International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Impact Factor (2012): 3.358

Design and Simulation of SPI Master / Slave Using Verilog HDL

T. Durga Prasad[1], B. Ramesh Babu[2]

[1]Student, Department of Electronics and Communication Engineering, DRK College of Engineering and Technology, Jawaharlal Nehru Technological University, Hyderabad, Andhra Pradesh, India

[2]Assistant professor, Department of Electronics and Communication Engineering, DRK College of Engineering and Technology, Hyderabad, Andhra Pradesh, India

Abstract: ***The object of this paper is to design and simulation of SPI (serial peripheral interface) master and slave using verilog HDL. The SPI (serial peripheral interface) is a kind of serial communication protocol. It transfers synchronous serial data in full duplex mode. The SPI is commonly used for communications between Integrated Circuits for communication with On-Board Peripherals. The SPI communicate in two modes master and slave. Where the master device generates serial clock and multiple slave devices are allowed with individual salve select lines. And the whole design is simulated and synthesized with Xilinx ISE design suite 13.2.***

Keywords: SPI (serial peripheral interface), Verilog HDL.

1. Introduction

The Serial Peripheral Interface (SPI) is a popular and widely used device for serial data transmission. The SPI is a high speed (up to 400 MHz) synchronous serial interface/protocol designed by Motorola [1]. It is a popular interface used for connecting peripherals to each other and to microprocessors. Most literature indicates that the interface can only be used for eight or sixteen bit block data transfers, but many Motorola microcontrollers allow transfers of any range of blocks between two and sixteen bits at a time. Because of the serial nature of the interface, data transfers of more than sixteen bits at a time can be implemented easily through control signals. The SPI can implement either the master or slave protocol. When configured as a master, the SPI can control up to 32 independent SPI slaves. The width of the receive and transmit registers are configurable more than sixteen bits. With SPI we can connect as many devices as many pins we have on the main microcontroller. The speed of the communication between ICs is much faster. Full Duplex communication is done using SPI [2].

2. Serial Peripheral Interface Design

The internal architecture of SPI mainly consists of two modules, master module and slave module as shown in figure 1. The SPI module allows a full duplex, synchronous, serial communication between the MCU and peripheral devices. It is enabled by setting the SPI enable (SPE) bit in SPI Control Register. The SPI communicates using two data lines, a control line, and a synchronization clock:

- Master out Slave in (MOSI) - Output data from the master to the inputs of the slaves.
- Master in Slave out (MISO) - Output data from a slave to the input of the master.
- Serial Clock (SCLK) - Clock driven by the master to slaves, used to synchronize the data bits.
- Slave Select (SS) - Select signal driven by the master to individual slaves, used to select the target slave [2].

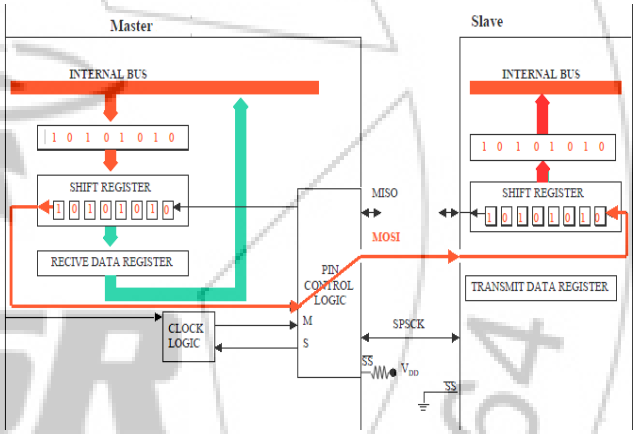


Figure 1: SPI Block Diagram

The main element of the SPI system is the SPI Data Register. The 8-bit data register in the master and the 8-bit data register in the slave are linked by the MOSI and MISO pins to form a distributed 16-bit register. When a data transfer operation is performed, this 16-bit register is serially shifted eight bit positions by the serial clock from the master, so data is exchanged between the master and the slave. Data written to the master SPI Data Register becomes the output data for the slave, and data read from the master SPI Data Register after a transfer operation is the input data from the slave. The SPI system operates in two modes master and slave.

