# iCE65 as SPI Slave

SiliconBlue™

**08-DEC-2010 (1.1)** **Design Example #025**

## Overview

The Serial Peripheral Interface Bus or SPI bus is a synchronous serial data link standard that operates in full duplex mode. Devices communicate in master/slave mode where the master device initiates the data transfer.

This design example illustrates the implementation of SPI Slave on SiliconBlue's low power iCE65 FPGAs. This SPI Slave design supports all modes of CPOL and CPHA - 00, 01, 10, 11. SPI Slave data sampling from MOSI line depends upon current bit count and CPOL/CPHA mode. A shift register on receive data path converts serial to parallel conversion. Similarly parallel to serial conversion takes place in transmit data path.

## Features Supported

- Supports all 4 modes of CPOL and CPHA operation (00/01/10/11).
- Supports variable data width(8 and 16)
- Provision for easy integration of any processor interface.
- IP-XACT version 1.2 compliant.

## Features not Supported

- Read and Write Data FIFOs
- Interrupt Generation

## Resource Utilization

*Table 1:* **Resource Utilization**

| LUTs | Registers | Memory | I/Os | GBs |
|---|---|---|---|---|
| 211 | 120 | 0 | 0 | 0 |

Note: Resource Utilization is based on iCEcube 2010.12.14671 release.

