Analog Unb
3	P2[1]	GPIO [unused]			HiZ Analog Unb
4	P2[2]	GPIO [unused]			HiZ Analog Unb
5	P2[3]	GPIO [unused]			HiZ Analog Unb
6	P2[4]	GPIO [unused]			HiZ Analog Unb
7	P2[5]	GPIO [unused]			HiZ Analog Unb
8	P2[6]	GPIO [unused]			HiZ Analog Unb
9	P2[7]	GPIO [unused]			HiZ Analog Unb
10	VSS	VSS	Power		
11	P3[0]	\SPIS_1:mosi_s\	Dgtl In	HiZ digital	HiZ Analog Unb
12	P3[1]	\SPIS_1:miso_s\	Dgtl Out	Strong drive	HiZ Analog Unb
13	P3[2]	GPIO [unused]			HiZ Analog Unb
14	P3[3]	GPIO [unused]			HiZ Analog Unb
15	P3[4]	GPIO [unused]			HiZ Analog Unb
16	P3[5]	GPIO [unused]			HiZ Analog Unb
17	P3[6]	GPIO [unused]			HiZ Analog Unb
18	P3[7]	GPIO [unused]			HiZ Analog Unb
19	VDDD	VDDD	Power		
20	P4[0]	GPIO [unused]			HiZ Analog Unb
21	P4[1]	GPIO [unused]			HiZ Analog Unb
22	P4[2]	GPIO [unused]			HiZ Analog Unb
23	P4[3]	GPIO [unused]			HiZ Analog Unb
24	P0[0]	GPIO [unused]			HiZ Analog Unb
25	P0[1]	GPIO [unused]			HiZ Analog Unb
26	P0[2]	GPIO [unused]			HiZ Analog Unb
27	P0[3]	GPIO [unused]			HiZ Analog Unb
28	P0[4]	GPIO [unused]			HiZ Analog Unb
29	P0[5]	GPIO [unused]			HiZ Analog Unb
30	P0[6]	\SPIS_1:sclk_s\	Dgtl In	HiZ digital	HiZ Analog Unb
31	P0[7]	\SPIS_1:ss_s\	Dgtl In	HiZ digital	HiZ Analog Unb
32	XRES	XRES	Power		
33	VCCD	VCCD	Power		
34	VDDD	VDDD	Power		
35	VDDA	VDDA	Power		
36	VSSA	VSSA	Power		
37	P1[0]	GPIO [unused]			HiZ Analog Unb
38	P1[1]	GPIO [unused]			HiZ Analog Unb
39	P1[2]	GPIO [unused]			HiZ Analog Unb
40	P1[3]	GPIO [unused]			HiZ Analog Unb
41	P1[4]	GPIO [unused]			HiZ Analog Unb
42	P1[5]	GPIO [unused]			HiZ Analog Unb
43	P1[6]	GPIO [unused]			HiZ Analog Unb
44	P1[7]	GPIO [unused]			HiZ Analog Unb

Abbreviations used in Table 3 have the following meanings:

- HiZ Analog Unb = Hi-Z Analog Unbuffered
- Dgtl In = Digital Input
- HiZ digital = High impedance digital
- Dgtl Out = Digital Output

2.2 Software Pins

Table 4 contains information about the software pins on this device in alphabetical order. (Only software-accessible pins are shown.)

Table 4. Software Pins

Name	Port	Type	Reset State
\SPIS_1:miso_s\	P3[1]	Dgtl Out	HiZ Analog Unb
\SPIS_1:mosi_s\	P3[0]	Dgtl In	HiZ Analog Unb
\SPIS_1:sclk_s\	P0[6]	Dgtl In	HiZ Analog Unb
\SPIS_1:ss_s\	P0[7]	Dgtl In	HiZ Analog Unb

Abbreviations used in Table 4 have the following meanings:

- Dgtl Out = Digital Output
- HiZ Analog Unb = Hi-Z Analog Unbuffered
- Dgtl In = Digital Input

For more information on reading, writing and configuring pins, please refer to:

- Pins chapter in the System Reference Guide
    - CyPins API routines
- Programming Application Interface section in the cy_pins component datasheet

3 System Settings

3.1 System Configuration

Table 5. System Configuration Settings

Name	Value
Device Configuration Mode	Compressed
Unused Bonded IO	Allow but warn
Heap Size (bytes)	0x0100
Stack Size (bytes)	0x0400
Include CMSIS Core Peripheral Library Files	True

