IJEEE

International Journal of Ethics in Engineering & Management Education

Website: www.ijeee.in (ISSN: 2348-4748, Volume 2, Issue 6, June 2015)

FPGA Implementation of Serial Protocol Bridge using SPI and I2C

Saniya Farheen
Dept of VLSI And Embedded System
VTU RO Kalaburagi
Kalaburagi, India

Dr Baswaraj Gadgay
PG Coordinator
VTU RO Kalaburagi
Kalaburagi, India

***Abstract*: For the frequent communication between the integrated circuits in an electronic system, which is designed to work in a group rather than a standalone unit, it becomes necessary to adapt serial communication, which is an effective and simple protocol, that provides efficient communication among these various components as compared to parallel communication in terms of the pin count and the ease of implementation. In today's world, which comprises of multiple application based products , there is a huge demand of multiple devices connected to the system that follow different communication protocols, paving way for the need of an intermediate system that can act as a bridge between two different devices that work on different communication protocols . The two most widely accepted, tried and true serial global standards are SPI and I2C, which are used for inter-chip and inter-chip serial communication for a low or medium bandwidth. The Project focuses on the detailed discussion of the above mentioned serial protocols, a bridge between them and their implementation on FPGA.**

1. INTRODUCTION

Today, embedded electronics is all about interlinking of various processors and integrated circuits in order to create a system which is completely symbiotic. These individual components need to share a common communication protocol so that they can swap their information for the data exchange. In order to achieve this data exchange, hundreds of communication protocols have been defined which are broadly classified into serial and parallel.

Even though the parallel communication has advantage of being fast and easy to implement, the number of i/o pins required for this communication is more than the serial communication, which are very precious and few on the microcontroller. Thus serial communication is preferred over parallel communication while sacrificing the potential speed for real estate.

Serial peripheral interface (SPI) and inter integrated circuit (I2C) are the two standard synchronous serial protocols that are widely used and accepted for serial communication between the devices.

In today's world, embedded systems are associated with a single, standard propriety protocol to enable serial communication. Different devices on the embedded system operate with different protocols (SPI and I2C) based on their requirement. Thus in order to achieve interoperability and flexibility between various devices, we need to devise a protocol converter, which is a SPI to I2C interface or I2C to SPI interface.

1.1 BASICS OF SPI

Serial peripheral interface is a hardware\firmware communication protocol, which also is commonly known as 'Four Wire Serial Bus'. The credit of designing this serial protocol lies in the hand of Motorola. SPI was primarily developed to enable reliable communication between the microcontroller/host processor and the peripheral devices. Later on the communication between two processors was also made possible through SPI.

It possesses the advantages of simplicity, high transmission speed and less number of pins for the interface. Since it supports full duplex communication and is a synchronous type, a data link can be set up with a master/slave interface that can support a speed up to 10Mbps.

SPI can support both single master and multi master protocol for the purpose of communication. But multi master protocol is rarely used due to its limitation to a single slave.

The serial peripheral interface is used to transfer data at a very high speed between various IC chips, Due to this reason, the SPI bus lines cannot be made too long, since it gives rise to increased reactance, resulting in an unstable bus. This makes the spi bus to be mostly used on the PCB only. Although it is possible to use the bus outside the PCB at low speeds, the effect does not seem practical.

The different data and control lines that are associated with the SPI bus protocol are as follows

- ✓ **MOSI**: Master out slave in. Here the signals are generated by the master and the slave is the recipient.
- ✓ **MISO**: Master in slave out. Here the signals are generated by the slave and are received by master.
- ✓ **SCLK OR SCK:** Serial Clock. This control signal is used to synchronise the transfer of data between master and slave and is always generated by the master

SS-slave select .it is used to select individual slave\peripheral device. It is an active low signal which is also generated by master

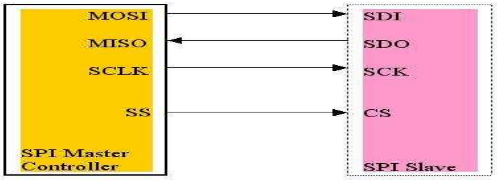


Fig 1 shows the implementation of SPI bus protocol with a single master single slave configuration and their related pins.

1.2 BASICS OF I2C

Inter integrated circuit bus was originally developed by Philips for the purpose of communication between various devices within a TV set. I2c devices are designed to operate at speeds up to 400 Kbps, with a few venturing up into the range of low megahertz.I2c is used to provide good and proper support in terms of communication with various on board peripherals that are intermittently accessed. It is basically a short distance protocol which is simple and has low bandwidth.I2c is termed as a two wire serial bus which has a built in addressing scheme that provides the ease to link multiple devices together. It eliminates the need of chip select or arbitration logic. This advantage of i2c makes the protocol cheap and easily implementable in hardware.SDA, serial data bus and SCL, serial clock are the two signals that support the i2c bus protocol. Both these signals allow the transmission of serial data comprising of 8 bit bytes and 7 bit device address plus control bits.

