

Data Transfer of Flash Commands Through Different Serial Peripheral Interface (SPI) Approaches

Laxmi

Dept. of Information Science and Engineering
SJB Institute of Technology
Bengaluru - 560060, Karnataka - India.

Dr. Yogish H K

Dept. of Information Science and Engineering
SJB Institute of Technology
Bengaluru - 560060, Karnataka - India

Abstract— This paper focuses on data transmission of flash instructions via Serial Peripheral Interface(SPI) bus. Flash memory is a type of storage which can be utilized even when the power is off. Examples of non-volatile memory are Read Only Memory (ROM), hard disks, floppy disks etc. In this paper, we are developing the validation suite application to test the characteristic of the flash memory. Here the program code is implemented from flash. The microcontroller used in the implementation has the features of advanced protection that includes boot ROM and supports encrypted software. SPI is used as an interface for data transmission. The flash supports three approaches. 1. NormalSPI approach. 2. Quad approach 3. Octal approach. Normal SPI approach uses two data lines for transmitting and receiving the data where as in Quad approach, it uses four data lines and in Octal approach it uses eight data lines transmission and receiving purpose.

Keywords—Flash memory, non volatile memory, encrypted software, NormalSPI, Quad, Octal.

I. INTRODUCTION

The Flash memory is a non volatile memory that is used to clear the data in units [1]. Before storing the data stored in the flash should be cleared first then it is programmed. If the data is not cleared it cannot be reprogrammed in the chip. The data remains in the flash for a long period of time even after the power is off [3]. Flash memory comes into picture from Erasable Programmable Read Only Memory(EPROM). Flash memory clears the information in block level and data can be rewritten at the byte level.

There are two types of flash memory: NAND flash and NOR flash[2]. NOR flash has a feature of Execute-in-Place and it also has the characteristics of high read efficiency. NOR memory is used in the implementation because it has lower standby power and it is very easy for code execution. NOR is having fast response than NAND [15]. Taking into consideration of NOR and NAND flash architecture, the cells in NOR flash are joined in parallel way to the bit lines. This way of position allows the cells to read and program the data separately. In NAND, the cells are joined in series which is similar to the NAND gate.

FPGA is an Integrated circuit (IC) that is designed by the customer after logic is mechanized [9] hence the name field programmable. Logic blocks are nothing but the memory elements which are used in FPGA. For ex: Flip-Flops,

complete blocks of memory [2]. Serial Peripheral Interface (SPI) is the synchronized serial communication bus that is used for diminutive distance communication purpose primarily in embedded systems. SPI bus was produced by Motorola Company. Serial Peripheral Interface devices are able to transfer the data in full duplex approach which uses Master – Slave architecture [12]. Out of I2C and SPI protocol we are using SPI protocol [11]. SPI bus describes the method of data transmission between two devices. SPI uses single master communication protocol. To achieve the data transfer between master and slave, there are four signal lines:

- A clock (CLK): A clock signal is sent from the master to all slaves. The signals are synchronized to the clock.
- Select signal (SS): A select signal for slave is used to select the master data transfer.
- Master out slave in (MOSI): Master to slaves data transfer.
- Master in slave out (MISO): Slaves to master data transfer.

II. LITERATURE SURVEY

Literature survey is the survey to examine the previous work that helps to find the flaws in the current system. This section deals with the former studies that have been granted in the field of flash memory and SPI data transfer protocol. Memory is class divided into primary and secondary memory. The paper deals with the flash [15]. The memory can be also be divided into permanent memory and temporary memory as shown in the Fig. 1.

