100 MHz High Speed SPI Master: Design, Implementation and Study on Limitations of using SPI at High Speed

Mitu Raj
Department of Electronics and Communication,
Electronics Research and Development Center of India- Institute of Technology, Trivandrum, India
iammituraj@gmail.com

Abstract— SPI or Serial Peripheral Interface is among the fastest synchronous serial communication protocols used in embedded systems. High throughput and simplicity of SPI communication has made SPI protocol; a de facto standard. Designs based on FPGAs (Field Programmable Gate Arrays) enhance reusability, flexibility and faster prototyping of digital systems, especially serial buses, which are inevitable in almost all designs. This paper discusses the design and implementation of a 100 MHz High Speed SPI Master Core on FPGA. State machine approach is employed for the RTL (Register Transfer Level) design of the core. The paper also discusses the challenges and limitations of implementing such a high speed SPI bus in digital systems. The SPI Master Core was successfully implemented on Altera Cyclone III FPGA for a speed of 100 MHz.

Keywords— *Field Programmable Gate Array, Intellectual Property Core, Reusability, Serial Communication, Serial Peripheral Interface.*

*****

I. INTRODUCTION

Serial Communication is the communication process in which one data bit is sent at a time. Parallel bits of data are converted into a stream of data bits and transmitted serially over a single line; Hence the term ‘’serial”. The advantages of Serial Communication over Parallel Communication are reduction of noise from adjacent lines, propagation delay, cost and area. These benefits constitute the primary reason why it is the most dominant form of data communication in embedded systems. Some of the most popular Serial Communication protocols are USB, UART, I2C, SPI etc.

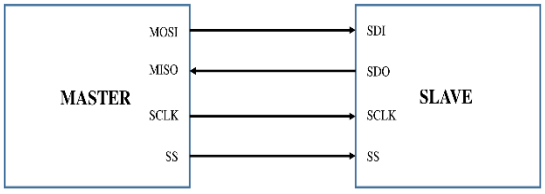

Fig. 1. SPI Master- Slave Communication

SPI or Serial Peripheral Interface is a full duplex, synchronous serial communication protocol. Unlike UART or I2C, no definite speed is defined for an SPI Bus. Because of its high data rate and protocol simplicity, it has become a de facto standard in communication in most digital systems. SPI has single master- multi slave architecture [1]. Usually single slave structure is used, as shown in Fig. 1. Next section discusses the interface signals used in SPI communication and modes of operation.

II. SPI BUS COMMUNICATION

A. *Bus Interface Signals*

Typical master- slave structure of SPI communication is shown in Fig. 1. Various interface signals used in SPI are:

- MOSI – Master Out Slave In
- MISO – Master In Slave Out
- SCLK – Clock
- SS – Slave Select

MOSI is the master’s output port for data transmission to the slave, and MISO is the master’s input port for data reception from the slave. MOSI is controlled by the master and MISO is controlled by the slave. SCLK is the clock generated by the master for data transmission and reception. Frequency of SCLK determines the speed of communication. SS is the signal (normally active low) to select the slave for communication. For multi- slave structure, multiple SS signals can be there, or daisy chain technique is employed by the master [2], [3]. In multi- slave protocol, MISO is pulled to high impedance state by rest of the slaves to avoid bus contention.

B. *Modes of Operation*

SPI is a full duplex bus; i.e., Both transmission and reception of data happen at the same time. It has a ring like topology with two shift registers: one at the master, and the other at the slave (refer to Fig. 2). They shift out one bit, every clock cycle, in a circular fashion. Even if only data reception is required, it is inevitable that the data in the master’s register is shifted out to the slave’s register.

