# iCE65 as SPI Slave Port Expander

SiliconBlue™

**04-DEC-2010 (1.1)** **Design Example #005**

## 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 an SPI Slave Port Expander using iCE65 FPGAs. Port expanders provide the system host the capability and advantage of expanding its ports thereby reducing pins on the host system. This port expander uses the standard 4 wire SPI bus interface that are available on most host systems to expand into 32 serial input and 32 serial output ports.

## Features Supported

- Expansion up to 32 serial input and output ports
- All modes of CPOL and CPHA (00/01/10/11)
- Compile-time configurable data widths (8, 16 and 32)
- 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 | GBs | I/Os |
|---|---|---|---|---|
| 75 | 29 | 0 | 0 | 0 |

Note: Resource Utilization is based on iCECube 2010.12.14671 release.

## System Block Diagram

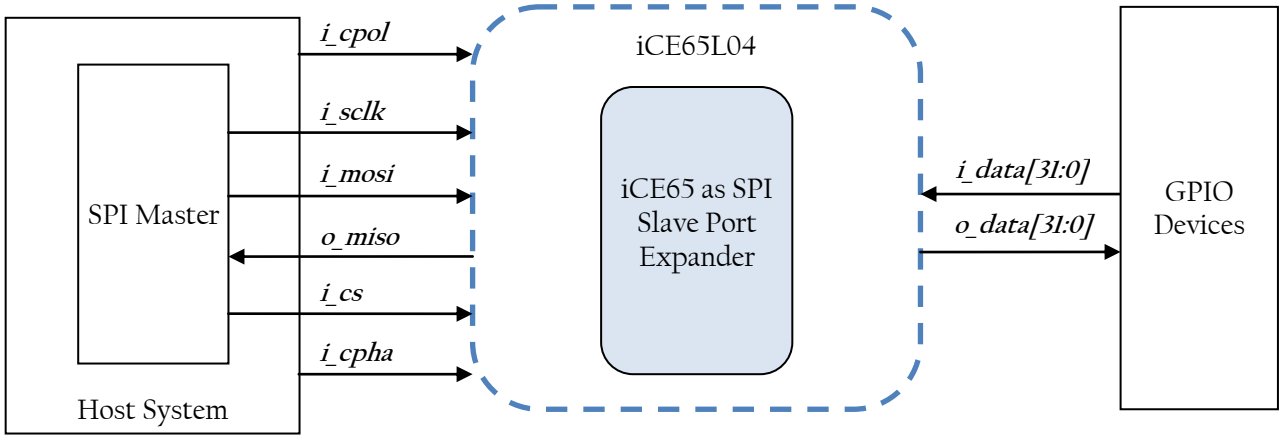


*Figure 1:* **System Block Diagram**

## Functional Block Diagram

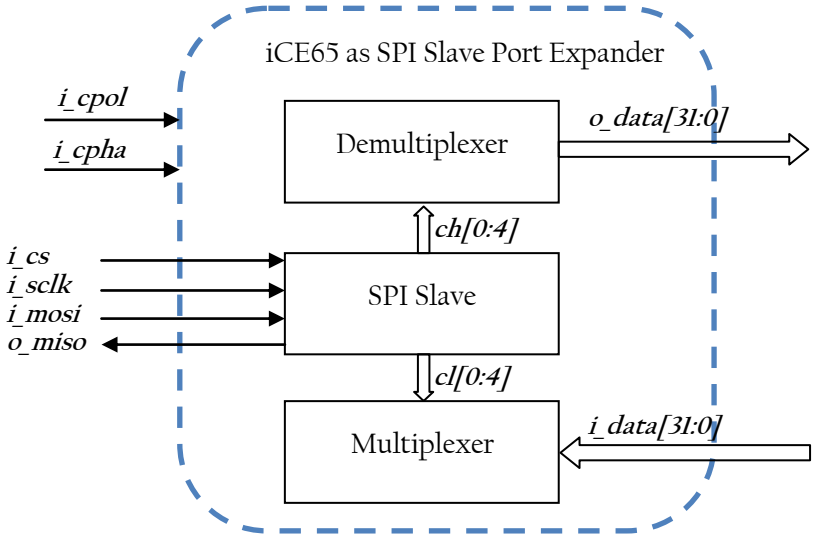


*Figure 2:* **Functional Block Diagram**

## Design Interface

*Table 2:* **Pin Description**

| Signal Name | Pin Type | Signal Description |
|---|---|---|
| **i_cs** | Input | Active high chip select (Acts like Reset) |
| **i_data [31:0]** | Input | 32 expanded input ports |
| **o_data [31:0]** | Output | 32 expanded output ports |
| **i_mosi** | Input | Slave input from Master |
| **o_miso** | Output | Slave output to Master |
| **i_sclk** | Input | Serial Clock from Master |
| **i_cpol** | Input | Clock Polarity |
| **i_cpha** | Input | Clock Phase |

## Configurable Parameters :

- GPIO_WIDTH : This parameter controls the number of expandable input and output ports. Supported values are 8, 16 (default) and 32.

## Register Map

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

## Design Details

SPI™, or Serial Peripheral Interface is a popular 4 wire serial interface that is adapted in most systems. It is a Master - Slave system using 4 lines (3 common and 1 exclusive) as follows:

- MOSI (Master Out, Slave In)
- MISO (Master In, Slave Out)