2.1 Master Mode Operation

The SPI operates in master mode when the MSTR control bit is set. The transmission begins by writing to the master SPI Transmit Data Register. If the shift register is empty, the byte immediately transfers to the shift register. Once the byte is transferred from the SPI transmit register to the shift register, the SPTE control bit is set indicating that another byte can be written to the SPI transmit data register. In master mode,

before transmission SS pin is connected to V_{DD} if the single slave module is used. Then the byte begins shifting out a bit at a time on the MOSI pin synchronized with the master serial clock. The data transmission will continue for 8 clock cycles, transferring all 8-bits. The transmission ends when the whole byte is shifted out of the master SPI shift register into the slave SPI shift register. The slave shift register is then automatically transferred to the slave SPI receive data register if it is empty. The SPRF control bit is set indicating that the SPI receive register is full and waiting to be read [3].

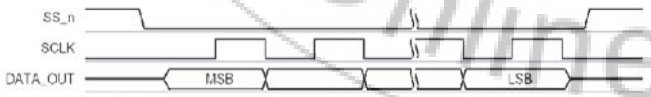
2.2 Slave Mode Operation

The SPI operates in slave mode when the MSTR control bit is clear. In slave mode, the function of serial data output pin MISO and serial data input pin MOSI. The serial clock is input to the slave from master. And the SS pin is the slave select input. Before a data transmission the SS pin is connected to ground if the single slave module is used. Once the byte is transferred from the SPI transmit register to the shift register, the SPTE control bit is set indicating that another byte can be written to the SPI transmit data register. The byte of data shifting a bit at time on the MISO pin synchronized with the master serial clock. The master shift register is then automatically transferred to the master SPI receive data register if it is empty. The SPRF control bit is set indicating that the SPI receive register is full and waiting to be read.

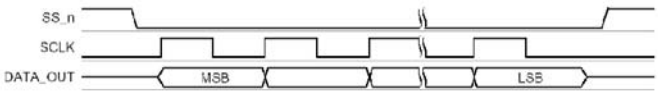
3. SPI Data Transmission

In SPI data transmission, Clock polarity (COPL) and Clock phase (CPHA) are two timing settings. The timing settings affect the timing relationship between the SS, SCLK, MOSI and MISO signals. The configuration is done by two bits in the SPI control register (SPCR). The Clock polarity (COPL) control bit, this can be set 0 or 1. When clock polarity is set to 0, the idle state for SCLK is low. When clock polarity is set to 1, the idle state for SCLK is high [4]. And the clock phase (CPHA) control bit, this can be set 0 or 1. When clock phase is 0, data is latched on the leading edge of SCLK, and data changes on trailing edge. When clock phase is 1, data is latched on the trailing edge of SCLK, and data changes on the leading edge. The figure 2 shows the all possible cases of clock polarity and clock phase.

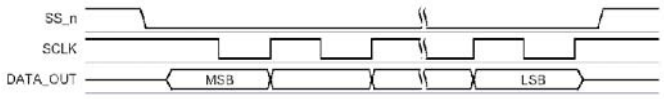
Clock polarity = 0, Clock phase = 0



Clock polarity = 0, Clock phase = 1



Clock polarity = 1, Clock phase = 0



Clock polarity = 1, Clock phase = 1

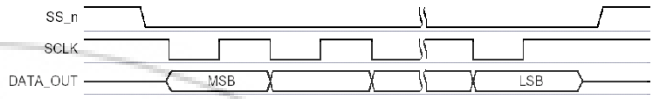


Figure 2: possible cases of clock polarity and clock phase.

4. Simulation and Results

The data transmission from master to slave and slave to master has been synthesized using the Xilinx ISE design suite 13.2 and the simulation results are shown in figure respectively.