3.2 System Debug Settings

Table 6. System Debug Settings

Name	Value
Chip Protection	Open
Debug Select	SWD (serial wire debug)

3.3 System Operating Conditions

Table 7. System Operating Conditions

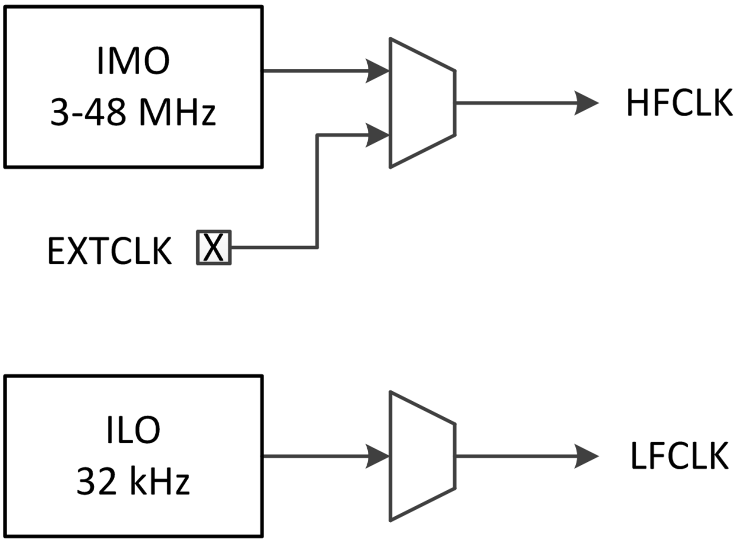
Name	Value
Vddd (V)	3.3
Vdda (V)	3.3
Variable Vdda	True
Temperature Range	-40C - 85C

4 Clocks

The clock system includes these clock resources:

- Two internal clock sources:
  - 3 to 48 MHz Internal Main Oscillator (IMO) ±2% at 3 MHz
  - 32 kHz Internal Low Speed Oscillator (ILO) output
- HFCLK can be generated using an external signal from EXTCLK pin
- Twelve clock dividers, each with 16-bit divide capability:
  - Eight can be used for fixed-function blocks
  - Four can be used for the UDBs

Figure 3. System Clock Configuration



4.1 System Clocks

Table 8 lists the system clocks used in this design.

Table 8. System Clocks

Name	Domain	Source	Desired Freq (MHz)	Nominal Freq (MHz)	Accuracy (%)	Start at Reset	Enabled
LFCLK	NONE	ILO	0	0.032	±30	True	True
ILO	NONE		0.032	0.032	±30	True	True
SYSCLK	NONE	HFCLK	0	24	±2	True	True
EXTCLK	NONE		24	0	±0	False	False
IMO	NONE		24	24	±2	True	True
HFCLK	NONE	Direct_Sel	24	24	±2	True	True

4.2 Local and Design Wide Clocks

Local clocks drive individual analog and digital blocks. Design wide clocks are a user-defined optimization, where two or more analog or digital blocks that share a common clock profile (frequency, etc) can be driven from the same clock divider output source.

Figure 4. Local and Design Wide Clock Configuration

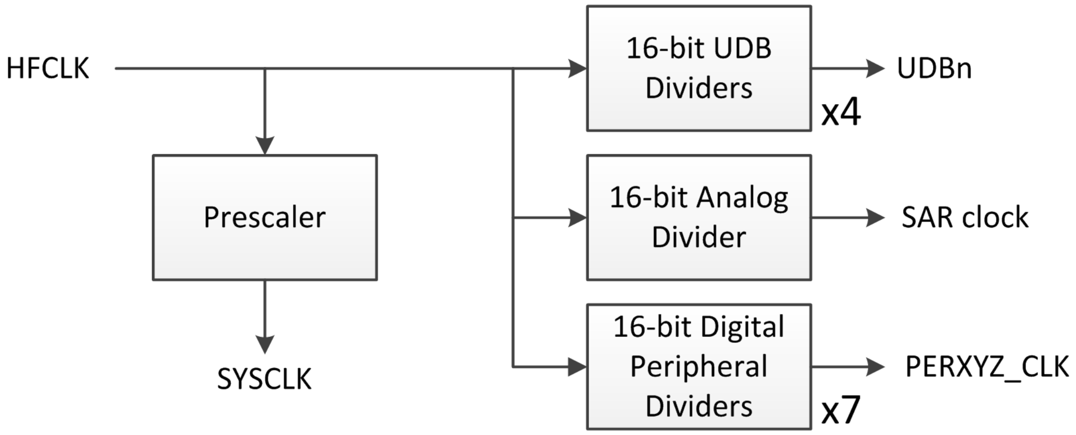


Table 9 lists the local clocks used in this design.

