ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 5, May 2016

Design and Implementation of SPI Bus Protocol

Aviraj Ghanekar[1], Braj Kishor[2], Sachin Bandewar[3]

M. Tech. Student [VLSI], Dept. of ECE, SSSCE, Bhopal, M.P., India[1]

Assistant professor, Dept. of ECE, SSSCE, Bhopal, M.P., India[2]

H.O.D. Dept. of ECE, SSSCE, Bhopal, M.P., India[3]

ABSTRACT: A SPI bus is a communication protocol that allows serial data transfer between a master and a slave device. In this paper, we focus on present a full explanation of a Serial peripheral interface Master/Slave design and implementation. The Design specification is based on Motorola's Serial peripheral interface Guide version V03.06. The purposee of this paper is providing to communication among serial peripheral interfaces (SPI) Master to Serial peripheral interface Slave. This paper design approach that Can offer prospective way designs master/slave in SPI mode. The Complete design Implement in verilog Hardware Description language, and mapped onto Xilinx FPGA device.

KEYWORDS: SPI, Serial Communication, FPGA, Xlinx ISE 14.2, Modelsim.

I.INTRODUCTION

One SPI is the mainly used serial communication protocols for mutually inter-chip and intra-chip small/middle speed data-stream transfers**.** It is apply to interface among a microcontroller and other devices like peripheral EEPROMs, DACs, ADCs, etc. [1]

In the world of serial data communication we are use different data rate communication protocols. Serial Peripheral interface is considered as a small communication protocol. It is an important protocol each protocols have their specific purpose. USB, Ethernet and Serial AT Attachment, are meant for "outside the box communications" and data transmit in the whole system while serial Peripheral interface communication between integrated circuits for little or middle data transfer rate with peripheral devices [3].

II.SERIAL PERIPHERAL INTERFCE

The SPI is a average speed synchronous serial protocol designed by Motorola Corporation limited. It is a trendy interface used for linking peripherals to each other and to microprocessors/microcontroller. The majority journalism indicates to facilitate the interface can only be used for 8 bit or 16-bit building block data transfers, except various Motorola microcontrollers permit transfers of the specific range of blocks connecting 8 and 16 bits at a time. For the reason that of the serial nature of the interface, data transmit more than 16 bits at a time can be implementing simply through control signals.

The SPI interface uses as three-wire bus protocol a slave select line for each device linked to the bus. The four bus lines are as follows:

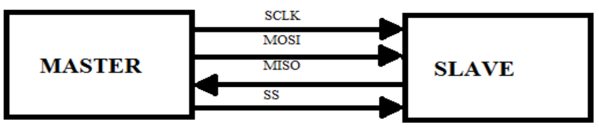

Fig. 1 SPI Master and Slave

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 5, May 2016

SCLK - The clock generated by the "Master". Their SCLK clock signals synchronize data transfer into the bus.
MISO - Master Input Slave Output. It's link to use data transfer from Slave to Master.
MOSI - Master out Slave In. It's link to transmit data from Master to Slave.
SS - This stands for Slave Select. When it goes low, the corresponding slave device will be selected. The SS line is used by Master device to select which slave to start communication with the Master [3].

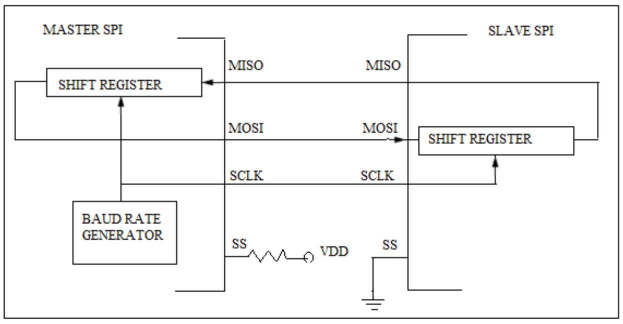

Fig. 2 SPI Shift Register

Data Register and eight-bit shift register: - It forms the main part of the SPI system. When an SPI transfer gets a place, a bit of data gets shifted out of the SPI master's data register and then, the serial data which comes from the slave's data register in sequence shifted into the master's data register. Therefore, by the time one SPI transmission completes, i.e. after 16th clock cycles, the contents of the master and slave will have been swap.
Control Register: - This is the register which offers the user preface control over the SPI operation. This register, when properly configured, can be used to control the data transfer to the SPI bus. These include enabling the SPI, configuring the SPI in master or slave mode.