Permanent memory is a memory which keeps its information even though the power is off. With the help of ROM, it is possible to read the information but cannot write on it [6]. Permanent memory is also called as non volatile memory and temporary memory is also called as volatile memory. The memory is saved perpetually during the manufacturing. Programmable Read Only Memory (PROM) is one that can be modified by the user only once. The user buys a blank PROM and enters the required information using a PROM. Erasable Programmable Read Only Memory (EPROM) is used to clear the information and reuse the program. Electrically Erasable Programmable Read-Only Memory (EEPROM) is utilized to programme and clear

electrically [5], it can be cleared and reprogrammed as many times the user or the customer wants. NOR and NAND are the kinds of flash memory [10]. These two memories are unique in their operation. Volatile or temporary memory includes RAM, Static RAM [7], Dynamic RAM. SPI is a general technology which is used for data transfer with peripheral devices so that the speed of data transfer increase. RAM is called as read or write memory, with the aid of RAM, the programmer or the developer can be able to read or write the data. One of the example of Static RAM (SRAM) is flip flop which utilizes the transistors. Dynamic RAM requires refreshing of memory elements. SRAM is having speed response with comparison to DRAM since SRAM utilizes less access time.

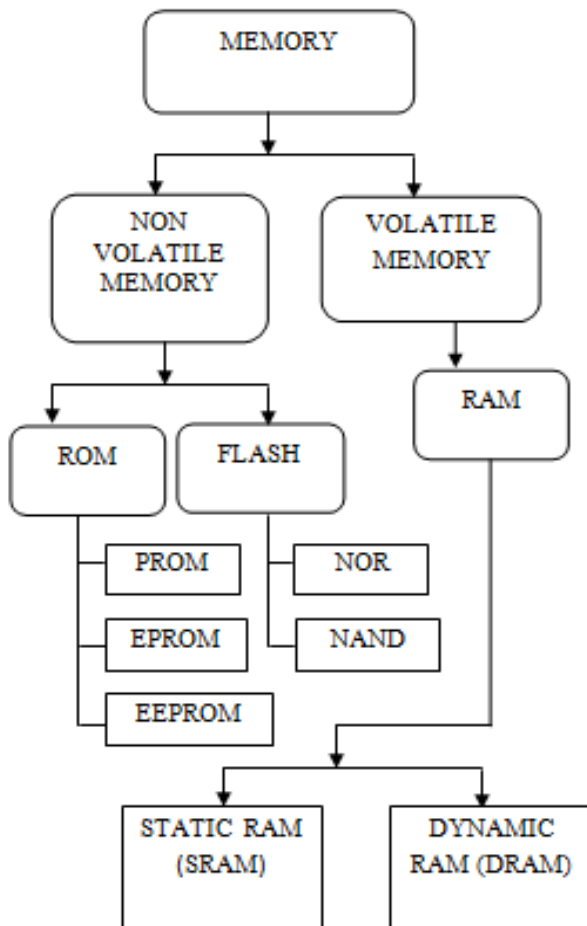


Fig. 1. Memory Classification diagram

The SPI protocol is a simple 4-wire serial interfacing approach which can be used for microcontroller or microprocessor peripheral integrated circuits that enables the controllers and peripheral devices to communicate with each other. In SPI, the data is moved in or out one at a time and transfer the information from master device to or from one or more slave devices with high speed.

III. PROBLEM STATEMENT AND OBJECTIVES

A. Problem Statement

“Problem Statement is to transfer the Flash commands from host to target via SPI bus”.

B. Objectives

- To develop the architecture for testing the application which will run on the K82 reference platform.
- To communicate between host and target that is achieved via SPI bus.

IV. IMPLEMENTATION

In this paper we are achieving the communication between the host and the target through SPI bus as Fig. 2, k82 reference platform acts as a host and FPGA acts as a target. We cannot connect the k82 reference platform(microcontroller) and FPGA [13]. There are extra switches needed to join both these two devices. The microcontroller uses 24 pin connector and FPGA uses interface connector. SPI bus is utilized to communicate between the microcontroller and FPGA [14] [15]. Three approaches of data transfer is possible: NormalSPI approach, Quad approach and Octal approach.