Table 9. Local Clocks

Name	Domain	Source	Desired Freq (MHz)	Nominal Freq (MHz)	Accuracy (%)	Start at Reset	Enabled
SPIS_1_SC-BCLK	FIXED_-FUNCT-ION	HFCLK	16	12	±2	True	True

For more information on clocking resources, please refer to:

- Clocking System chapter in the PSoC 4 Technical Reference Manual
- Clocking chapter in the System Reference Guide
    - CyIMO API routines
    - CyILO API routines

5 Interrupts

5.1 Interrupts

This design contains no interrupt components.

6 Flash Memory

PSoC 4 devices offer a host of Flash protection options and device security features that you can leverage to meet the security and protection requirements of an application. These requirements range from protecting configuration settings or Flash data to locking the entire device from external access.

Table 10 lists the Flash protection settings for your design.

Table 10. Flash Protection Settings

Start Address	End Address	Protection Level
0x0	0x7FFF	U - Unprotected

Flash memory is organized as rows with each row of flash having 128 bytes. Each flash row can be assigned one of four protection levels:

- U - Unprotected
- F - External read protect (Factory upgrade)
- R - External write protect (Field upgrade)
- W - Full Protection

For more information on Flash memory and protection, please refer to:

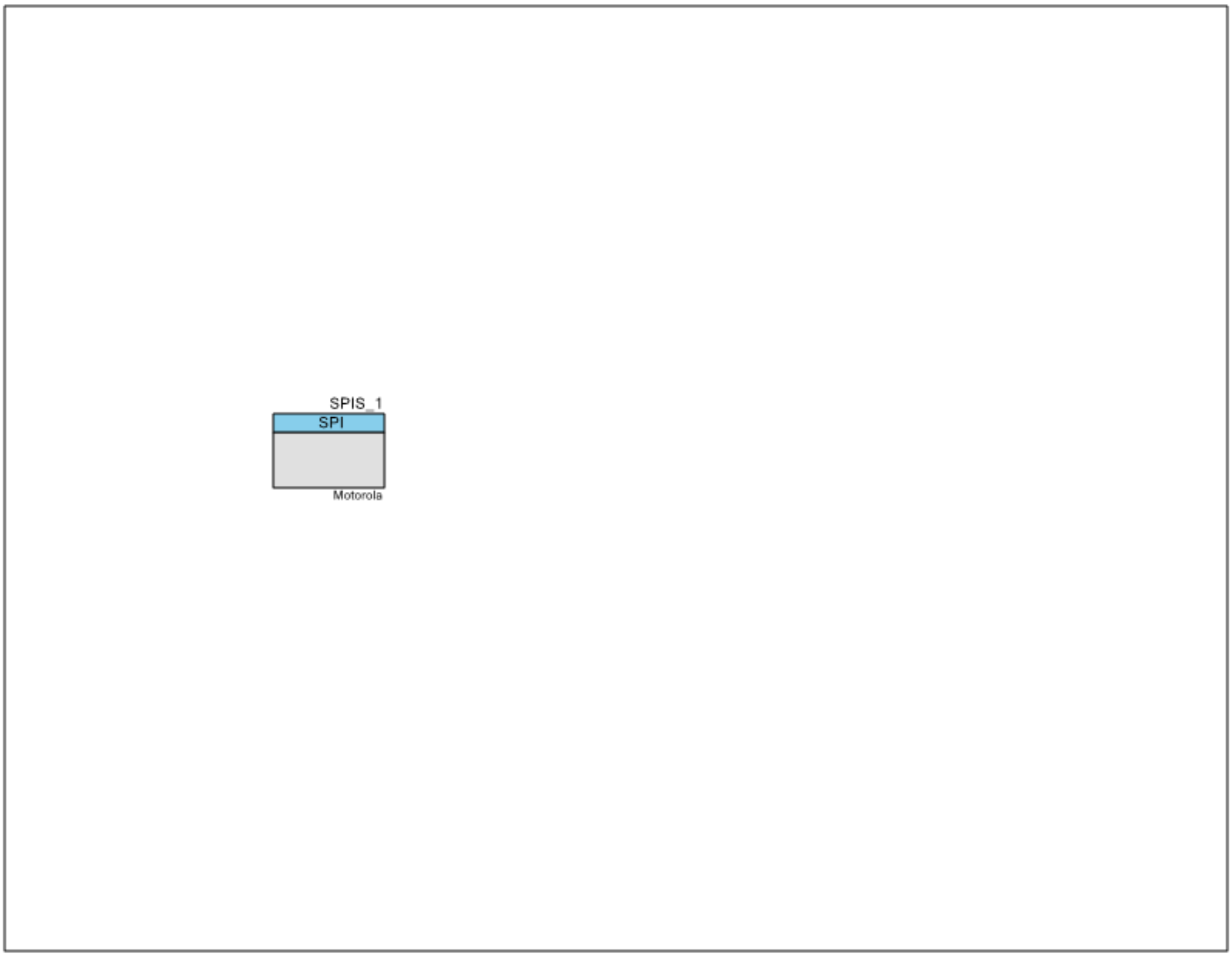
- Flash Protection chapter in the PSoC 4 Technical Reference Manual
- Flash and EEPROM chapter in the System Reference Guide
  - CyFlash API routines
  - CyWrite API routines

