CE209976 – SPI Slave Select Inversion with Smart IO®

Objective

This example demonstrates how to use the Smart IO® Component in PSoC® 4 to invert the output of the serial communication block (SCB) SPI slave select signal. This method can be used for any of the SCB unidirectional signals to invert the signal polarity.

Overview

One application of the Smart IO Component is to invert the polarity of SCB communication signals without using an external inverter. This example demonstrates how to invert the SPI Master (SCB) Component's slave select signal using the Smart IO Component. The example can be applied to other unidirectional SCB signals (I2C is not supported).

Requirements

Tool: PSoC Creator 3.3 SP2

Programming Language: C (GCC 4.9), ARM MDK

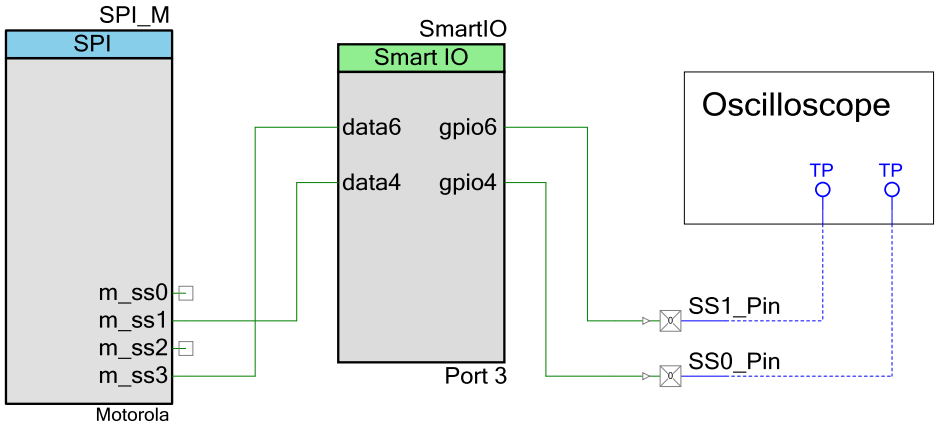
Associated Parts: PSoC 4000S, PSoC 4100S, PSoC 4200L

Related Hardware: CY8CKIT-041, CY8CKIT-046

Design

The SCB Component is configured as a SPI Master with two slave select pin connections exposed as symbol terminals. The MOSI, MISO and SCLK pins are not used in this design. The slave select terminals are connected to data4 and data6 terminals of the Smart IO Component as shown in Figure 1. The slave select configurations are active low. The Smart IO Component is configured to take the data4 signal, invert it through LUT4 and then output to gpio4. It tn akes the data6 slave select signal and replicates it to gpio6 by using LUT6. The gpio4 and gpio6 terminals are connected to the SS0_Pin and SS1_Pin, respectively.

Figure 1. Design Schematic



The firmware for the Component is implemented in *main.c* and performs the following functions.

1. Starts the Smart IO Component.
2. Starts the SCB Component.

3. Selects the slave on ss1 and sends sample data.
4. Selects the slave on ss3 and sends sample data.

Design Considerations

This code example is designed for the specified ports on the stated devices. The design is portable to other PSoC 4 devices with Smart IO, but it may require LUT reconfiguration due to the close relationship between the device port and the component.

Note SCB connections use a dedicated channel and cannot be routed to other gpio pins using the Smart IO Component. If an SCB connection is located on data4 then it must connect to gpio4.

Hardware Setup

Connect SS0_Pin and SS1_Pin to an oscilloscope.

Note these pins are also connected to two of the RGB LED connections for the CY8CKIT-041 (Red and Blue).

Software Setup

The Smart IO Component is a port-wide resource; you must define its port before it can be used. Follow these steps to configure your design.

CY8CKIT-041

Default Smart IO port is Port 3. No changes necessary.

CY8CKIT-046

1. Open the Smart IO configuration dialog in the design schematic, and define the Port parameter to be 11.
2. Click OK, and close. Rebuild the PSoC Creator project.