## System Block Diagram

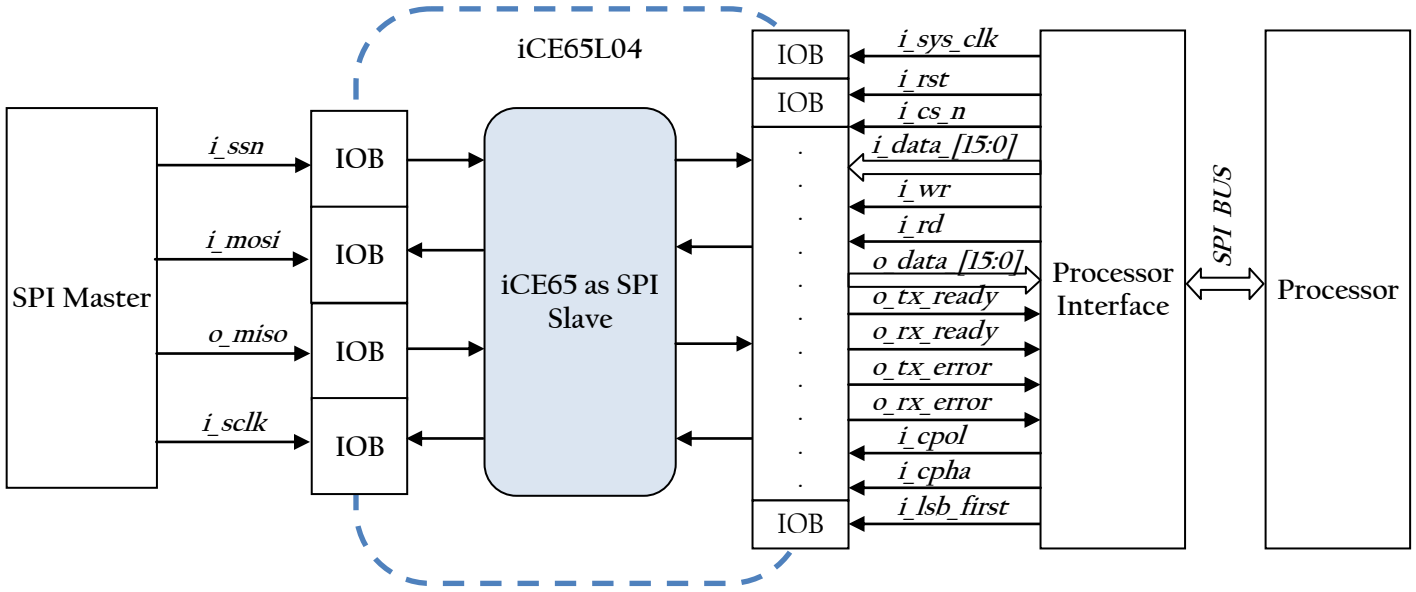



*Figure 1:* **System Block Diagram**

## Functional Block Diagram

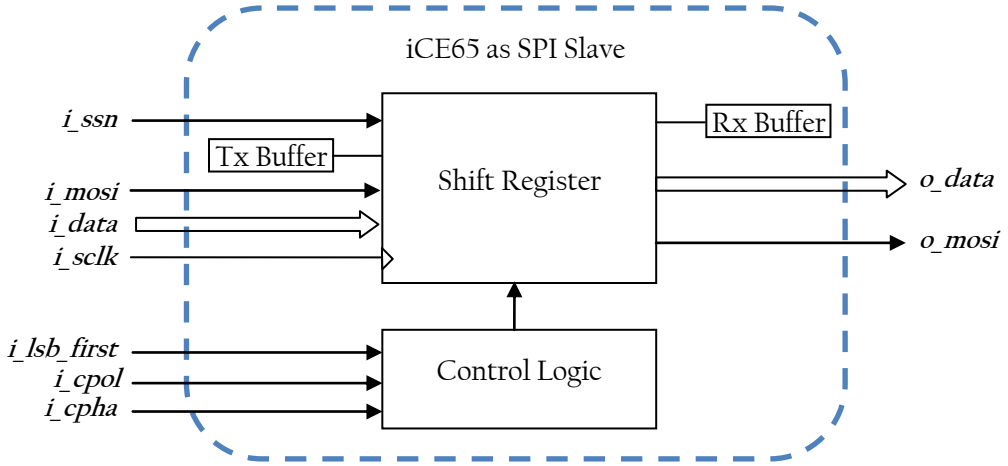



*Figure 2:* **Functional Block Diagram**

## Design Interface

*Table 2:* **Pin Description**

| Signal Name | Pin Type | Signal Description |
|---|---|---|
| **i_csn** | Input | Active low chip select. |
| **i_data [15:0]** | Input | Input data from Processor Interface. |
| **i_wr** | Input | Active low write enable |
| **i_rd** | Input | Active high read enable |
| **o_data [15:0]** | Output | Output data to Processor Interface. |
| **o_tx_ready** | Output | Transmitter ready status– A HIGH indicates transmitter is ready to send another data. |
| **o_rx_ready** | Output | Receiver ready – A High indicates receiver is ready to receive another data. |
| **o_tx_error** | Output | Indicates error in transmission of data. |
| **o_rx_error** | Output | Indicates error in reception of data. |
| **i_cpol** | Input | Polarity of the clock |
| **i_cpha** | Input | Phase of the clock |
| **i_lsb_first** | Input | LSB sent first when '1' and MSB goes first when '0'. |
| **o_miso** | Output | Slave output to Master |
| **i_mosi** | Input | Slave input from Master |
| **i_ssn** | Input | Slave select from Master |
| **i_sclk** | Input | Serial Clock from Master. |
| **i_sys_rst** | Input | Asynchronous active low reset. |
| **i_sys_clk** | Input | System clock. |

## Configurable Parameters

- DATA_SIZE – This parameter configures the data width. Its default value is 16, it can be configured to 8 as well.

## Register Map

This design does not have any user accessible registers or memory..

## Design Details

### SPI Slave Receiver :

MOSI Sampling – SPI Slave receives the data on MOSI line based on CPOL and CPHA modes as follows :

Sample at positive edge of SCLK for

1. i_cpol = '0' and i_cpha = '0'
2. i_cpol = '1' and i_cpha = '1'

Sample at negative edge of SCLK for

1. i_cpol = '1' and i_cpha = '0'
2. i_cpol = '0' and i_cpha = '1'

At the end of reception of data, rx_ready goes High and a valid received data is available on the bus. If a new data has arrived and last received data has not read yet then rx_error signal goes High to indicate receive error.