7 Design Contents

This design's schematic content consists of the following schematic sheet:

7.1 Schematic Sheet: Page 1

Figure 5. Schematic Sheet: Page 1



This schematic sheet contains the following component instances:

- Instance SPIS_1 (type: SCB_P4_v1_20)

8 Components

8.1 Component type: SCB_P4 [v1.20]

8.1.1 Instance SPIS_1

Description: Serial Communication Block (SCB)
Instance type: SCB_P4 [v1.20]
Datasheet: online component datasheet for SCB_P4

Table 11. Component Parameters for SPIS_1

Parameter Name	Value	Description
EzI2cClockFromTerm	false	When the SCB mode is EZI2C, this parameter provides a clock terminal to connect a clock outside the component.
EzI2cClockStretching	true	When the SCB mode is EZI2C, this parameter specifies whether the SCL is stretched while in EZI2C operation.
EzI2cDataRate	100	When the SCB mode is EZI2C, this parameter defines EZI2C Data rate in kbps. The standard data rates are: 50, 100, 400 and 1000 kbps.
EzI2cIsPrimarySlaveA-ddressHex	true	When the SCB mode is EZI2C, this parameter notifies that the EZI2C slave primary address was entered in hexadecimal. This parameter is used only by the component customizer.
EzI2cIsSecondarySlav-eAddressHex	true	When the SCB mode is EZI2C, this parameter notifies that the EZI2C slave secondary address was entered in hexadecimal. This parameter is used only by the component customizer.
EzI2cMedianFilterEnable	true	When the SCB mode is EZI2C, this parameter applies a digital 3 tap median filter to the EZI2C input lines.
EzI2cNumberOfAddresses	1	When the SCB mode is EZI2C, this parameter defines the number of I2C slave addresses that device respond to.
EzI2cOvsFactor	16	When the SCB mode is EZI2C, this parameter defines the oversampling factor of the SCBCLK.
EzI2cPrimarySlaveAddress	8	When the SCB mode is EZI2C, this parameter specifies EZI2C primary 7-bits slave address (MSB ignored).

Parameter Name	Value	Description
EzI2cSecondarySlaveAddress	9	When the SCB mode is EZI2C, this parameter specifies EZI2C secondary 7-bits slave address (MSB ignored). Only applicable when EZI2C clock stretching option is set.
EzI2cSubAddressSize	8	When the SCB mode is EZI2C, this parameter specifies the maximum size of the slave buffer that is exposed to the master: 8bits – maximum buffer size is 256 bytes, 16 bits – maximum buffer size is 65536 bytes.
EzI2cWakeEnable	false	When the SCB mode is EZI2C, this parameter enables wakeup from Deep Sleep on I2C address match event.
I2cAcceptAddress	false	When the SCB mode is I2C, this parameter specifies whether to accept a match I2C slave address in the RX FIFO or not. This option could be used for software address matching.
I2cClockFromTerm	false	When the SCB mode is I2C, this parameter allows the provision of a clock terminal to connect a clock from outside the component.
I2cDataRate	100	When the SCB mode is I2C, this parameter specifies the data rate in kbps. The standard data rates are: 50, 100, 400 and 1000 kbps.
I2cExternIntrHandler	false	When the SCB mode is I2C, this parameter specifies whether the I2C interrupt handler is configured in SCB_I2CInit(). This parameter is intended to be used by the PM/SM bus component. The modification parameter default value causes I2C mode failures.
I2cIsSlaveAddressHex	true	When the SCB mode is I2C, this parameter notifies that the I2C slave address was entered in hexadecimal. This parameter is used only by the component customizer.
I2cIsSlaveAddressMaskHex	true	When the SCB mode is I2C, this parameter notifies that the I2C slave address mask was entered in hexadecimal. This parameter is used only by the component customizer.
I2cMedianFilterEnable	true	When the SCB mode is I2C, this parameter applies a digital 3 tap median filter to the I2C lines.