- SCLK (Serial Clock)
- CS (Chip select or Slave select)

The main features of the SPI interface, unlike many other serial interfaces, are that it allows for a full duplex serial communication. This is possible due to the presence of exclusive lines for data in both directions. The availability of another exclusive line used to select a particular Slave, the CS line, reduces the overhead of address decoding in a multi-Slave environment. These two features combine to add great speeds on the SPI bus (with clock speeds up to 70 MHz).

This design example of the SPI Slave Port Expander is implemented with configuring the iCE FPGA as an SPI Slave. The Slave connects to the system SPI bus and expands to 32 serial inputs and 32 serial outputs.

There are 3 data formats supported in this IP – 32-bit, 16-bit and 8-bit (shown in Table 3, Table 4 and Table 5). By default, it has 16-bit data format. The SPI slave port expander is implemented in the iCE65 FPGA by decoding appropriate bits in the SPI's MOSI data frame to obtain the address of the input and output port the Slave has to expand to. Once this address is decoded in the Slave, the serial data that the Master sends to the Slave on its MOSI line is routed to the appropriate output port on the Slave. At the same time the serial input data that the Slave received on a selected input port is routed on to the MISO line. This selection of the input port is also by means of an address in the SPI Master's MOSI data frame (illustrated in the below tables). This port expander is designed for a 32 bit SPI frame, with the MSB being sent first.

*Table 3:* **32-bit data frame format**

| Input Address [31 : 27] | Output Address [26 : 22] | Data [21 : 0] |
|---|---|---|

*Table 4:* **16-bit data frame format**

| Input Address [31 : 28] | Output Address [27 : 24] | Data [23 : 0] |
|---|---|---|

*Table 5:* **8-bit data frame format**

| Input Address [31 : 29] | Output Address [28 : 26] | Data [25 : 0] |
|---|---|---|

### Initialization Conditions

i_cs acts as Asynchronous active low reset. At reset, i_data and o_data are initialized to x"FFFFFFFF".