Components

Table 1 lists the PSoC Creator Components used in this example, as well as the hardware resources used.

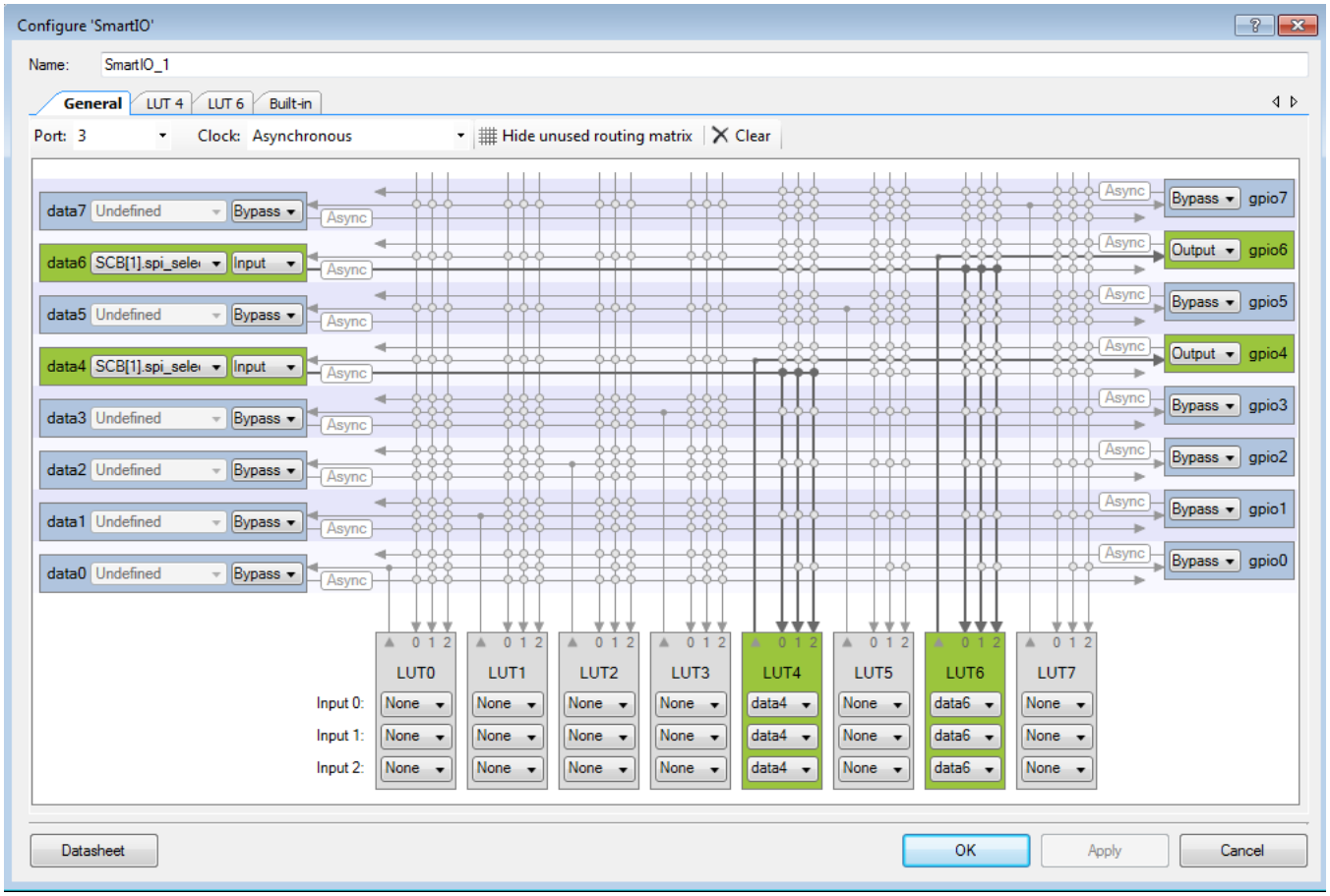
Table 1. List of PSoC Creator Components