Parameter Name	Value	Description
I2cMode	Slave	When the SCB mode is I2C, this parameter defines the I2C operation mode as: Slave, Master, Multi-Master or Multi-Master--Slave.
I2cOvsFactor	16	When the SCB mode is I2C, this parameter defines the oversampling factor of SCBCLK.
I2cOvsFactorHigh	8	When the SCB mode is I2C, this parameter defines the high oversampling factor of SCBCLK. Only applicable for I2C Master modes.
I2cOvsFactorLow	8	When the SCB mode is I2C, this parameter defines the low oversampling factor of SCBCLK. Only applicable for I2C Master modes.
I2cSlaveAddress	8	When the SCB mode is I2C, this parameter specifies the I2C 7-bits slave address (MSB ignored).
I2cSlaveAddressMask	254	When the SCB mode is I2C, this parameter specifies the I2C Slave address mask. Bit value 0 – excludes bit from address comparison. Bit value 1 – the bit needs to match with the corresponding bit of the I2C slave address.
I2cWakeEnable	false	When the SCB mode is I2C, this parameter enables wakeup from Deep Sleep on an I2C address match event.
ScbCustomIntrHandlerEnable	true	This parameter is reserved.
ScbMisoSdaTxEnable	true	This parameter defines the availability of the spi_miso_i2c_sda_uart_tx pin.
ScbMode	SPI	This parameter defines the mode of operation for the SCB component.
ScbMosiSclRxEnable	true	This parameter defines the availability of the spi_mosi_i2c_scl_uart_rx pin.
ScbRxWakeIrqEnable	false	This parameter defines the availability of the spi_mosi_i2c_scl_uart_rx_wake pin.
ScbSclkEnable	false	This parameter defines the availability of the sclk pin.
ScbSs0Enable	false	This parameter defines the availability of the ss0 pin.
ScbSs1Enable	false	This parameter defines the availability of the ss1 pin.
ScbSs2Enable	false	This parameter defines the availability of the ss2 pin.

Parameter Name	Value	Description
ScbSs3Enable	false	This parameter defines the availability of the ss3 pin.
SpiBitRate	1000	When the SCB mode is SPI, this parameter specifies the SPI Bit rate in kbps. The standard bit rates are: 500, 1000-8000 kbps.
SpiBitsOrder	MSB First	When the SCB mode is SPI, this parameter defines the bit order as: MSB first or LSB first.
SpiClockFromTerm	false	When the SCB mode is SPI, this parameter provides a clock terminal to connect a clock outside the component in SPI mode.
SpiInterruptMode	None	When the SCB mode is SPI, this parameter specifies the interrupt mode. None: Removes all interrupt support. Internal: Leaves the interrupt SCBIRQ inside the component - the interrupt terminal becomes invisible. External: Provides an interrupt terminal to connect an interrupt outside the component.
SpiIntrMasterSpiDone	false	When the SCB mode is SPI, this parameter enables the SCB.INTR_M. SPI_DONE inte-rrupt source. SCB.INTR_M. SPI_DONE: all data are sent into TX FIFO and the TX FIFO and the shifter register are emptied. Only applicable for SPI Master mode.
SpiIntrRxFull	false	When the SCB mode is SPI, this parameter enables the SCB.INTR_RX.FULL interrupt source. SCB.INTR_RX.FULL: RX FIFO is full.
SpiIntrRxNotEmpty	false	When the SCB mode is SPI, this parameter enables the SCB.INTR_RX.NOT_EMPTY interrupt source. SCB.INTR_RX.NOT_EMPTY: RX FIFO is not empty. There is at least one entry to get data from.
SpiIntrRxOverflow	false	When the SCB mode is SPI, this parameter enables the SCB.INTR_RX.OVERFLOW interrupt source. SCB.INTR_RX.OVERFLOW: attempt to write to a full RX FIFO.

