FPGA Implementation of I2C and SPI Protocols using VHDL

Satish M Ghuse[1], **Prof. Surendra K. Waghmare**[2]
[1, 2] Department of ENTC
[1, 2] SPPU/G.H.Raisoni College of Engineering and Management, Pune, Maharashtra/Zone, India

Abstract- *I2C and SPI are the serial communication protocols that are commonly used for both intra-chip and inter-chip low/medium bandwidth data transfer. It can support bi-directional data transfers at up to 100 Kbit/s in the standard-mode, up to 400 Kbit/s in the Fast-mode, up to 1 M bit/s in the Fast-mode plus, or up to 3.4 M bit/s in the High-speed mode. These protocols have a preferable speed and power consumption capability when implemented with different devices but their speed is low, when used with BIST or checksums.*

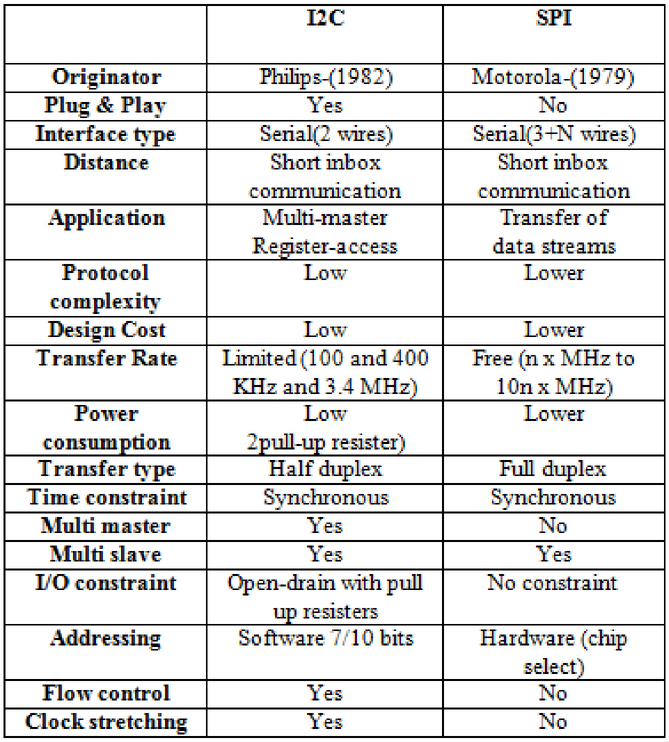
In the earlier systems speed and delay was not taken into the consideration and the protocol was implemented as it is in the standard mode. We are about to implement the I2C and SPI protocols efficiently so that the speed of the data transfer increases and there delay is reduced. In order to do so we have proposed a design in which we are using pipelined buffer in between Master and Slave so that it will reduce the delay and there would be synchronization and increase in the speed of data transfer as the delay in data transfer decreases. The pipelined buffer results in the speed enhancement for the critical paths. Using this pipelined buffer the data transferred from master is first stored in buffer and then transferred to the slave and vice-versa. The delay is in the form of nanoseconds in which when used pipelined buffer in the system route delay is neglected and only logic delay is considered. The speed is calculated using the delay of the system in the form of Mbps or Gbps depending on the delay. The delay report for both the protocols are taken from synthesis report generated in the Xilinx software and the simulation of the system is done in the model-sim software.

Keywords- Xilinx Software 14.5, Model Sim Software, I2C Serial communication Protocol, SPI Serial Communication Protocol, Pipelined Buffer

I. INTRODUCTION

Today, at the low end of the communication protocols we can find two worldwide standards: Inter integrated circuits (I2C) and serial peripheral interface (SPI). Both protocols are well specified for communication between integrated circuits for low or medium data transfer speed with onboard peripherals. The two protocols co-exist in modern digital electronics systems, and they might probably will continue to compete in the future, I2C and SPI both are actually complementary for this kind of communication. The I2C and SPI protocol specifications are well defined. Consequently, they will not be discussed here. But, a quick overview is provided in table 1.

