

## iCE65 as SPI to UART Interface

### Overview

SPI to UART interface provides a cost-effective solution for communicating from host processor's SPI bus to multiple UART devices. This design example acts as an SPI port expander, multiplexing and demultiplexing read/write data through SPI Slave to multiple UARTs using iCE FPGAs. .

### Description

SPI to UART Interface module interfaces host processor/microcontroller's SPI master to a number of UARTs through SPI Slave. This interface module consists of two modules 1) SPI Slave which interfaces with to SPI master in host processor and acts as port expander 2) Multiplexer/Demultiplexer to write/read data from UARTs. This interface module selects appropriate UART device, based on address and mode bits on the MOSI frame performs following operations 1. Writes configuration register and configures baud rate, data length, enables/disables parity and parity type (even/odd) 2. Writes data to UART write register/FIFO 3. Reads data UART status register 4. Reads data from UART's read FIFO. The data read from UART is communicated to host processor through MISO line. SPI Slave operates in CPOL = 0 and CPHA = 0 mode, MSB first read/write and configured for 16-bit SPI data frame.

### Implementation

Figure 1 and Figure 2 shows the simplified block diagram of SPI to UART interface and simplified state diagram. SPI to UART expansion and interface is achieved in the iCE65 FPGA by decoding appropriate bits in the SPI's MOSI data frame to obtain the address of the UART 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 detects the mode of operation of UART, viz., read data, read status, write data and write configuration. At the same time the serial input data that the Slave received on a selected input UART is routed on to the MISO line based on bit address in the SPI Master's MOSI data frame.

This port expander is designed for a 16 bit SPI frame, with the MSB being sent first. The first 3-bits of the 16 bit MOSI frame signifies the address of the UART to be expanded into. The next 2-bits signifies the mode of the UART as follows:

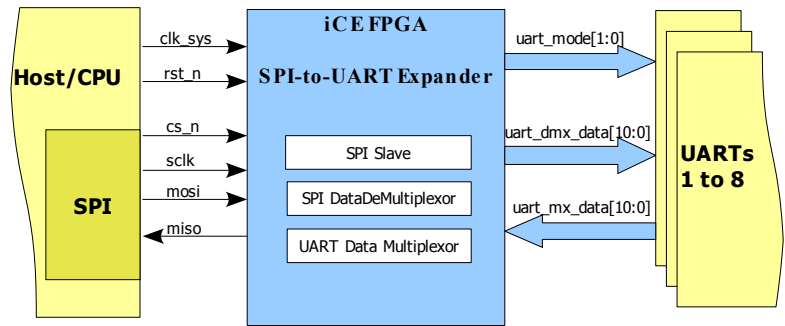
- when "00" and cs\_n = '0' then reads UART status register
- when "01" and cs\_n = '0' then reads UART data register/FIFO
- when "10" and cs\_n = '0' then writes UART data register/FIFO
- when "11" and cs\_n = '0' then writes UART configuration register

The balance 8 bits in the MOSI frame contain the data byte to be written to the Slave's output UART device. The first 6 bits on the MISO data frame are insignificant and are ignored. The following 10 bits contain the input data byte that the Slave received on a selected UART port.

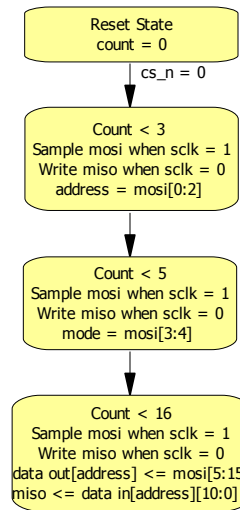
The data format of the SPI 16 bit word and the significance of its bit positions are indicated in Fig 3 in the following page.

Table 1 below lists interface details of this Design Example. Table 2 below shows the post P&R resource utilization summary of this design when implemented on an iCE65 FPGA.

Table 3 below shows the post P&R resource utilization summary of this design when implemented on an iCE65 FPGA.



**Fig 1: SPI to UART Interface**



**Fig 2: Simplified State Diagram**

**MOSI Frame – Config Write (M1M0: 11)**

DIN15	DIN14	DIN13	DIN12	DIN11	DIN10	DIN9	DIN8	DIN7	DIN6	DIN5	DIN4	DIN3	DIN2	DIN1	DIN0
A2	A1	A0	M1	M0	X	X	X	X	PT	PE	L1	L0	B2	B1	B0

**MOSI Frame – Data Write (M1M0: 10)**

DIN15	DIN14	DIN13	DIN12	DIN11	DIN10	DIN9	DIN8	DIN7	DIN6	DIN5	DIN4	DIN3	DIN2	DIN1	DIN0
A2	A1	A0	M1	M0	X	X	X	D7	D6	D5	D4	D3	D2	D1	D0

**MISO Frame – Data Read (M1M0: 00)**

DO15	DO14	DO13	DO12	DO11	DO10	DO9	DO8	DO7	DO6	DO5	DO4	DO3	DO2	DO1	DO0
X	X	X	X	X	X	PER	BER	D7	D6	D5	D4	D3	D2	D1	D0

**MISO Frame – Status Read (M1M0: 01)**

DO15	DO14	DO13	DO12	DO11	DO10	DO9	DO8	DO7	DO6	DO5	DO4	DO3	DO2	DO1	DO0
X	X	X	X	X	CTS	RTS	RXBSY	TXBSY	PER	BER	WFULL	RFULL	PE	L1	L0

**Fig 2: SPI Data frame**

Pin	Direction	Description
clk_sys	Input	System Clock
rst_n	Input	Active low system reset
cs_n	Input	Active low chip select
sclk	input	SPI serial clock
mosi	input	Master Out Slave In, data line from SPI master
miso	output	Master In Slave out, Data line from SPI Slave (This interface)
uart_mode[1:0]	output	Mode selection signals to UART
uart_dmx_data[10:0]	output	Data/Status lines from UART
uart_mx_data[10:0]	input	Configuration/Data lines to UART

**Table 2: Pin Description**

Device	Logic Cells	IO Cells
iCE65L04-UCB284	148	102

**Table 3.1: Resource Utilization for SPI to 4 UART Interface (SPI to 4 UART interface only, without UARTs)**

Device	Logic Cells	IO Cells
iCE65L04-UCB284	3169	38

**Table 3.2: Resource Utilization for SPI to 8 UARTs (SPI to UART interface + 8 UARTs)**

## Conclusion

This design example demonstrates the implementation of a SPI to UART interface using iCE FPGAs. iCE FPGA's very low power capabilities makes iCE FPGAs an obvious choice for implementing a SPI to UART interface for battery operated compact and handheld devices like PDAs, cellular phones etc..

## Disclaimer

SiliconBlue is providing this document, design example, or information "as is." SiliconBlue does not ensure that this implementation or information is void of any claims of infringement. As a reader / implementer, you are responsible for obtaining any rights you may require for your implementation. SiliconBlue also does not warranty the fitness of this design example to be readily implemented in any specific context. It is your sole responsibility to validate this design example and its correctness while implementing it for your requirement.