VHDL Implementation of an SPI Interface for an FRAM Memory over FPGA

Prajna, Mrs. Deepthi Dayanand , Shri Kanhu Charan Padhy

***Abstract*— SPI stands for Serial Peripheral Interface. It is a synchronous serial bus developed by Motorola for interfacing microprocessors and various devices such as memory chips, sensors, data converters, and printers etc. It's operation uses master slave architecture, but its implementation is simple and operation speed will be high .The designed SPI can be able to transfer the data between spartan6 FPGA and FRAM.**

***Index Terms*— Serial Peripheral Interface (SPI), Field Programmable Gate Array (FPGA), Ferroelectric Random Access Memory (FRAM).**

I. INTRODUCTION

We use different types of protocols for both long and short distance communication. The protocols like ETHERNET, SATA, USB are short distance protocols & I2C and SPI are short distance protocols. SPI is one of the most widely used protocols in chip communication. SPI is the best protocol, when there is a requirement to implement the data transfer between an IC and a set of peripherals. It is a synchronous serial bus, it has advantages like simple hardware and data transmission rate will be high compared to other protocols. It uses master slave architecture for transmitting the data. Master is the one which always generates the clock. SPI has got four interfacing signals. We can find reduced overhead in SPI protocol because there is no slave addressing.

II. SPI PROTOCOL

The SPI is a synchronous, full-duplex, serial data link that allows communication between a master and slave devices.

The Protocol uses two control lines and two data lines. The two control lines are as follows:

- SCLK-This is a clock signal generated by the master. All the data transmission operation will be controlled by this signal.
- SS-Slave Select. Line used by the master to select the slave device which is connected to bus. This line is active low signal, and needed when we want to connect more than one slave to the master.

The two data lines are:

- MOSI (Master out Slave In) –data line from master to slave.
- MISO (Master in Slave Out) –data line from slave to master.

In this type of protocol data transfer occurs in full duplex mode. Figure 1 shows the data transfer in SPI between master and slave.

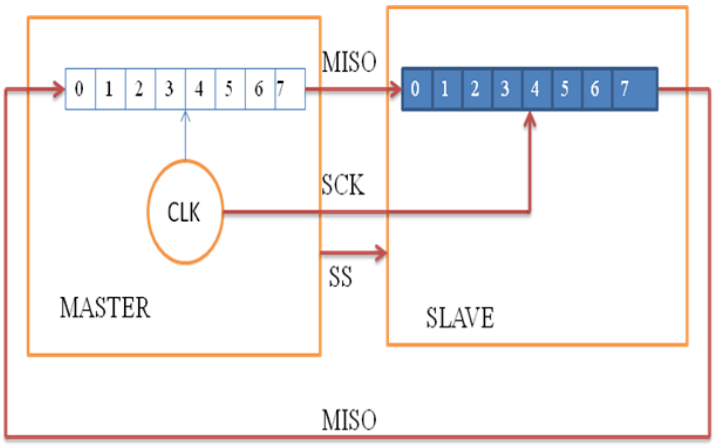

Fig.1. Data transfer in SPI

The SPI has four operating modes, which are determined by clock polarity(CPOL) and clock phase(CPHA).Modes are from 0 to 3.The way in which data is clocked in or out of an SPI device is controlled by these modes. In modes 0 and 1 once the SS goes low, the slave device samples the data at the first clock transition. . In modes 2 and 3, the slave device samples the data at the second clock transition after SS goes low. Most commonly used modes are 0 and 1.Both master and slave have to run in the same mode in order to achieve the proper communication between them.

SPI MODE	CPOL	CPHA
0	0	0
1	0	1
2	1	0
3	1	1

Table.1 SPI Modes

III. PROPOSED ARCHITECTURE

A. FM25V02 FRAM

The FM25V02 is a 256-Kbit ferroelectric random access memory. It is a nonvolatile, low voltage, high data retention, very fast SPI slave device. Unlike EEPROM, this memory has no write delays. Immediately after each byte successfully transferred to the slave device data is written to the array. Like EEPROM and serial flash, it has 8 pins, but it consumes less power and has high endurance.