Initiation of a transaction on the i2c bus is done by the master, which usually controls the clock signal. The peripheral device, which is being addressed by the master, is termed as a slave. I2c communication protocol supports multiple masters, multiple slave configurations, wherein both the master and slave support the transmission and reception of data bytes.I2c bus protocol supports clock stretching in which the i2c slave holds off the master during the middle of the transaction. This is done by pulling the SCL line low until the slave is ready to continue. Even though the i2c slave devices do not use this feature quite often, every master should and must support it. There is a predefined device address associated with each i2c compatible hardware slave device and the lower bits of the address can be made configurable at the board level.

At the beginning of every transaction, the device address of the intended slave is transmitted by the master. It is the responsibility of the slave to monitor the bus and respond only to its own address. In order to avoid contention problem, the number of identical slave devices that can exist on the i2c bus are limited by means of the addressing scheme and this limit is determined by the number of user configurable address bits. Normally there are two bits, but even up to 4 identical devices are allowed.

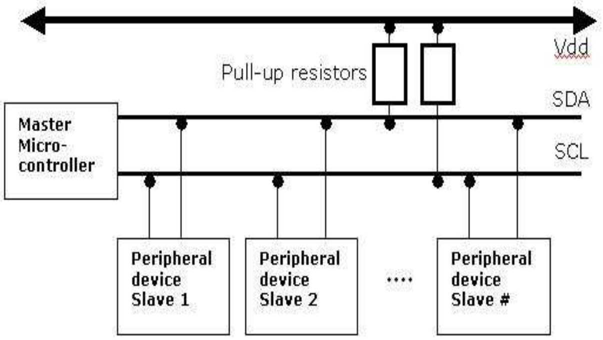


Fig 2 shows the basic block diagram of the I2C bus protocol.

2. DESIGN METHODOLOGY

I2C MASTER

The state diagram of the I2C master is shown in the fig below.

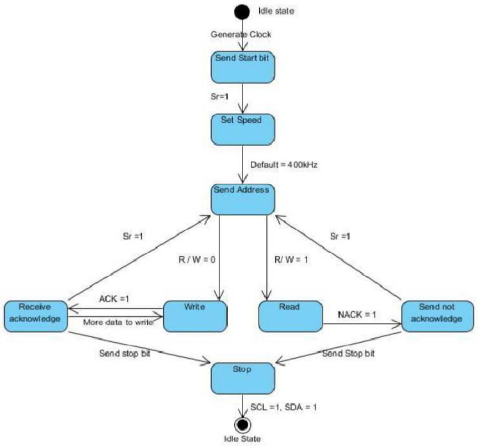
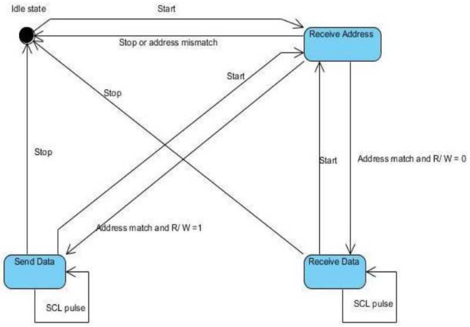