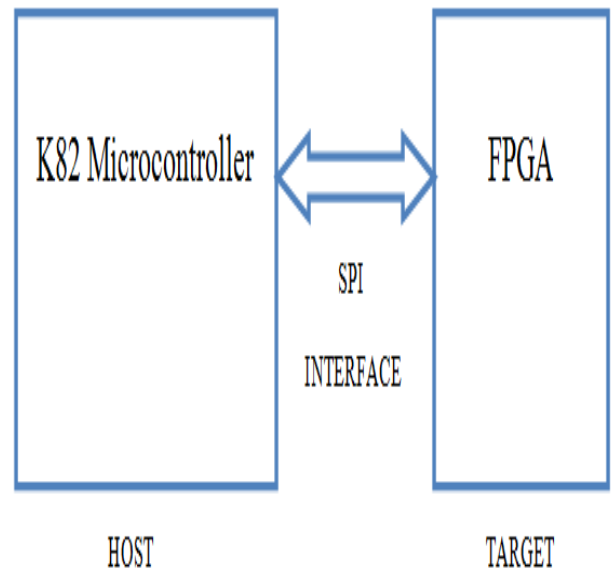


Fig. 2. Block Diagram of System Architecture

Normal SPI approach uses only two data lines for data transfer and receiving the information [8] .Quad approach uses four data lines whereas octal approach uses eight data. The actual logic for the flash is coded in host device and test logic is coded in target device. The memory of the flash is classified as follows:

- 16 page = 1Block
- 64Blocks = 1Sector
- 1Sector = 1024Pages

Implementation part includes four main classes. Normal SPI class, Quad class, octal class and Test Suite class for testing the application.

A. Normal SPI approach

A group of instructions are to be sent from host to target referred as SPI master. Host i.e SPI master communicates through four signal lines. Chip select, serial clock, data select, serial input and serial output.

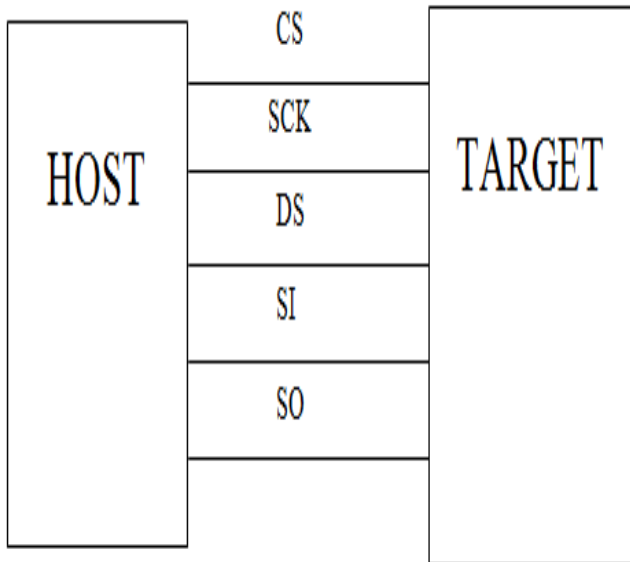


Fig. 3. Normal SPI approach

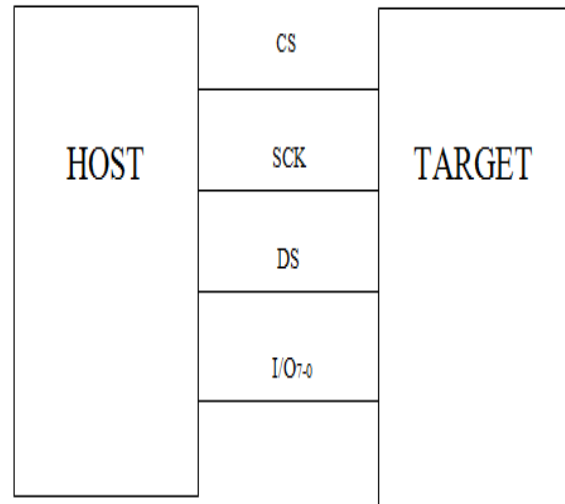


Fig. 5. Octal approach

B. Quad approach

This approach increases the throughput speed or performance. In this approach four data lines are there for transmission and receiving the data or information. Single transfer method and dual transfer method are supported by Quad approach.