B. STATUS REGISTER:

The status register of FM25V02 has 8 bits as shown in figure.2. These bits play the important role in the configuration of the device.

ISSN: 2278 – 1323

(MSB)	status register						(LSB)
bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
WPEN	0	0	0	BP1	BP0	WEL	0

Fig.2.FRAM Status Register

Fig. 3 shows the setup for the design. The data (4 bit data), address (4 bit address), clk (system clock),rst(optional reset) ,rd(read) and wr (write) are the inputs from toggle switches to the FPGA.SPI signals(SCK,MOSI,MISO and SS) are the outputs ,plus ssd, which is used to feeds an SSD display to display the data extracted from the connected FRAM.

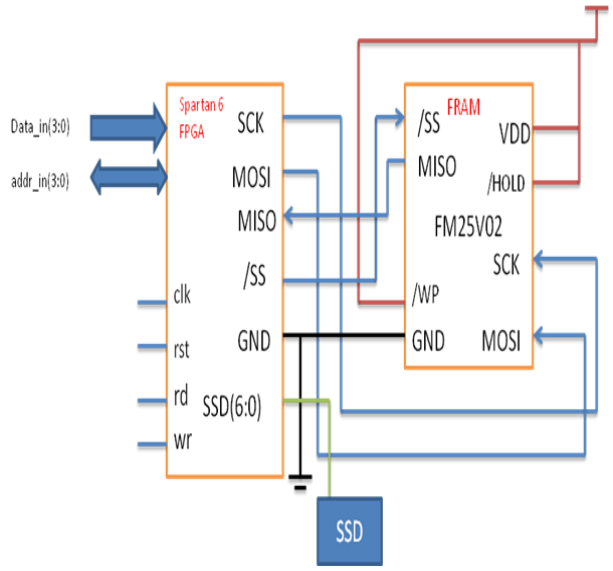

Fig.3.SPI Interface for a FRAM

Since writing to FRAM is fast (20Mbps in this project, against a few ms/word for EEPROMs), they can operate in operate in sequential write mode. RM25VO2 allows both sequential read and sequential write. The operation of the above circuit is explained using flowchart which is shown in fig.4.

Flow chart has two parts. Left side part is for write operation and other side shows read operation. Initially system will be in idle state. If wr signal is high master sends WREN command to set the write enable latch. Before sending WRITE command, master unselects the slave (ss='1'),then WRITE command. Once the WRITE command has been sent , master sends memory address ,and finally the data to be written to that address. Master keeps on writing the data to the FRAM until wr='0'.If wr input is low and rd='1' ,then master starts reading data from FRAM through MOSI line. So read operation consists of a read enable command (READ),followed by intended address, after which the FRAM responds with the data stored at that address.