Although the literature of I2C and SPI is old (early 1980), to the best of the writers knowledge there is no comprehensive comparison of I2C and SPI problem. By comprehensive comparison we mean a treatment that start from Philips and Motorola specifications and goes down to the actual ASIC or FPGA implementation, linguistics the two designs and then considering similarities of the obtained results based on included protocol features.



	I2C	SPI
Originator	Philips-(1982)	Motorola-(1979)
Plug & Play	Yes	No
Interface type	Serial(2 wires)	Serial(3+N wires)
Distance	Short inbox communication	Short inbox communication
Application	Multi-master Register-access	Transfer of data streams
Protocol complexity	Low	Lower
Design Cost	Low	Lower
Transfer Rate	Limited (100 and 400 KHz and 3.4 MHz)	Free (n x MHz to 10n x MHz)
Power consumption	Low 2pull-up resister)	Lower
Transfer type	Half duplex	Full duplex
Time constraint	Synchronous	Synchronous
Multi master	Yes	No
Multi slave	Yes	Yes
I/O constraint	Open-drain with pull up resisters	No constraint
Addressing	Software 7/10 bits	Hardware (chip select)
Flow control	Yes	No
Clock stretching	Yes	No

There are many software applications developed in the implementation of the communication protocols SPI and I2C. In general, these researches are focused on the comparisons and implementations of different architectures in order to meet characteristics required by current technologies.

In 2006, Oudjida et al. developed a code to implement to medium or low speed a transmitter slave I2C in a VLSI-architecture that allowed meeting some specific terms that were not implemented in other designs such as drive noise filtering, a data unit, a unit equipment side interface, a control unit, etc.

More recently, in 2009 Oudjida et al., present an implementation of the SPI and I2C protocols in different FPGA devices, help designers to choose the right architecture for their systems. To do this, they designed the code in Verilog HDL (according to each protocol) for the slave SPI and I2C to the different FPGA devices, considering similarities their functionality in response times and clock settings, concluding that logic can predict certain behaviors for master devices from the results of the slaves[2].

Then in 2011 Lazaro et al.[1], presented a new design in Verilog I2C protocol, focusing on the security of the electronic communication devices, integrating AES-GCM authentication and encryption algorithms. To do this, author adapted the features of the I2C protocol with verification techniques and secret data, comparing with the final design with original protocol, and they concluded that their work reduces the memory of data flow and it is easily implemented in FPGA[3].

Zhou et al., developed a verification environment considering many different and connected parts of electronic systems from the master SPI interface, and integrate Verilog with object-oriented programming (OOP). To achieve this, they started with the functional description of the requirements for the master SPI and environment design, and they implemented the APB controller in OOP classes[4]. The SPI Master interface was developed and implemented in FPGA Verilog. Finally, it is important to take into account that in the above mentioned works, the design makers have used the method of hardware description (Verilog HDL), which helps to implement and to model the concurrent behavior of the electronic embedded devices, especially when it is a new architecture design.

Software Radio technique is mainly based on recent communication theory, a wireless device with configurable hardware platform, which can go across a variety of communication standards[5]. The basic principle is to make sure the bandwidth close to the antenna as much as possible, to use the software instead of hardware for signal processing. As lower cost, greater susceptible of modification or adaptation and higher performance, it was been widely used in many areas.TD-LTE communication system also uses the software radio technology.

As required higher performance of TD-LTE communication system, the hardware requirements are getting higher and higher. TD-LTE system of base and signal includes the decoding operations and FFT transform complex algorithms, bringing in a huge amount of computation. As far as the current processing rate of hardware is concerned, it is difficult to use Digital Signal Processing or FPGA to finish the task separately. Based on the systematic evaluation, we propose a new hardware processing platform based on DSP,FPGA and ARM. In this way, the DSP, FPGA and ARM can play to their strengths, work together and achieve great efficiency. The TD-LTE Comprehensive Test Instrument is just based on the hardware platform[6].