Figure 3 I2C Master



I2C SLAVE

The figure below shows the state machine diagram of the operation of I2C Slave

SPI SLAVE

The state diagram of the SPI slave is shown in the fig below

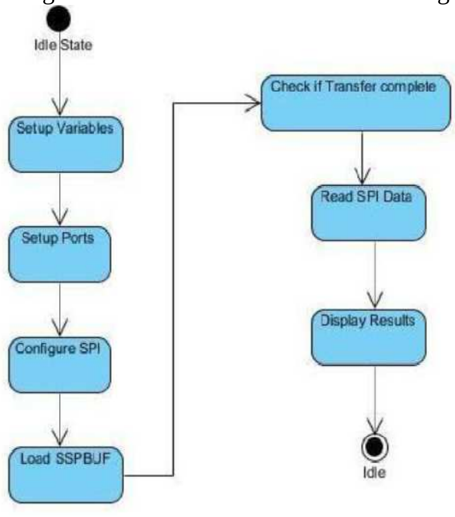


Figure 5 SPI Slave

SPI MASTER

The state diagram of the SPI master is shown in the fig below

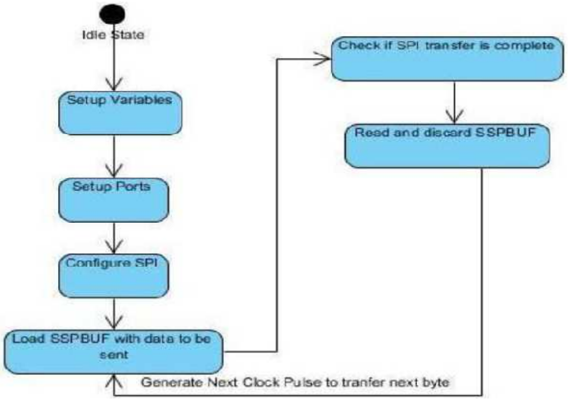

Idle State
Setup Variables
Setup Ports
Configure SPI
Load SSPBUF with data to be sent
Check if SPI transfer is complete
Read and discard SSPBUF
Generate Next Clock Pulse to tranfer next byte

Figure 5 SPI master

UNIVERSAL BRIDGE

For the communication between the two defined protocols seamlessly, we can implement a bridge which comprises of the entire four components spi master, spi slave, i2c master and i2c slave, thus enabling the complete communication between all the devices. The bridge is a universal bridge, which is shown in the figure below