Component	Hardware Resources
SmartIO	1 PRGIO
Digital Output Pins	2 pins
SPI_M	1 SCB

Parameter Settings

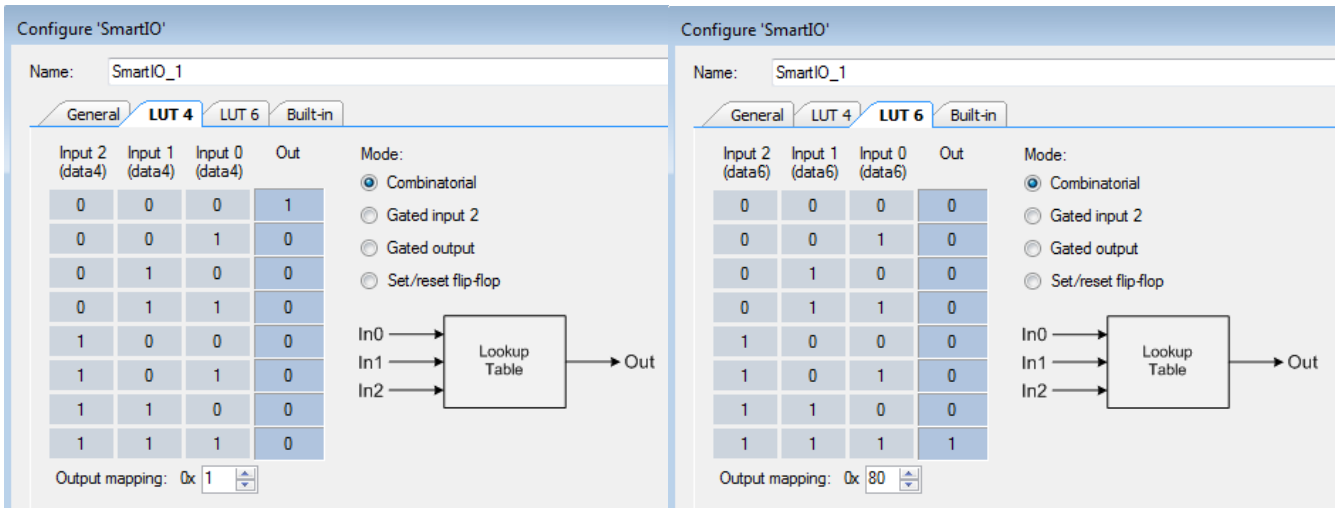
The Smart IO Component is configured to route data4 to gpio4 through LUT4, as Figure 2 shows. It is also configured to route data6 to gpio6 through LUT6.

Figure 2. Smart IO Routing Configuration



The LUTs are configured in “combinatorial” mode. LUT4 is configured to invert the input signal, whereas LUT6 is configured to repeat the input signal. These settings are located in the individual LUT tabs in the configuration dialog.

Figure 3. Smart IO LUT Configuration



Operation

Plug the kit into your computer’s USB port. Program the project into the kit, and observe on the oscilloscope that the signal on SS0_Pin exhibits an active high behavior. Also observe that SS1_Pin exhibits an active low behavior and produces a signal that is of opposite polarity to SS0_Pin.

Related Documents

Table 2 lists all relevant application notes, code examples, knowledge base articles, device datasheets, and Component datasheets.

Table 2. Related Documents

Code Examples		
CE209974	PSoC®4 Breathing LED with Smart IO®	Implements a breathing LED using a PWM and a Smart IO component.
CE209975	PSoC®4 Clock Buffer with Smart IO®	Uses the Smart IO to implement a clock buffer that can operate in chip deep-sleep mode
PSoC Creator Component Datasheets		
Smart IO	Supports Smart IO peripheral	
SCB	Supports SPI, UART and I2C serial communication	
Pins	Supports connection of hardware resources to physical pins	
Device Documentation		
PSoC 4 Datasheets		
PSoC 4 Technical Reference Manuals		
Development Kit (DVK) Documentation		
PSoC 4 Kits		

Document History

Document Title: SPI Slave Select Inversion with Smart IO® - CE209976

Document Number: 002-09976

Revision	Date	Description of Change
**	04/15/16	New code example.