## Timing Diagram

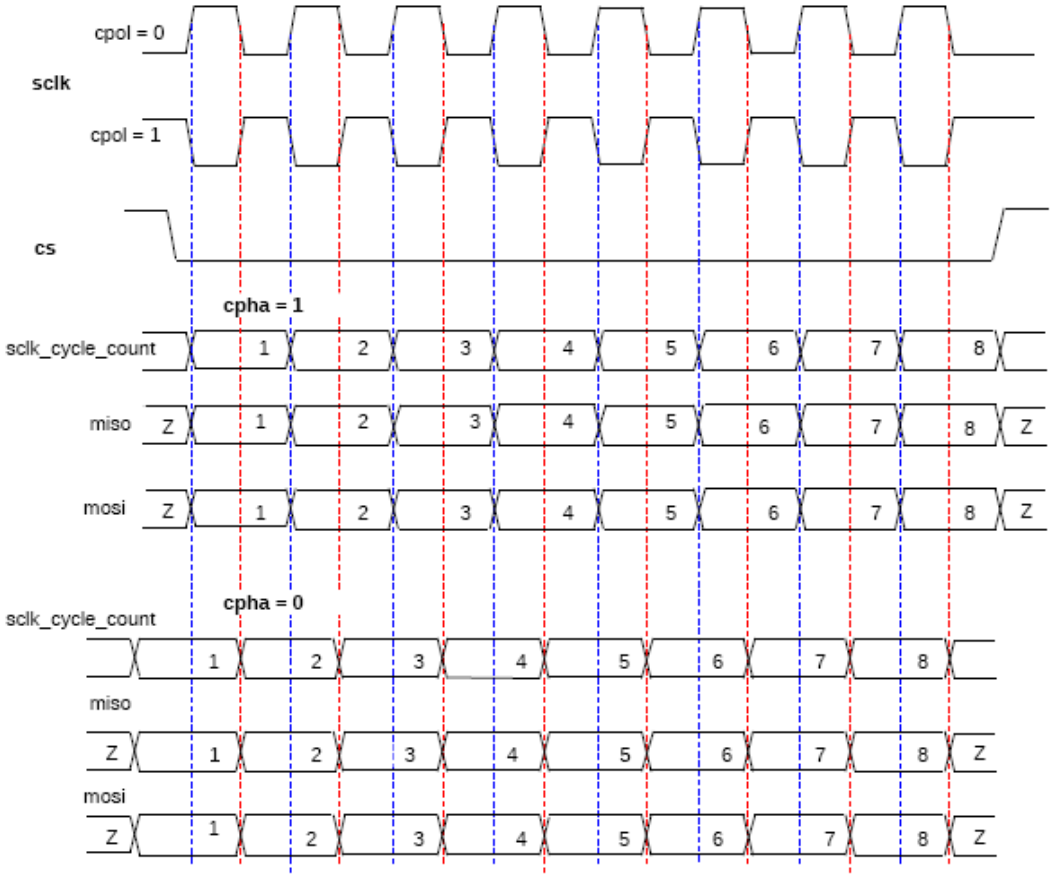


*Figure 3:* **Timing Diagram showing the various modes with CPOL and CPHA**

## Usage Examples

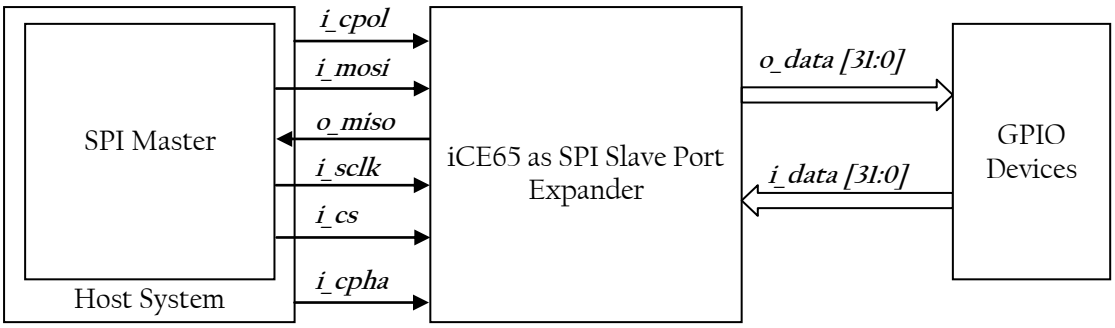


*Figure 4:* **Usage example**

- Make i_cs = '0', i_cpol = '0' and i_cpha = '0'
- Write the 32-bit SPI Data frame (MSB first) through the MOSI line with input and output port address as 0, and data as all 1s i.e. "00000000111111111111111111111111"
- Check if the data at the output port (o_data) is x"FFFFFFFF"
- Repeat the above steps by changing the input and output address, the polarities of i_cpol and i_cpha, as well as changing the data format to 8-bit and 32-bit.

## System Designer Flow

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

- GPIO_WIDTH – This parameter controls the number of expandable input and output ports. The values supported are 8, 16 and 32, with 16 being the default value.

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 modify the number of expandable input and output ports, right-click on the component instance, and click on "Open Configuration". Go to "Edit Instance Parameters" tab, change the "GPIO_WIDTH" 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 iCECube2.

## 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** | 04-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