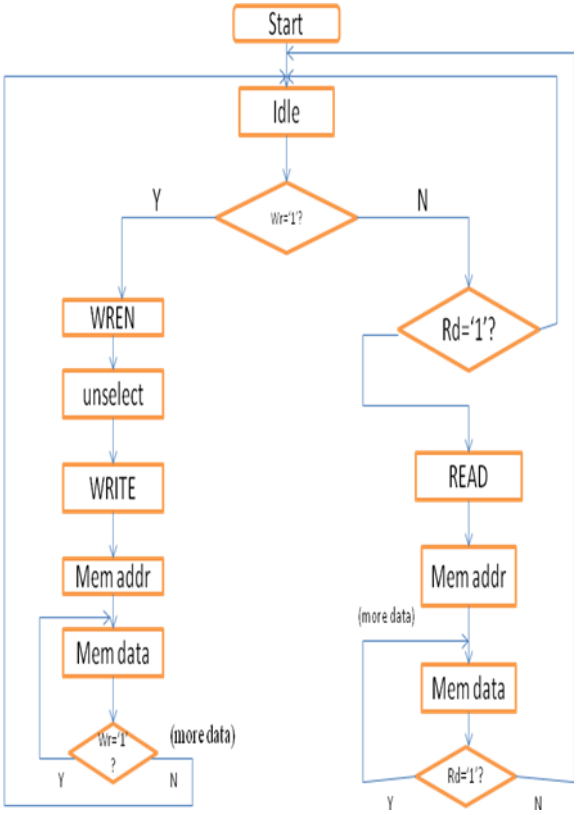

Fig.4.flowchart

The finite state machine for the above flowchart is shown in fig. 5, where lower branch represents reading and the upper one is for writing. The states for writing reading are rd-enable(READ command), rd-addr(memory address from which data must be read), rd-data(data read) and finally wait-rd(waits for rd to return to zero).The write states are wr-latch(WREN command), unselect(SS='1' to end the WREN command), wr-enable(WRITE command),wr-addr(memory address where data must be written),wr-data(data to be written),and wait_wr(waits for wr to return to zero).Under each state ,the corresponding SPI signal values are listed.

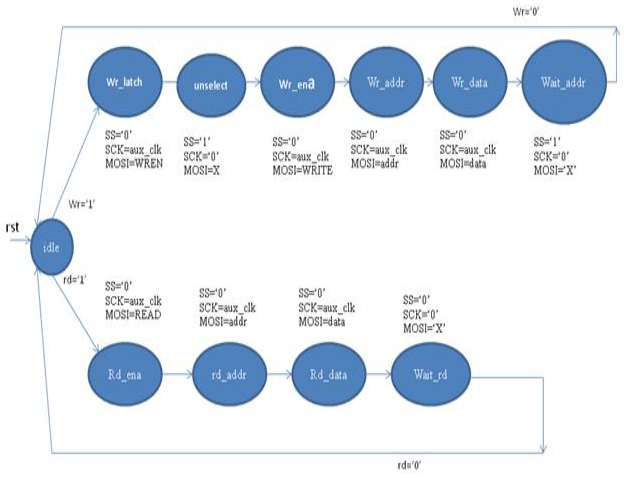

Fig.5. FSM for FRAM writing and reading.

ISSN: 2278 – 1323

IV. IMPLEMENTATION RESULTS

The results have been verified using 13.1 simulator. Simulation result for the design is shown in Fig.6.

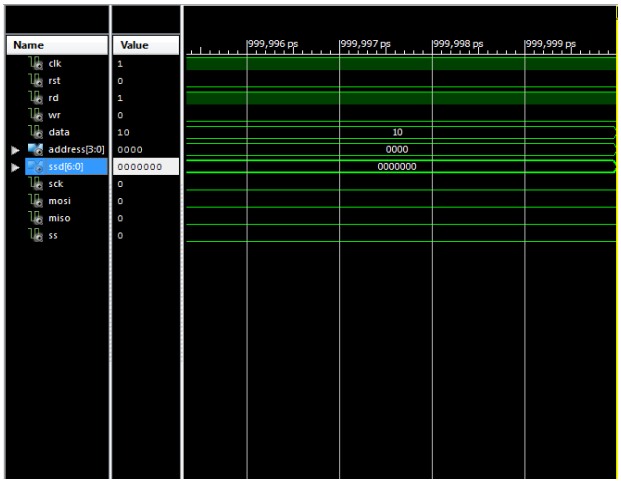

Fig.6.Simulation Result

As shown in Fig.7 the entire design uses 0% slice registers,0% LUTs, i.e. minimum resources are utilized by the design.

fram Project Status (05/06/2015 - 14:50:12)			
Project File:	spi.xise	**Parser Errors:**	No Errors
Module Name:	spi	**Implementation State:**	Synthesized
Target Device:	xc6slx45-3csg324	**• Errors:**	No Errors
Product Version:	ISE 13.1	**• Warnings:**	2 Warnings (2 new)
Design Goal:	Balanced	**• Routing Results:**	
Design Strategy:	Xilinx Default (unlocked)	**• Timing Constraints:**	
Environment:	System Settings	**• Final Timing Score:**	

Device Utilization Summary (estimated values)			[-]
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slice Registers	21	54576	0%
Number of Slice LUTs	54	27288	0%
Number of fully used LUT-FF pairs	21	54	38%
Number of bonded IOBs	23	218	10%
Number of BUFG/BUFGCTRLs	1	16	6%

Fig.7. Design summary.

Fig.8 shows the technology tree of the design.