SPI or Serial-Peripheral Interface is a worldwide accepted standard communication protocol. SPI protocol was invented by Motorola. SPI protocol is considered as one of the very best among the systems that are connected to a number of devices and make the communication smooth and fast. SPI as well as others serial protocols such as I2C and 1-wire for urgency are well fitted for data communications from integrated circuits for low or medium data transfer speed to peripherals which are on chip board.

Several works have been done using VHDL in designing SPI. A comparison between SPI and I2C Implementation over FPGA is shown in TABLE 1. On that paper, a comparative study of those two protocols on FPGA platform is presented and the entire design has been coded in VHDL. For various controlling purposes SPI is implemented. This is new approach of designing SPI with embedded BIST capability using Field Programmable Gate Array (FPGA) technology. Testing of a circuit has become increasingly tough as the scale of integration grows.

SPI with the BIST capability provides the specified testability requisites and lowest-price with the highest performance implementation. Much lesser blocks and modules are used to design this SPI so that the testing complexity can be reduced. This system can be fabricated into a single chip.

II. PROPOSED SYSTEM OF I2C PROTOCOL

The proposed system of I2C protocol consist of the master and slave in whom we have introduced a pipelined buffer in the bus SDA and SCL. The 8 bit address is used as an input to the system. The buffer stores the data transferred from the master or from slave to master. Due to which there is no data lose in between the data transfer. As the delay of the system gates reduced the speed of the data transfer or Transfer rate of the system increase.

Below is the proposed design for I2C Protocol.

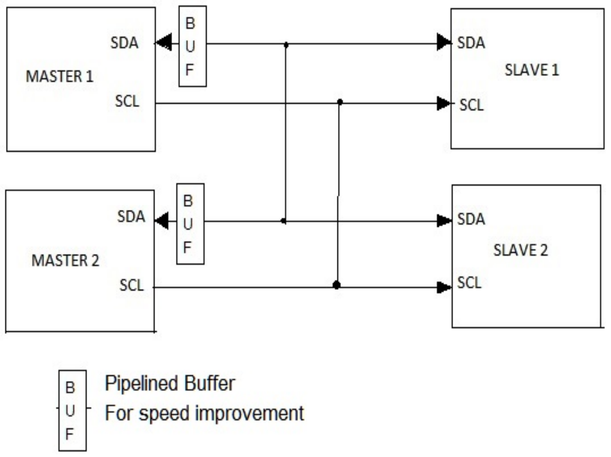


Fig 1. Proposed design for I2C Protocol.

III. PROPOSED SYSTEM OF SPI PROTOCOL

The proposed system of SPI protocol consist of the master and slave in whom we have introduced a pipelined buffer in the bus MOSI and MISO. The 1 bit address is used as an input to the system. The buffer stores the data transferred from the master .Due to which there is no data lose in between the data transfer. As the delay of the system gates reduced the speed of the data transfer or Transfer rate of the system increase.

Below is the proposed design for SPI Protocol.

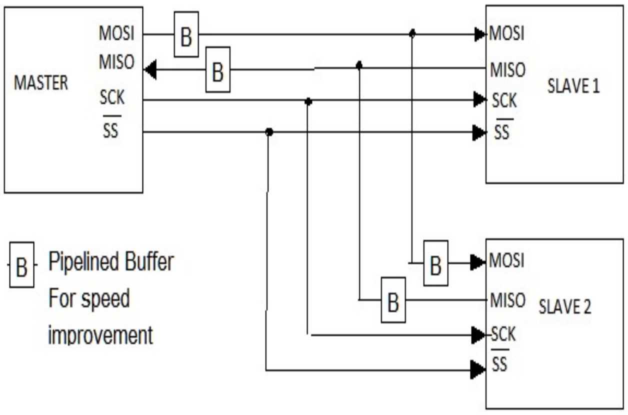


Fig 3.Proposed design for SPI Protocol.