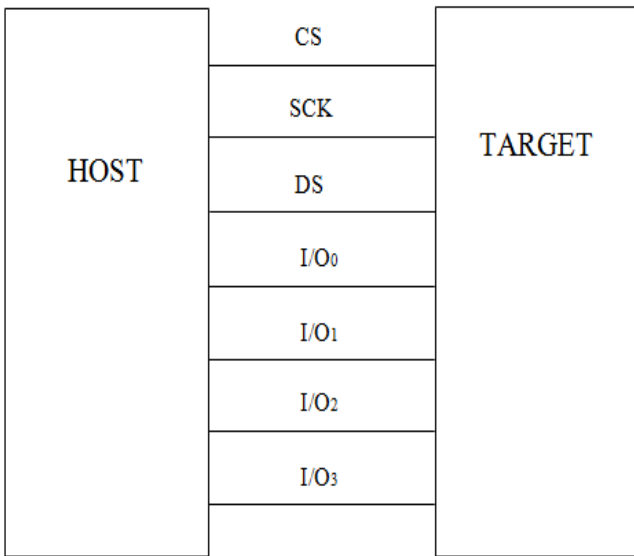


Fig. 4. Quad approach

C. Octal approach

Octal approach also supports single transfer method and it also supports dual transfer method. The octal approach is an extension for Normal SPI approach and Octal approach.

D. Test Suite for testing the application

The Test suite as shown in the Fig. 6 is deals with testing the application. It is the testing part of the paper with which we can obtain the outputs by testing “all the commands”, ”test individual test case” or “display all the test case” scenarios.

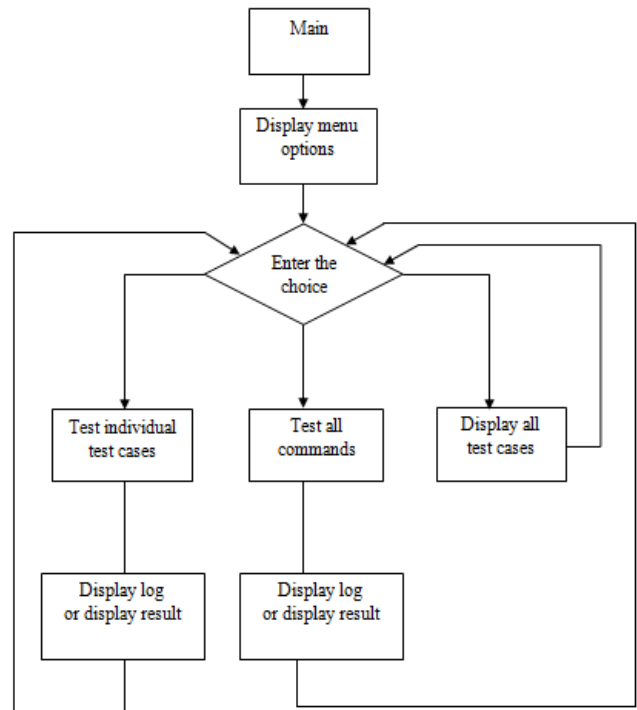


Fig. 6. Flow chart of Test Suite

E. Description of few instructions

1) *Read Array instruction:* The Read Array instruction is used to read serially the continuous stream of information by just giving the clock signal after initiating the starting address and also the device incorporates an internal address counter which repeatedly increases every clock cycle [8]. Read array of byte instruction has two operation codes “0Bh” and “03h”.