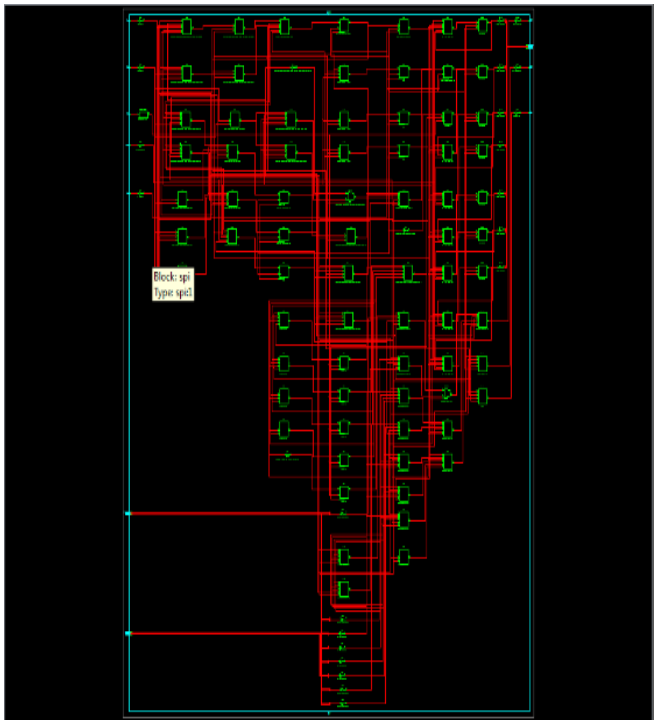

Fig.8. RTL View

V. CONCLUSION

In this paper I have illustrated an implementation of SPI protocol .FRAM is used as a slave device.The entire design is developed in VHDL code using Xilinx 13.1 version. The proposed design can be used in a system which requires high speed data transform. As many as slaves can be connected to this protocol.

REFERENCES

[1] Trupti D.shingare,R.T.Patil, "SPI Implementation on FPGA ", International Journal of Innovative Technology and Exploring Engineering,volume-2, pp. 7-9,January 2013.

[2] Leens, "An introduction to I2C and SPI protocols ", in proc.2009 IEEE Instrumentation and Measurement ,volume-12,pp.8-13,February 2009.

[3] Ananthula Srinivas, M.Kiran Kumar , Jugal Kishore Bandari, "Design and Verification of Serial Peripheral Interface ", Internationa Journal of Engineering Development and Research, pp.130-136,2008.

[4] "Realising the SPI communication in a multiprocessor system "Akos Szekacs, Tibor Szakaill and Zoltan Hegyk6zi SISY 2007 . 5th International Symposium on Intelligent Systems and Informatics 24-25 August, 2007 Subotica, Serbia.

Ms. PRAJNA received her B.E (Electronics & Communication) degree from KVG college of Engineering, Sullya in 2013 and currently pursuing her M.Tech in VLSI and Embedded systems from Sahyadri college of Engineering and Management, Mangalore. She has interest in the field VHDL modeling of digital circuits and testing, designing one layer PCB using Eagle, Schematic Diagram and Simulation of Basic Digital Circuits using Pspice and LT Spice. She has published one paper in IJEAM and another in IJSR.

ISSN: 2278 – 1323

Mrs.Deepthi Dayanand has Completed her B.E in Electronics and Communication Engineering from St Joseph College of Engineering, Mangalore and M.Tech in Digital Electronics and Communication from NITTE. Her area of specialization is digital electronics. She has 4.5 years of teaching experience.

Shri Kanhu Charan Padhy, has been graduated from UVCE, B.U,Bangalore (Electronics & communication) .He has done his PGDBA in HRM&Finance. He has served in DGAQA,Govt. Of India as a Group 'A' Officer. He has published some articles like 'Quality in Design and Reliability through Repair' in the Quality Focus Journal of ADE,DRDO. He has published one paper in the National Conference & another paper in the International Conference each during 2004 & 2006.He has been declared as the Best Citizen of India during 2003.He has been awarded with Commendation by the Gp Capt. P.Kamraju. He has served as a guide for the management students in the field of finnace. Presently he is the M.D of KCtronics innovative consultancy services private Ltd.

ISSN: 2278 – 1323