Worldwide Sales and Design Support

Cypress maintains a worldwide network of offices, solution centers, manufacturer's representatives, and distributors. To find the office closest to you, visit us at Cypress Locations.

Products

Automotive	cypress.com/go/automotive
Clocks & Buffers	cypress.com/go/clocks
Interface	cypress.com/go/interface
Lighting & Power Control	cypress.com/go/powerpsoc cypress.com/go/plc
Memory	cypress.com/go/memory
PSoC	cypress.com/go/psoc
Touch Sensing	cypress.com/go/touch
USB Controllers	cypress.com/go/usb
Wireless/RF	cypress.com/go/wireless

PSoC® Solutions

psoc.cypress.com/solutions

PSoC 1 |PSoC 3 | PSoC 4 |PSoC 5LP

Cypress Developer Community

Community | Forums |Blogs | Video |Training

Technical Support

cypress.com/go/support

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

Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709

Phone : 408-943-2600
Fax : 408-943-4730
Website : www.cypress.com

© Cypress Semiconductor Corporation, 2016. This document is the property of Cypress Semiconductor Corporation and its subsidiaries, including Spansion LLC ("Cypress"). This document, including any software or firmware included or referenced in this document ("Software"), is owned by Cypress under the intellectual property laws and treaties of the United States and other countries worldwide. Cypress reserves all rights under such laws and treaties and does not, except as specifically stated in this paragraph, grant any license under its patents, copyrights, trademarks, or other intellectual property rights. If the Software is not accompanied by a license agreement and you do not otherwise have a written agreement with Cypress governing the use of the Software, then Cypress hereby grants you a personal, non-exclusive, nontransferable license (without the right to sublicense) (1) under its copyright rights in the Software (a) for Software provided in source code form, to modify and reproduce the Software solely for use with Cypress hardware products, only internally within your organization, and (b) to distribute the Software in binary code form externally to end users (either directly or indirectly through resellers and distributors), solely for use on Cypress hardware product units, and (2) under those claims of Cypress's patents that are infringed by the Software (as provided by Cypress, unmodified) to make, use, distribute, and import the Software solely for use with Cypress hardware products. Any other use, reproduction, modification, translation, or compilation of the Software is prohibited.
TO THE EXTENT PERMITTED BY APPLICABLE LAW, CYPRESS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS DOCUMENT OR ANY SOFTWARE OR ACCOMPANYING HARDWARE, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. To the extent permitted by applicable law, Cypress reserves the right to make changes to this document without further notice. Cypress does not assume any liability arising out of the application or use of any product or circuit described in this document. Any information provided in this document, including any sample design information or programming code, is provided only for reference purposes. It is the responsibility of the user of this document to properly design, program, and test the functionality and safety of any application made of this information and any resulting product. Cypress products are not designed, intended, or authorized for use as critical components in systems designed or intended for the operation of weapons, weapons systems, nuclear installations, life-support devices or systems, other medical devices or systems (including resuscitation equipment and surgical implants), pollution control or hazardous substances management, or other uses where the failure of the device or system could cause personal injury, death, or property damage ("Unintended Uses"). A critical component is any component of a device or system whose failure to perform can be reasonably expected to cause the failure of the device or system, or to affect its safety or effectiveness. Cypress is not liable, in whole or in part, and you shall and hereby do release Cypress from any claim, damage, or other liability arising from or related to all Unintended Uses of Cypress products. You shall indemnify and hold Cypress harmless from and against all claims, costs, damages, and other liabilities, including claims for personal injury or death, arising from or related to any Unintended Uses of Cypress products.
Cypress, the Cypress logo, Spansion, the Spansion logo, and combinations thereof, PSoC, CapSense, EZ-USB, F-RAM, and Traveo are trademarks or registered trademarks of Cypress in the United States and other countries. For a more complete list of Cypress trademarks, visit cypress.com. Other names and brands may be claimed as property of their respective owners.