To carry out the Read Array operation, the chip select pin should be enabled and the appropriate operation codes 0Bh or 03h should be clocked into the device. After the operation code has been clocked in, three address bytes should be clocked in, just to recognize the starting address position of the first byte to read within the memory element.

2) *Erase instruction:* This command can be utilized to clear complete chip. The complete memory can be erased by using chip erase instruction [10]. The write enable command should be issued before using erase instruction. To trigger the Chip Erase instruction, the operation code which is related to erase instruction is sent to the device. On de-asserting the chip select pin, the erase instruction operation is completed.

3) *Sector Protection instruction:* Sector protection instruction helps in maintaining the software protection. After the device is reorganized the sector protection register sets the logic state to logical “1”. This instruction is mostly used for screening whether all the sectors are protected and so that we cannot able to clear the program once again. Before using sector protection instruction, the instruction write enable is executed. The operation begins with executing the chip select.

F. Some of the Function Prototypes

1) *ReadArray_SPIApproach:*

```
void ReadArray_SPIApproach (unsigned int ui32StartAddr, unsigned int
    len, unsigned char *ui8buff)
    {
        -----
        -----
        -----
    }
```

2) *BlockErase_OPIApproach:*

```
void BlockErase_OPIApproach (unsigned int ui32StartAddr)
    {
        /*Enable write execution before block erase*/
        -----
        -----
    }
```

V. PERFORMANCE EVALUATION

This section deals with the evaluation of experimental outputs of this paper using Vivado Tool. Vivado 2015 version 4 is used for the performance evaluation. Vivado2015.4 is a simulation tool which prints the outcomes through timing diagrams. Every instructions utilized in the implementation maintains the standard and particular timing diagrams as needed.

Three approaches such as Normal SPI approach, Quad approach and Octal approach can be compared as shown in the Fig.7, Fig.8 and Fig.9. Simulation of Read operation in all the three approach is measured for comparison. The read operation is faster in Octal approach as shown in Fig. 9, compared to other two approaches. As octal approach uses eight data lines for data transfer and receiving purpose. The clock cycles gets reduced so that the access time also reduces. As shown in Fig. 7, the Read operation in Normal SPI

approach uses two data lines and the clock cycle get increments with the transfer and receiving time. Consider the Fig. 8, the Read operation in quad approach utilizes four data lines for transfer of data and receiving, transfer time and receiving time is much faster than Normal SPI approach and slower than Octal approach.