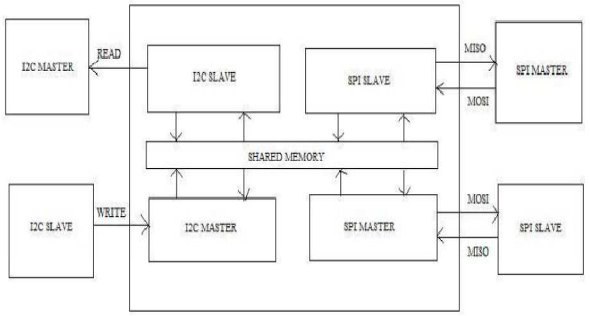


Figure 7 Universal Bridge

3. SIMULATION RESULTS

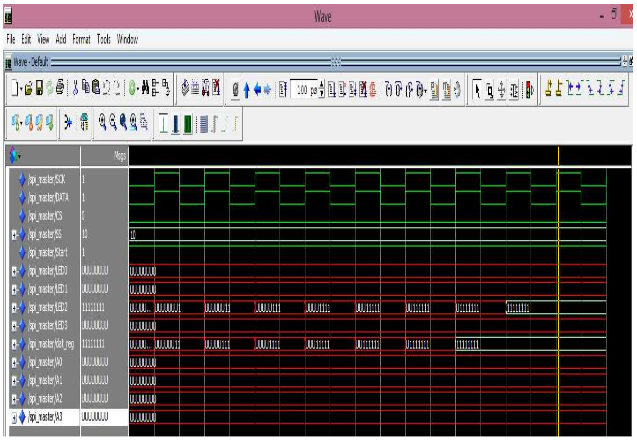


figure 8:simulation result of spi master

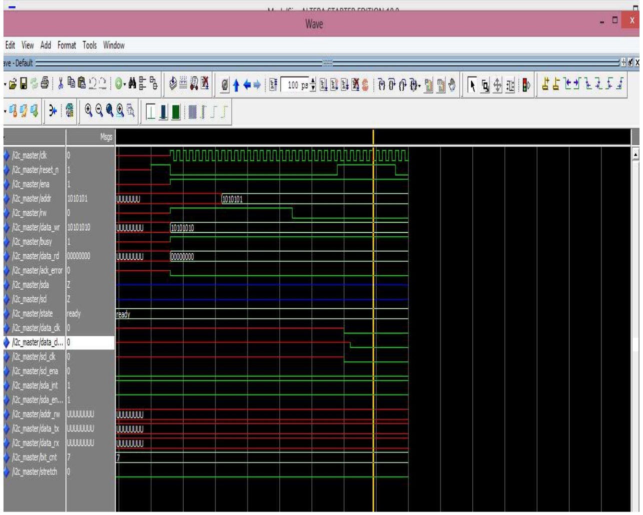


Figure 9:simulation result of I2C master

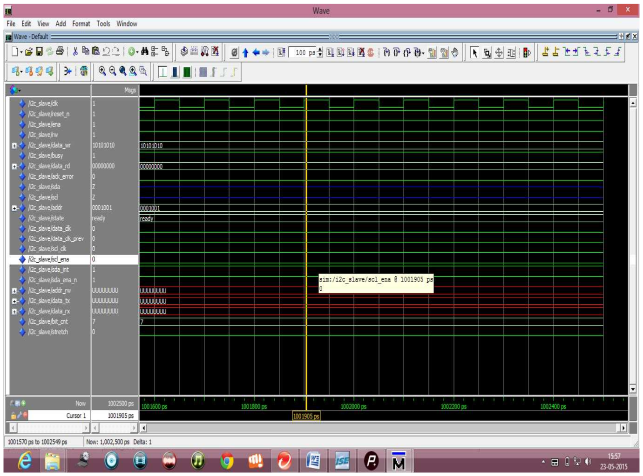


figure 10 :simulation result of i2c slave

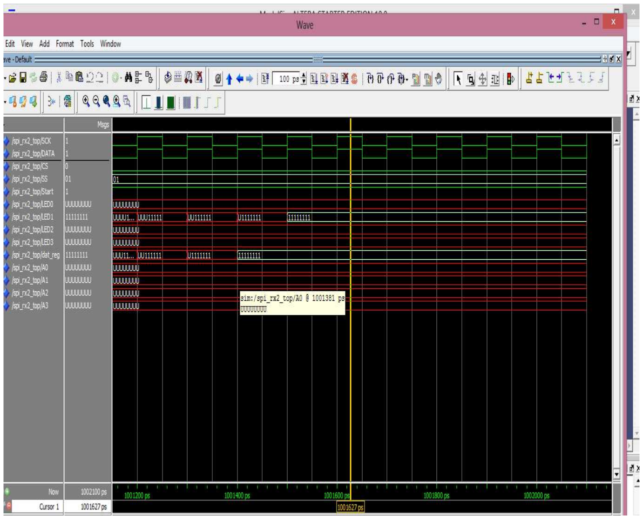


figure 11:simulation result of SPI slave

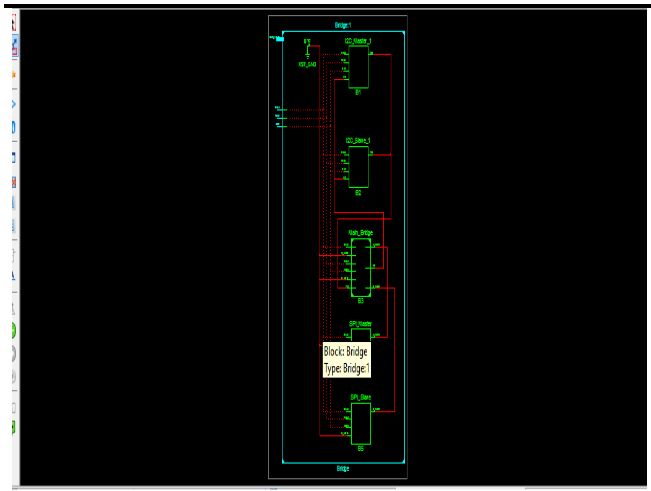


Figure 12: Complete RTL of Bridge

CONCLUSION

In the paper we design and implement an SPI I2C protocol bridge on an FPGA. All the peripherals within an embedded system use one of the serial protocols that work in master slave configuration. The bridge designed is capable of making seamless communication between the devices of different protocols as the master of one protocol can access the slave of another protocol via the serial protocol bridge.

Depending on the requirement, the bridge is implemented in different ways. Waveforms are generated through test benches for each mode. By using the bridge, all of the peripheral devices need to be slave to only one single protocol which reduces the number of wires and thus the overall size of the device.

REFERENCES

[1]. Abhishek Godara, Suprita Choudhury, "Implementation of I2C-DMA & SPI-DMA Interface: A Comparative Study" International Journal of VLSI and Signal Processing Applications, June 2014

[2]. Eswari, B. ; VLSI design group, Nat. Inst. of Electron. & Inf. Technol., Calicut, India ; Ponmagal, N. ; Preethi, K. ; Sreejeesh, S.G. "Implementation of I2C master bus controller on FPGA" Communications and Signal Processing (ICCSP), 2013 International Conference on 3-5 April 2013

[3]. Abhishek Godara, Suprita Chaudhary, Power consumption reduction using RTL Clock Gating in AHB-Slave SPI-Master Architecture" International Journal of VLSI and Signal Processing Applications, 2012.

[4]. Jai Karan Singh, Mukesh Tiwari, Vishal Sharma, "Design and Implementation of I2C to SPI Converter on FPGA Using VHDL," IJEECE Vol. 2, pp. 499-507, July 2012

[5]. K. Aditya, M. Sivakumar, Fazal Noorbasha, T. Praveen Blessington, "Design and Functional Verification of A SPI Master Slave Core Using System Verilog," IJSCE 2012, pp. 558-563, May 2012.

[6]. Zheng-wei HU, "I2C Protocol Design for Reusability", Third International Symposium on Information Processing, 978-0-7695-4261-4/10 © 2010 IEEE.

[7]. K. Oudjida, M. L. Berrandjia, R. Tiar, A. Liacha, K. Tahraoui, "FPGA Implementation of I2C & SPI Protocols: A Comparative Study," ICECS 2009, 16th IEEE International Conference, pp. 507-511, December 2009