IV. SIMULATION RESULT FOR STANDARD I2C PROTOCOL

Below is the Simulation result of the standard I2C Protocol it comprise of Start cycle, Read cycle, write cycle and reset.

TABLE 2: Signal Description

Signal	Type	Description
SDA	In-Out	I2C serial data
SCL	In-Out	I2C serial CLOCK
RESET	INPUT	RESET
RX (7:0)	OUTPUT	OUTPUT data
TX (7:0)	INPUT	INPUT data

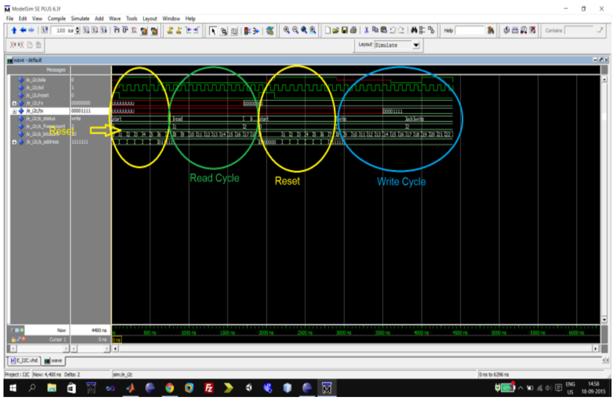


Fig 4.Simulation Result of Standard I2C Protocol.

4.1 Simulation Result for I2C Protocol with Pipelined Buffer

The below is the simulation result of the I2C protocol which comprises of the pipelined buffer. In this result will are able to see the transmitted address getting fully received and also some data stored in buffer**.**

Table 3: Signal Description for I2C Protocol with Pipelined Buffer

Signal	Type	Description
SDA	Buffer	I2C serial data
SCL	In-Out	I2C serial CLOCK
RESET	INPUT	RESET
RX_data (7:0)	OUTPUT	OUTPUT data
TX_data (7:0)	INPUT	INPUT data

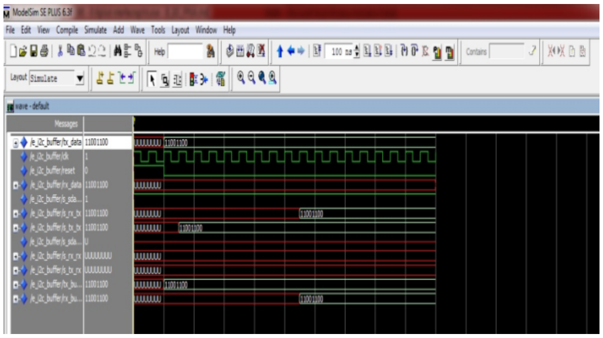

ModelSim SE PLUS 6.3f
File Edit View Compile Simulate Add Wave Tools Layout Window Help
Layout Simulate
wave - default
Messages

Fig:4.1. Simulation Result for I2C Protocol with Pipelined Buffer

4.2 Comparison

Below table shows the comparison between standard I2C Protocol Delay and Speed with the proposed Design of the system containing Pipelined buffer.

Table 4: Comparison

I2C Protocol	Delay	Speed(Transfer Rate)
Standard	5.535ns	186MHz
With Pipelined Buffer	1.718ns	582MHz

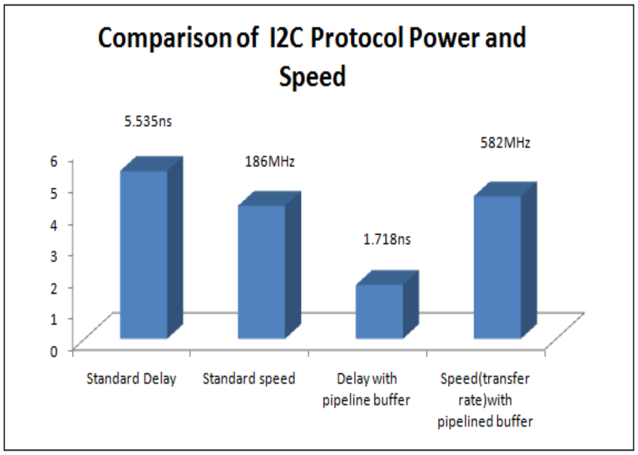


Fig4.2 Graphical Representation I2C Delay and Speed of System Containing Pipelined Buffer.