### SPI Slave Transmitter :

Sending data on MISO line – SPI Slave transmits the data on MISO line from a shift register based on CPOL and CPHA as follows :

1. CPOL = 0 and CPHA = 0 : Data is placed before rising edge of sclk

2. CPOL = 1 and CPHA = 0 : Data is placed before falling edge of sclk
3. CPOl = 0 and CPHA = 1 : Data is placed at rising edge of sclk
4. CPOL = 1 and CPHA = 1 : Data is placed at falling edge of sclk

At the end of transmission of data, tx_ready signal goes High. When transmitter busy transmitting a data stream, then tx_error signal goes high if transmit buffer data is over-written by processor interface.

### Clock Requirements

For proper operation of SPI Slave, System frequency should be minimum twice that of SCL Frequency.

### Initialization Conditions

An asynchronous active low reset signal assertion is necessary to initialize the SPI Slave to proper operating state. Receive and Transmit buffers are initialized to zero during the reset condition.

## Timing Diagram

### Signals definition:

i_sclk – Serial Clock Line (Generated by Master)

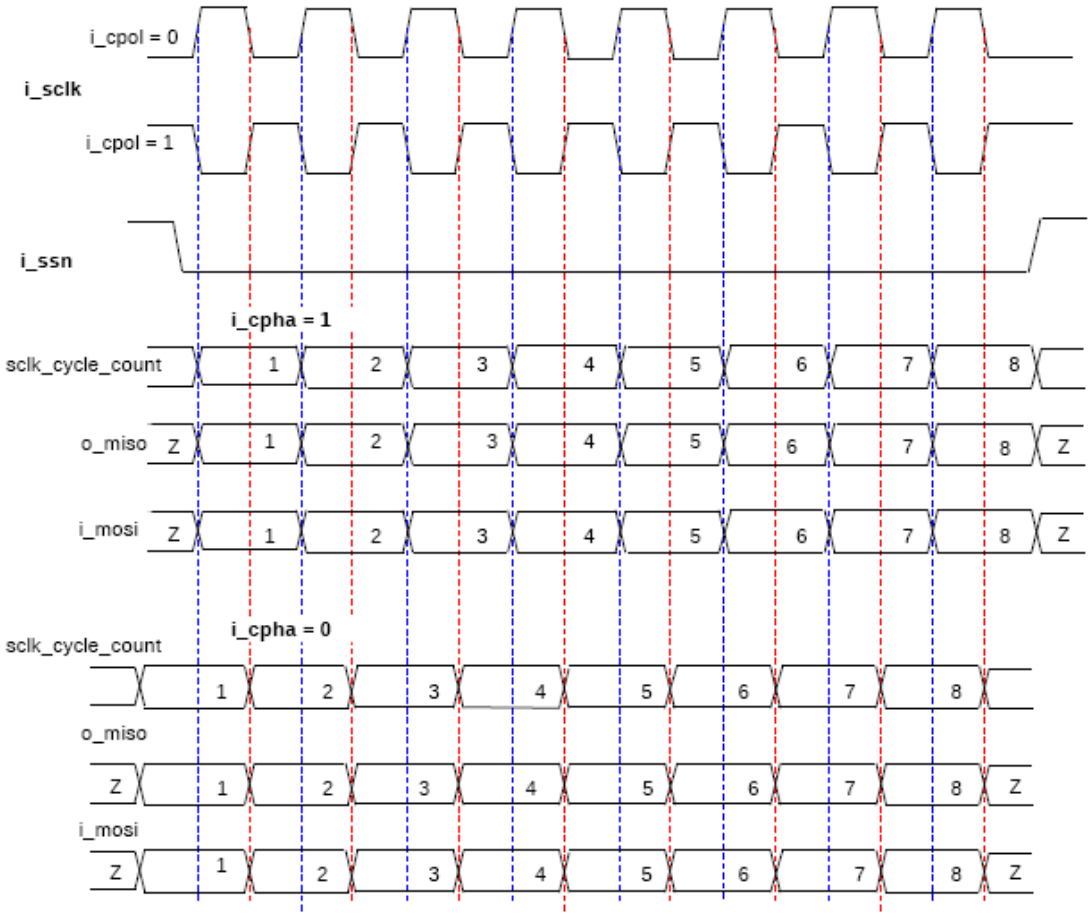
i_cpol- Clock Polarity

i_cpha – Clock Phase

i_ssn – Slave Select

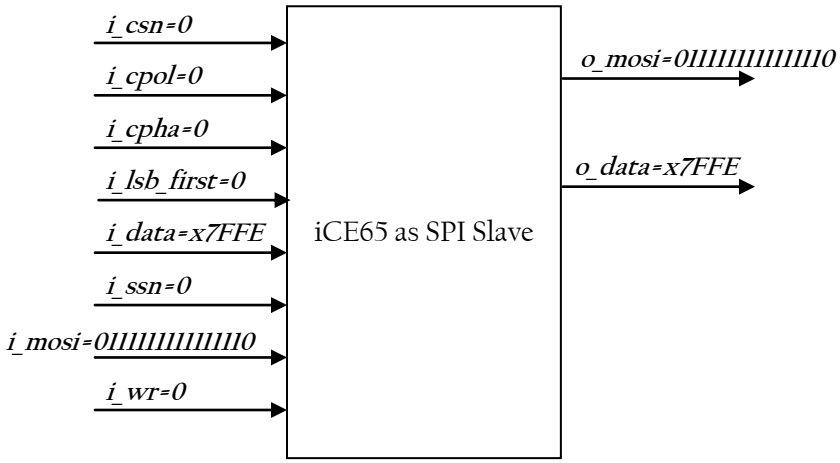
o_miso – Output from SPI Slave

i_mosi – Input to SPI Slave



### Usage Examples

Example #1




In the above example, Slave samples MOSI line at positive edge of SCLK and Slave puts the data on MISO line before the rising edge of SCLK.

## System Designer Flow

SPI Slave is compatible with System Designer/IP-XACT 1.2. Following parameter can be configured in the System Designer environment.