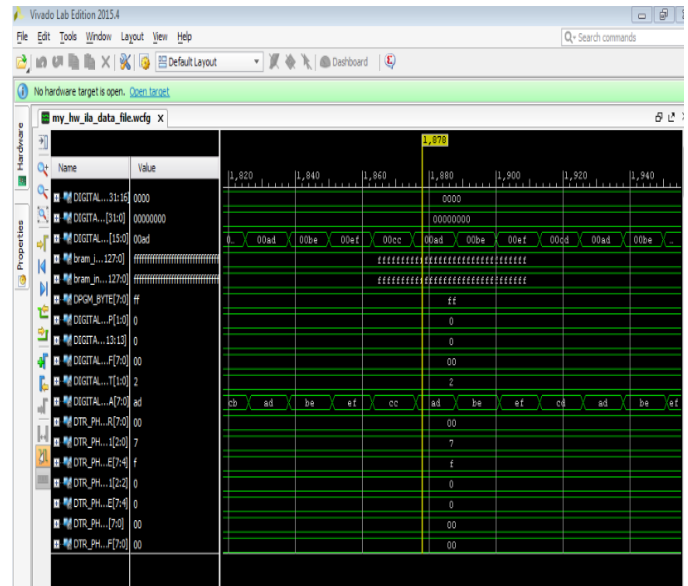


Fig.7.Simulation of Read operation in Normal SPI approach

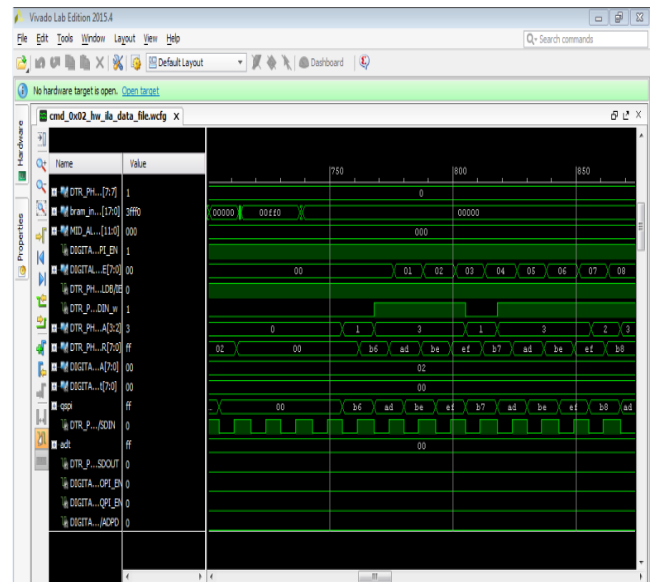


Fig. 8. Simulation of Read operation in Quad approach

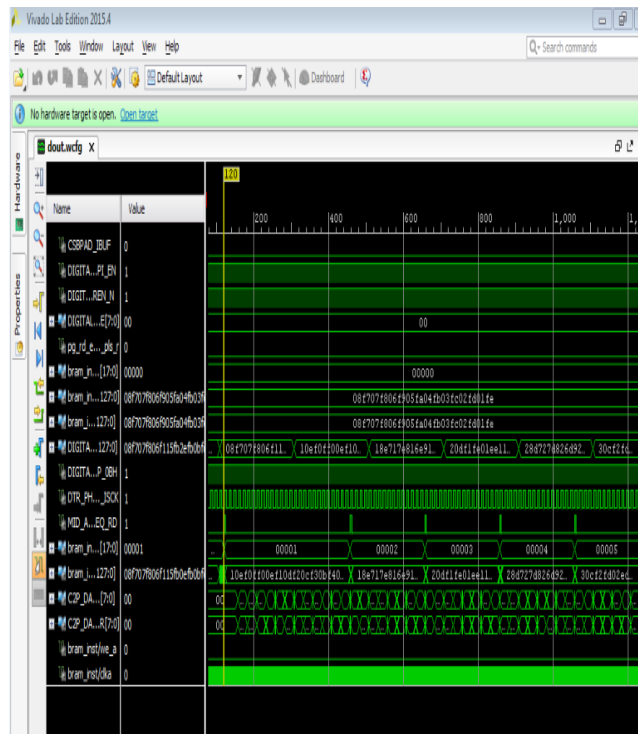


Fig. 9. Simulation of Read operation in Octal approach

VI. CONCLUSION

In this paper, we have deliberated about SPI bus and different approaches of flash such as Normal SPI, Quad and Octal. The comparison is done in between Normal SPI approach, Quad approach and Octal approach. Enhancement is with high estimation to the SPI bus. SPI is the synchronous serial data transfer that is used for very short range generally in Embedded Systems. The throughput of Octal approach is high compared to Normal SPI approach and Quad approach, because eight data lines are used in Octal approach for data transfer. In Quad approach four data lines are used for transmission of data as well as receiving the data. Normal SPI approach uses only two data lines for data transmission purpose and receiving purpose. The instructions utilized here are Read instructions, Program and Erase instructions, Protection instructions, Security instructions, Status instructions, Control instructions which increases the performance of SPI bus.

ACKNOWLEDGMENT

I express my sincere pleasure to Mr.Sangameshwar Pujari, Manager of the Power Square India Pvt. Ltd. for supporting me throughout my project and giving me such a wonderful experience as an intern.