V. SIMULATION RESULT FOR STANDARD SPI PROTOCOL

Below is the Simulation result of the standard SPI Protocol.

TABLE 5: Signal Description for SPI

Signal	Type	Description
In_data	INPUT	INPUT data
RESET	INPUT	RESET
MISO	INPUT	INPUT data
MOSI	OUTPUT	OUTPUT data
CLK	INPUT	INPUT CLOCK
ENABLE	INPUT	INPUT data
BUSY	OUTPUT	OUTPUT data

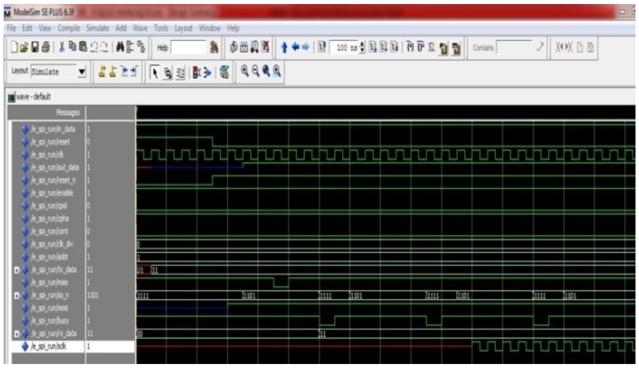


Fig 5 Simulation Result FOR Standard SPI Protocol

5.1 Simulation Result for SPI Protocol with Pipelined Buffer

Below is the Simulation result of the SPI Protocol with Pipelined Buffer.

TABLE 6: Signal Description for SPI with Pipelined Buffer

Signal	Type	Description
In data	INPUT	INPUT data
RESET	INPUT	RESET
MISO	INPUT	INPUT data
MOSI	OUTPUT	OUTPUT data
CLK	INPUT	INPUT CLOCK
ENABLE	INPUT	INPUT data
BUSY	OUTPUT	OUTPUT data
TX Buffer	INPUT	INPUT data
RX Buffer	OUTPUT	OUTPUT data

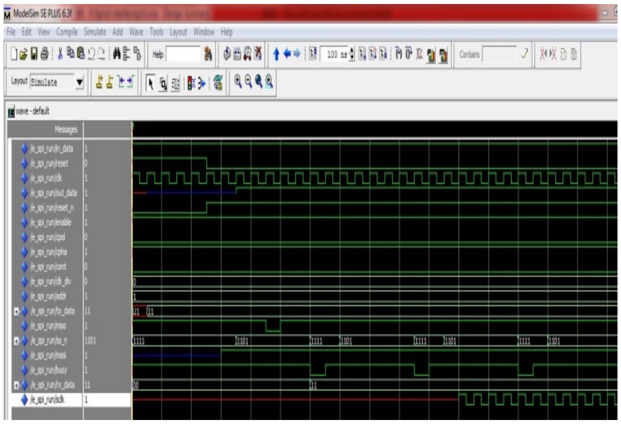


Fig 6

5.2 Comparison

Below table shows the comparison between standard SPI Protocol Delay and Speed with the proposed Design of the system containing Pipelined buffer.