Baud rate register: - Baud rate generation consists of series of divider stages. Eight bits in the SPI baud rate register establish the value of which the bus clock is divided. This provides the end user ample choices for baud rate generation with divisors ranging from 2 to 128. The baud rate generator is active only if the SPI is operating in the master mode.

The baud rate divisor equation is as follows:

$$\text{Baud Rate Divisor} = (\text{SPPR} + 1) \cdot 2^{(\text{SPR}+1)}$$

The BD can be calculated with the following equation:

$$\text{Baud Rate} = \text{Bus Clock} / \text{Baud Rate Divisor}$$

III.KEY FEATURE OF SPI MASTER AND SLAVE

In SPI has low or medium (n MHz to 10n MHz) data transfer rate depending on the implementation. [5]. SPI offers multiple transfer rates based on the SPI master baud rate, which can be programmed by the user. The SPI module supports multi-slave operation. The master and slave can be transmitter or receiver based on its mode of operation. It is capable of receiving and transmitting on both rising and falling edges of the clock independently.

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 5, May 2016

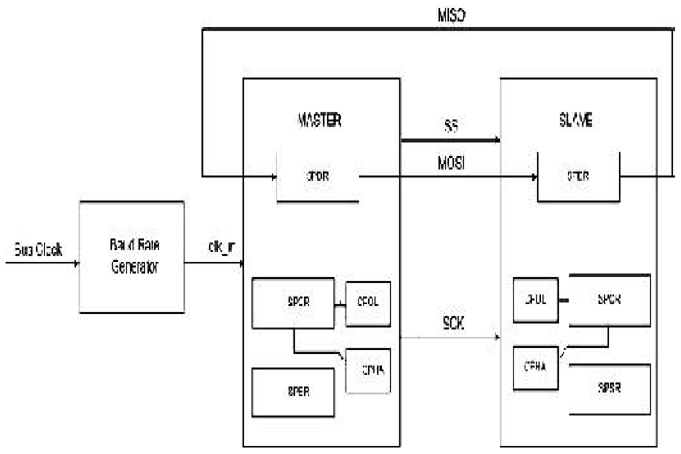

Fig. 2 SPI Block Architecture

IV.SPI DESIGN STEP

A normally standard protocol follows some Design steps and verifies Design Transmitter detection:

A. Start Signals Generation: Cognitive In this step, the Master generates transaction signals with respect to the system clock and modelling Master Clock (SCLK).

B. MOSI and SS: In this step, MOSI pin starts transferring data after to SS (Slave Select). Whenever SS is low it's able to receive data coming from master though MOSI pin.

C. Master Reading Signals: In these steps, Slave transmits data through MISO pin using the shift register one by one data shift in the shift register and each bit transmits every clock pulse. After sixteen clock pulse, data transmission completed.

V. CONCLUSION

Our work focuses on the performance analysis of SPI along with the RTL Design work. Design part involves around Motorola 3.0 Specification and codes write in the Verilog. Serial peripheral interface RTL divided into two blocks one is Master and other is Slave. RTL code written all blocks work done using XILINX ISE tool which gives simulation and synthesis validation of specification.

REFERENCES

[1] A.K. Oudjida, M.L. Berrandjia, A. Liacha, R. Tiar, K. Tahraoui & Y.N. Alhoumays, " Design and Test of General-Purpose SPI Master/Slave IPs on OPB Bus," 2010 IEEE.First Author and Second Author. 2002. International Journal of Scientific Research in Science, Engineering and Technology. (Nov 2002),
[2] Motorola Inc., "SPI Block Guide V03.06," March 2003.
[3] FreescaleSemiconductorInc., "MC68HC912D60 manual V 4.0,"September 2010.
[4] F. Leens, "An Introduction to I2C and SPI Protocols," IEEE Instrumentation & Measurement Magazine, pp. 8-13, February 2009.
[5] A.K. Oudjida, M.L. Berrandjia, R. Tiar, A. Liacha, K. Tahraoui "FPGA Implementation of I2C & SPI Protocols: a Comparative Study "Microelectronics and Nanotechnology Division Centre de Développement des Technologies Avancées CDTA 978-1-4244-5091-6/09©2009 IEEE.
[6] H Ananthula Srinivas, M.K.Kumar and Jugal Kishore Bhandari," Design and Verification of Serial Peripheral Interface", International Journal of Engineering Development and Research, ISSN: 2321-9939.