REFERENCES

- [1] P. Pavan, R. Bez, P. Olivo, and E. Zanoni, "Flash memory cells—An overview," Proc. IEEE, vol. 85, pp. 1248–1271, Aug. 1997.
- [2] G. Casagrande, "Flash memory testing," in Flash Memories, P. Cappelletti et al., Ed. Norwell, MA: Kluwer, 1999.
- [3] P. Cappelletti and A. Modelli, "Flash memory reliability," in Flash Memories, P. Cappelletti et al., Ed. Norwell, MA: Kluwer, 1999.
- [4] A. Modelli, A. Manstretta, and G. Torelli, "Basic feasibility constraints for multilevel CHE-programmed Flash memories," IEEE Trans. Electron Devices, vol. 48, pp. 2032–2042, Sept. 2001.
- [5] J.M.Portal & al., "Floating-Gate EEPROM Cell Model Based on MOS Model 9", IEEE Int'l Symp. on Circuits and Systems, 2002.
- [6] B. D. Yang and L. S. Kim, "A low-power ROM using charge recycling and charge sharing techniques," IEEE J. Solid-State Circuits, pp. 641-653, Apr. 2003.
- [7] K. Takeda, Y. Hagihara, Y. Aimoto, M. Nomura, Y. Nakazawa, T. Ishii and H. Kobatake, "A Read-Static-Noise-Margin-Free SRAM Cell for EE Journal Of Solid-State Circuits, vol. 41, no. 1, 2006
- [8] F. Leens, "Solutions for SPI Protocol Testing and Debugging in Embedded System," Byte Paradigm's White Paper, pp. 1-9, Revision 1.00, June 2008.
- [9] G. S. Jedhe, A. Ramamoorthy, and K. Varghese, A scalable high throughput firewall in FPGA, in 16th International Symposium on Field-Programmable Custom Computing Machines FCCM '08, pp. 43-52, 2008.
- [10] Jaehoon Jang, Han-Soo Kim, Wonseok Cho, Hoosung Cho, Jinho Kim, Sun Il Shim, Younggoan Jang, Jae-Hun Jeong, Byoung-Keun Son, Dong Woo Kim, Kihyun Kim, Jae-Joo Shim, Jin Soo Lim, Kyoung-Hoon Kim, Su Youn Yi, Ju-Young Lim, Dewill Chung, Hui-Chang Moon, Sungmin Hwang, Jong-Wook Lee, Yong-Hoon Son, U-In Chung, and Won-Seong Lee, " Vertical Cell Array using TCAT(Terabit Cell Array Transistor) Technology for Ultra High Density NAND Flash Memory," in VLSI Symp Tech. Dig., 2009, pp. 192-193
- [11] A. K. Oudjida, M. L. Berrandjia, R. Tiar, A. Liacha, K. Tahraoui, "FPGA implementation of I2C & SPI protocols: A comparative study," in Proc. 16th IEEE International Conference on Electronics, Circuits, and Systems, pp.507-510, Dec. 2009.
- [12] A. K. Oudjida, M. L. Berrandjia, A. Liacha, et al. "Design and test of general-purpose SPI master/slave IPs on OPB bus," 2010 7th International Multi-Conference on Systems Signals and Devices (SSD). 2010, pp. 1-6.
- [13] A.N. Gaidhane and M.P. Khorgade, "FPGA Implementation of Serial Peripheral Interface of FlexRay Controller," in Proc. 13th International Conference on Computer Modelling and Simulation (UKSim), pp.128-132, Apr. 2011.
- [14] N.Q. B. M. Noor and A. Saparon, "FPGA implementation of high speed serial peripheral interface for motion controller," in Proc. 2012 IEEE Symposium on Industrial Electronics and Applications (ISIEA), pp.78-83, Sept. 2012.
- [15] Jim Elliott and Eunseung Jung, "Ushering in the 3D Memory Era with V-NAND," in Flash Memory Summit 2013.