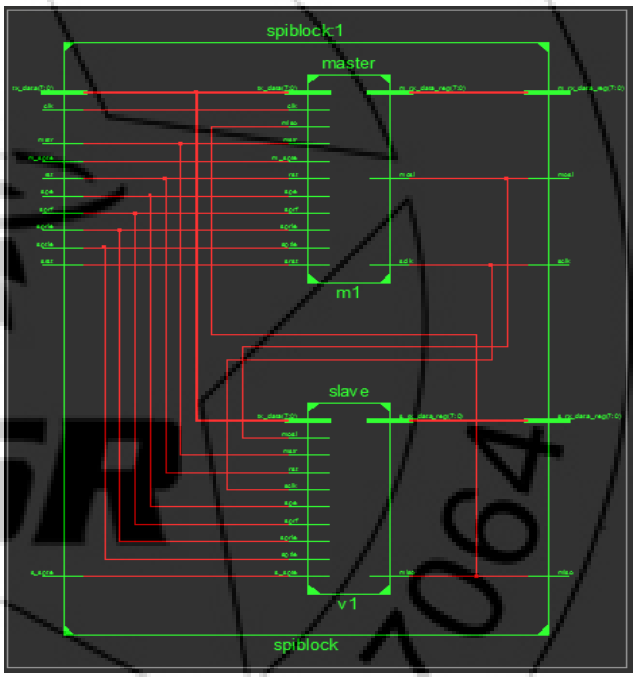


Figure 3: SPI RTL Schematic of Top Module

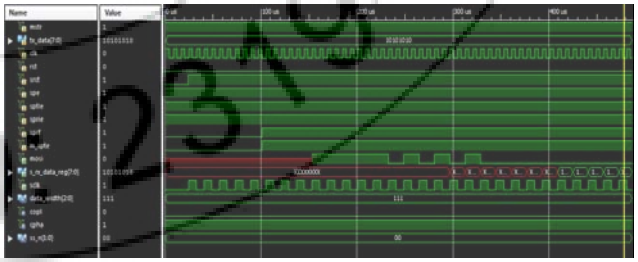


Figure 4: SPI Master to Slave Data Transmission

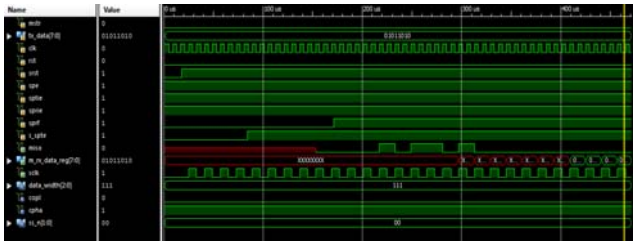


Figure 5: SPI Slave to Master Data Transmission

5. Conclusion

The SPI Master and Slave has been implemented using verilog HDL and Xilinx ISE 13.2 was used for simulation. In this paper, we have more focused on data transmission between master and slave modules. We have verified the data in slave device same as the data in the master device and various possible cases of clock polarity and clock phase are verified. Further, we have also done functional verification.

References

[1] Motorola Inc., "SPI Block Guide V03.06," February 2003.
[2] Quartus II Handbook version 9.0 volume 5: Embedded Peripherals. Altera SPI core interval time core chapter in volume 5 of the Quartus II.
[3] Digital DNA from Motorola, "SPI Tutorial introduction".
[4] Freescale Semiconductor, (2008, October 14). Freescale SPI Block Guide V04.01 July.
[5] F.Leens, "An Introduction to I2C and SPI Protocols,"IEEE Instrumentation & Measurement Magazine, pp. 8-13, February 2009.
[6] Microchip, serial peripheral interface section 18.

Author Profiles

T. Durga Prasad received the Bachelor of Technology Degree in Electronics and Communication Engineering from DRK College of Engineering and Technology, Jawaharlal Nehru Technological University in 2014.

B. Ramesh Babu received the Bachelor of technology Degree in Electronics and Communication Engineering from DJR Institute of Engineering and Technology, Jawaharlal Nehru Technological University, Kakinada and Master of Technology Degree in Electronics and Communication Engineering (VLSI and Embedded Systems) from Jawaharlal Nehru Technological University, Hyderabad. He is working as an Assistant professor in DRK College of Engineering and Technology.