Parameter Name	Value	Description
SpiIntrRxTrigger	false	When the SCB mode is SPI, this parameter enables the SCB.INTR_RX.TRIGGER interrupt source. SCB.INTR_RX.TRIGGER: RX FIFO has more entries than the value specified by SpiRxTriggerLevel.
SpiIntrRxUnderflow	false	When the SCB mode is SPI, this parameter enables the SCB.INTR_RX.UNDERFLOW interrupt source. SCB.INTR_RX.UNDERFLOW: attempt to read from an empty RX FIFO.
SpiIntrSlaveBusError	false	When the SCB mode is SPI, this parameter enables the SCB.INTR_SLAVE.BUS_ERROR interrupt source. SCB.INTR_SLAVE.BUS_ERROR: slave select line is deselected at an unexpected time in the SPI transfer. Only applicable for SPI Slave mode.
SpiIntrTxEmpty	false	When the SCB mode is SPI, this parameter enables the SCB.INTR_TX.EMPTY interrupt source. SCB.INTR_TX.EMPTY: TX FIFO is empty.
SpiIntrTxNotFull	false	When the SCB mode is SPI, this parameter enables the SCB.INTR_TX.NOT_FULL interrupt source. SCB.INTR_TX.NOT_FULL: TX FIFO is not full. There is at least one entry to put data.
SpiIntrTxOverflow	false	When the SCB mode is SPI, this parameter enables the SCB.INTR_TX.OVERFLOW interrupt source. SCB.INTR_TX.OVERFLOW: attempt to write to a full TX FIFO.
SpiIntrTxTrigger	false	When the SCB mode is SPI, this parameter enables the SCB.INTR_TX.TRIGGER interrupt source. SCB.INTR_TX.TRIGGER: TX FIFO has fewer entries than the value specified by SpiTxTriggerLevel.

Parameter Name	Value	Description
SpiIntrTxUnderflow	false	When the SCB mode is SPI, this parameter enables the SCB.INTR_TX.UNDERFLOW interrupt source. SCB.INTR_TX.UNDERFLOW: attempt to read from an empty TX FIFO.
SpiLateMisoSampleEnable	false	When the SCB mode is SPI, this parameter enables late sampling of the MISO line.
SpiMedianFilterEnable	false	When the SCB mode is SPI, this parameter applies a digital 3 tap median filter to the SPI input line.
SpiMode	Slave	When the SCB mode is SPI, this parameter selects SPI mode of operation as: Slave or Master.
SpiNumberOfRxDataBits	8	When the SCB mode is SPI, this parameter specifies the number of data bits inside the SPI byte/word for RX direction.
SpiNumberOfSelectLines	1	When the SCB mode is SPI, this parameter defines the number of slave select lines. The SPI Slave has only one slave select line. The SPI Master has up to 4 lines.
SpiNumberOfTxDataBits	8	When the SCB mode is SPI, this parameter define the number of data bits inside the SPI byte/word for TX direction.
SpiOvsFactor	16	When the SCB mode is SPI, this parameter defines the oversampling factor of SCBCLK.
SpiRxBufferSize	8	When the SCB mode is SPI, this parameter defines the size of the RX buffer. The value 8 implies usage of hardware RX FIFO. Greater values imply usage of internal software buffer along with RX FIFO.
SpiRxTriggerLevel	7	When the SCB mode is SPI, this parameter defines the number of entries in the RX FIFO to trigger the SCB.INTR_-RX.TRIGGER interrupt event.
SpiSclkMode	CPHA = 0, CPOL = 0	When the SCB mode is SPI, this parameter defines the serial clock phase (CPHA) and polarity (CPOL).
SpiSubMode	Motorola	When the SCB mode is SPI, this parameter defines the sub mode of the SPI as: Motorola, TI, or Microwire.