Table 8: Comparison

I2C Protocol	Delay	Speed(Transfer Rate)
Standard	6.216ns	160MHz
With Pipelined Buffer	5.535ns	186MHz

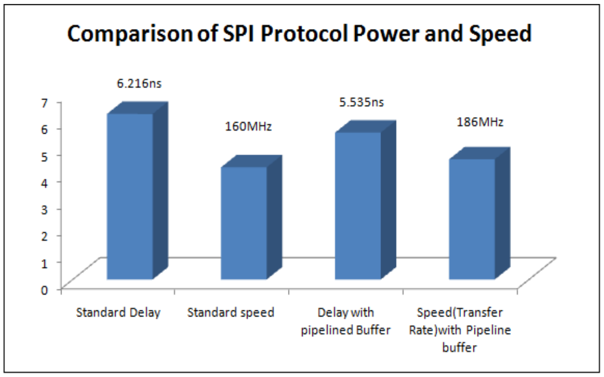


Fig 8.Graphical Representation of Delay and Speed with the proposed Design of the system containing Pipelined buffer.

VI. CONCLUSIONS

The results of the simulations and from the comparison tables we can conclude that the above system works as per the requirement. The decrease delays and increase in transfer rate can be seen from comparison and as well as the synthesis report illustrates the systems requirements more clearly. The design of both the protocols using VHDL simplifies the design process. If a new technology emerges, designers do not need to redesign the circuit. He simply input the design program to the logic synthesis tool and creates a new gate level netlist using the new fabrication technology. The logic synthesis tool will optimize the circuit in area and timing for the new technology. Through this implementation we can conclude that the system can be used for serial communication for faster transfer of data. This also takes less delay because of which the speed of the I2C and SPI protocols has been increased.

ACKNOWLEDGMENTS

I would like to take this opportunity to thank my guide Prof. SurendraK. Waghmare GHRCEM, Pune for his valuable guidance for this project. This project could not be completed in time but due to the sincere hard work of the teaching staff. Special Thanks to our Teaching, non-teaching staff, friends and parents who helped us all the time and in every way for this successful attempt and completion of this project.

I am equally indebted to Prof. Vijay Kumar Joshi, M. E. Coordinator Electronics and Telecommunication Department, for necessary help, providing, facilities and time to time valuable guidance. No word be good enough to express my deep gratitude to our respected principal Dr. R. S. Bichkar for his kind blessings, inspiration and necessary support whenever needed.

REFERENCES

[1] Chen Fatang, YangLinyu “The Design of I2C Bus in the TD-LTE Comprehensive test Instrument” 978-0-765-4464-9/11 2011 IEEE DOI10.1109/BCG-In.2011.165 662.

[2] A.K Oudjida, M.L Berrandjia, R. Tiar, A. Liacha, K. Tahraui “FPGA Implementation of I2C and SPI Protocols:A Comparative Study” 978-1-4244-5091-6/092009 IEEE

[3] Zheng-Wei HU in “I2C Protocol Design for Reusability” 978-0-7695-4261-4/10 2010 IEEE.

[4] Wasim Hussain, YvonSavaria, Yves Blaqui`ere in” An Interface for the I2C Protocol in theWaferBoardTM” 978-1-4673-5762-3/13/$31.00 ©2013 IEEE.

[5] Sindhu, Dr. Vijaya Prakash A M and Ankit K v, Proposed “ASIC Implementation of I2C Master Bus Controller Firm IP Core” International Journal of VLSI design and Communication Systems (VLSICS) Vol.6 No. 4, August 2015.

[6] Tatiana Leal-Del Rio, Gustavo Juarez-Gracia, L.Noe olive Moreno, “Implementation of communication protocols SPI and I2C using a FPGA by the HDL-Verilog Language” pp.31-41; rec. 2014-06-19; 2014-07-21. Research in computer Science 75 (2014)

Satish M. Ghuse received the B.E degree in Electronics Engineering from Om College of Engineering, Wardha in 2013 and pursuing M.E in Electronics and Telecommunication with Specialization in VLSI and Embedded Systems from G.H Raisoni College of Engineering and Management, Pune University.