- DATA_SIZE – This parameter configures the data width. Its default value is 16, it can be configured to 8 as well.

The System Designer flow is as follows,

1. Launch the System Designer from Synplify Pro using menu 'Import -> Launch System Designer'.
2. Create a new project(open an existing old project, as necessary) and import the IP-XACT XML file
3. Drag and place the component from the 'Library' pane to the 'Design' pane
4. To change the data width, right-click on the component instance, and click on “Open Configuration”. Go to “Edit Instance Parameters” tab, change the “DATA_SIZE” parameter. Click on the “Apply” button, and then close it.
5. Click on the “Generate Files” button, which generates the necessary files required for synthesis and simulation.
6. Go to Synplify Pro and click on the “Run” button to synthesize the System Designer generated files. Synplify Pro generates all the necessary files for P&R in iCECube.

## References

The following references were used in the creation of this design:

- SiliconBlue Technologies, Inc. "iCE65 Ultra Low-Power mobileFPGA Family" datasheet (26-MAY-2010).
- http://en.wikipedia.org/wiki/Serial_Peripheral_Interface_Bus

## Revision History

| Version | Date | Description |
|---|---|---|
| **1.0** | 16-SEP-2010 | Initial Draft Document |
| **1.1** | 08-DEC-2010 | IP-XACT format Update |

## Disclaimer



SiliconBlue™

SiliconBlue Technologies Corporation

3255 Scott Blvd.,
Building 7, Suite 101
Santa Clara, CA 95054

Tel: 408-727-6101
Fax: 408-727-6085

www.SiliconBlueTech.com