Parameter Name	Value	Description
SpiTransferSeparation	Continuous	When the SCB mode is SPI, this parameter defines the type of SPI transfers separation as: continuous or separated.
SpiTxBufferSize	8	When the SCB mode is SPI, this parameter defines the size of the TX buffer. The value 8 implies usage of hardware TX FIFO. Greater values imply usage of an internal software buffer along with TX FIFO.
SpiTxTriggerLevel	0	When the SCB mode is SPI, this parameter defines the number of entries in TX FIFO to trigger the INTR_TX.TRIGGER interrupt event.
SpiWakeEnable	false	When the SCB mode is SPI, this parameter enables wakeup from Deep Sleep on slave select event.
UartClockFromTerm	false	When the SCB mode is UART, this parameter provides a clock terminal to connect a clock outside the component.
UartDataRate	115200	When the SCB mode is UART, this parameter defines the UART baud rate in kbps. The standard baud rates are provided.
UartDirection	TX + RX	When the SCB mode is UART, this parameter enables RX or TX direction or both simultaneously.
UartDropOnFrameErr	false	When the SCB mode is UART, this parameter defines whether the data is dropped from RX FIFO on a frame error event.
UartDropOnParityErr	false	When the SCB mode is UART, this parameter determines whether the data is dropped from RX FIFO on a parity error event.
UartInterruptMode	None	When the SCB mode is UART, this parameter specifies the interrupt mode. None: Removes all interrupt support. Internal: Leaves the interrupt SCBIRQ inside the component - the interrupt terminal becomes invisible. External: Provides an interrupt terminal to connect an interrupt outside component.

Parameter Name	Value	Description
UartIntrRxFrameErr	false	When the SCB mode is UART, this parameter enables the SCB.INTR_RX.FRAME_-ERROR interrupt source. SCB.INTR_RX.FRAME_E-RROR: frame error in received data frame.
UartIntrRxFull	false	When the SCB mode is UART, this parameter enables the SCB.INTR_RX.FULL interrupt source. SCB.INTR_RX.FULL: RX FIFO is full.
UartIntrRxNotEmpty	false	When the SCB mode is UART, this parameter enables the SCB.INTR_RX.NOT_EMPTY interrupt source. SCB.INTR_RX.NOT_EMPTY: RX FIFO is not empty. There is at least one entry to get data from.
UartIntrRxOverflow	false	When the SCB mode is UART, this parameter enables the SCB.INTR_RX.OVERFLOW interrupt source. SCB.INTR_RX.OVERFLOW: attempt to write to a full RX FIFO.
UartIntrRxParityErr	false	When the SCB mode is UART, this parameter enables the SCB.INTR_RX.PARITY_-ERROR interrupt source. SCB.INTR_RX.PARITY_-ERROR: parity error in received data frame.
UartIntrRxTrigger	false	When the SCB mode is UART, this parameter enables the SCB.INTR_RX.TRIGGER interrupt source. SCB.INTR_RX.TRIGGER: RX FIFO has more entries than the value specified by UartRxTriggerLevel.
UartIntrRxUnderflow	false	When the SCB mode is UART, this parameter enables the SCB.INTR_RX.UNDERFLOW interrupt source. SCB.INTR_RX.UNDERFLOW: -attempt to read from an empty RX FIFO.
UartIntrTxEmpty	false	When the SCB mode is UART, this parameter enables the SCB.INTR_TX.EMPTY interrupt source. SCB.INTR_TX.EMPTY: TX FIFO is empty.

Parameter Name	Value	Description
UartIntrTxNotFull	false	When the SCB mode is UART, this parameter enables the SCB.INTR_TX.NOT_FULL interrupt source. SCB.INTR_TX.NOT_FULL: TX FIFO is not full. There is at least one entry to put data.
UartIntrTxOverflow	false	When the SCB mode is UART, this parameter enables the SCB.INTR_TX.OVERFLOW interrupt source. SCB.INTR_TX.OVERFLOW: attempt to write to a full TX FIFO.
UartIntrTxTrigger	false	When the SCB mode is UART, this parameter enables the SCB.INTR_TX.TRIGGER interrupt source. SCB.INTR_TX.TRIGGER: TX FIFO has fewer entries than the value specified by UartTxTriggerLevel.
UartIntrTxUartDone	false	When the SCB mode is UART, this parameter enables the SCB.INTR_TX.UART_DONE interrupt source. SCB.INTR_TX.UART_DONE: all data are sent in to TX FIFO and the transmit FIFO and the shifter register are emptied.
UartIntrTxUartLostArb	false	When the SCB mode is UART, this parameter enables the SCB.INTR_TX.UART_ARB_LOST interrupt source. SCB.INTR_TX.UART_ARB_-LOST: UART lost arbitration, the value driven on the TX line is not the same as the value observed on the RX line. This event is useful when the transmitter and the receiver - share a TX/RX line. Only applicable for UART SmartCard mode.
UartIntrTxUartNack	false	When the SCB mode is UART, this parameter enables the SCB.INTR_TX.UART_NACK interrupt source. SCB.INTR_TX.UART_NACK: UART transmitter received a negative acknowledgement. Only applicable for UART SmartCard mode.
UartIntrTxUnderflow	false	When the SCB mode is UART, this parameter enables the SCB.INTR_TX.UNDERFLOW interrupt source. SCB.INTR_TX.UNDERFLOW: - attempt to read from an empty TX FIFO.

Parameter Name	Value	Description
UartIrdaLowPower	false	When the SCB mode is UART, this parameter enables the low power receiver option. Only applicable for UART IrDA mode.
UartIrdaPolarity	Non-Inverting	When the SCB mode is UART, this parameter inverts the incoming RX line signal. Only applicable for UART IrDA mode.
UartMedianFilterEnable	false	When the SCB mode is UART, this parameter applies a digital 3 tap median filter to the UART input line.
UartMpEnable	false	When the SCB mode is UART, this parameter enables the UART multi-processor mode. Only applicable for UART Standard mode.
UartMpRxAcceptAddress	false	When the SCB mode is UART, this parameter define whether to put the matched UART address into RX FIFO. Only applicable for UART multi-processor mode.
UartMpRxAddress	2	When the SCB mode is UART, this parameter defines the UART address. Only applicable for UART multi-processor mode.
UartMpRxAddressMask	255	When the SCB mode is UART, this parameter defines the address mask in multi-processor operation mode. Bit value 0 – excludes bit from address comparison. Bit value 1 – the bit needs to match with the corresponding bit of the UART address. Only applicable for UART multi-processor mode.
UartNumberOfDataBits	8 bits	When the SCB mode is UART, this parameter defines the number of data bits inside the UART byte/word.
UartNumberOfStopBits	1 bit	When the SCB mode is UART, this parameter defines the number of Stop bits.
UartOvsFactor	12	When the SCB mode is UART, this parameter defines the oversampling factor of SCBCLK.
UartParityType	None	When the SCB mode is UART, this parameter applies UART parity check as Odd or Even or discards the parity entirely.

Parameter Name	Value	Description
UartRxBufferSize	8	When the SCB mode is UART, this parameter defines the size of the RX buffer. The value 8 implies the usage of hardware RX FIFO. Greater values imply usage of internal software buffer along with RX FIFO.
UartRxTriggerLevel	7	When the SCB mode is UART, this parameter defines the number of entries in the RX FIFO to trigger the SCB.INTR_-RX.TRIGGER interrupt event.
UartSmCardRetryOnNack	false	When the SCB mode is UART, this parameter defines whether to send a message again when a NACK response is received. Only applicable for UART SmartCard mode.
UartSubMode	Standard	When the SCB mode is UART, this parameter defines the sub mode of UART as: Standard, SmartCard or IrDA.
UartTxBufferSize	8	When the SCB mode is UART, this parameter defines the size of the TX buffer. The value 8 implies usage of hardware TX FIFO. Greater values imply the usage of internal software buffer along with TX FIFO.
UartTxTriggerLevel	0	When the SCB mode is UART, this parameter defines the number of entries in the TX FIFO to trigger the SCB.INTR_-TX.TRIGGER interrupt event.
UartWakeEnable	false	When the SCB mode is UART, this parameter enables the wakeup from Deep Sleep on start bit event. The actual wakeup source is RX GPIO. The skip start UART feature allows it to continue receiving bytes.

9 Other Resources

The following documents contain important information on Cypress software APIs that might be relevant to this design:

- Standard Types and Defines chapter in the System Reference Guide
    - Software base types
    - Hardware register types
    - Compiler defines
    - Cypress API return codes
    - Interrupt types and macros
- Registers
    - The full PSoC 4 register map is covered in the PSoC 4 Registers Technical Reference Manual
    - Register Access chapter in the System Reference Guide
        - CY_GET API routines
        - CY_SET API routines
- System Functions chapter in the System Reference Guide
    - General API routines
    - CyDelay API routines
    - CyVd Voltage Detect API routines
- Power Management
    - Power Supply and Monitoring chapter in the PSoC 4 Technical Reference Manual
    - Low Power Modes chapter in the PSoC 4 Technical Reference Manual
    - Power Management chapter in the System Reference Guide
        - CyPm API routines
- Watchdog Timer chapter in the System Reference Guide
    - CyWdt API routines