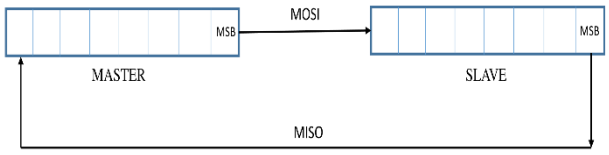

Fig. 2. Ring Topology in SPI

The SPI communication starts with clock configuration; configuring it for clock frequency, supported by the slave. The Master then selects the slave by pulling the SS line low. One bit of data is then transmitted and received by the master and the slave in every clock cycle. Master ends the communication by

TABLE 1

Mode	Idle clock state	Data capture	Data write
0	low	rising edge	falling edge
1	low	falling edge	rising edge
2	high	falling edge	rising edge
3	high	rising edge	falling edge

deselecting the slave, by pulling SS high. Lack of start bit, stop bit and acknowledgment makes SPI a simple, less strict and faster communication protocol than its contemporaries like UART, I2C.

SPI has four modes of operation: Mode 0, Mode 1, Mode 2 and Mode 3. The features of these modes are represented in Table 1.

RTL Design and FPGA Implementation of a High Speed SPI Master are described in next sections.

III. RTL Design of Spi Master

Following are the design specifications of the SPI Master, presented in this work:

- 8-bit data frame
- SCLK frequency = input clock frequency/ 4
- Configurable mode of operation
- Supports single slave

RTL design of the SPI Master is done in VHDL/Verilog. The top level view of the design is shown in Fig. 3.

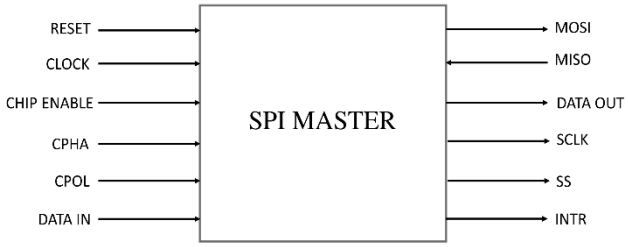

Fig. 3. Top Level View of the SPI Master

The ports associated with the designed SPI Master are described in Table 2.

TABLE 2

Port	Direction	Length
RESET	input	1
CLOCK	input	1
SCLK	output	1
CHIP ENABLE	input	1
CPHA	input	1
CPOL	input	1
DATA IN	input	8
DATA OUT	output	8
MOSI	output	1
MISO	input	1
SS	output	1
INTR	output	1

The port signals are described below:

- RESET– to reset the core to idle state.
- CLOCK – input clock.
- SCLK – SPI Master output clock.
- CHIP ENABLE – to enable the chip.
- CPOL, CPHA – clock polarity and phase; to choose the mode of operation (00,01,10,11).
- DATA IN – 8-bit input data.
- DATA OUT – 8-bit output data.
- MOSI – serial data output.
- MISO – serial data input.
- SS – Slave Select.
- INTR – interrupt.

FSM approach is employed to design the SPI Master in HDL. Thus the core can be efficiently implemented as Moore or Mealy machine, with minimum no. of states [4], [5].

A. *Algorithm for the FSM*

FSM approach is employed for efficient implementation of the design. In this paper, Moore based FSM is used to design the SPI Master. Algorithm for FSM design (Mode 0) is described below. For reducing power consumption due to state switching, gray encoding may be used for state encoding.

Step 1. Reset state; SS= high, INTR= SCLK= low – *State 1*
Step 2. Configure Mode.
Step 3. Chip enabled? Go to step 4, else go to Step 1.
Step 4. SCLK= low, Write MSB to MOSI – *State 2*
Step 5. SCLK= high, Read MISO – *State 3*
Step 6. Go to Step 4 until all eight bits are transmitted/received
Step 7. SCLK= low, INTR= high – *State 4*
Step 8. INTR= low, go to Step 4.
Step 9. Go to Reset state.

B. *Functional Simulation*

After RTL design, functional simulation is performed to verify the correct operation of the design. The simulation results for Mode 0 operation are shown in Fig. 4. The RTL simulation of the SPI Master Core was done in Modelsim EDA Tool.

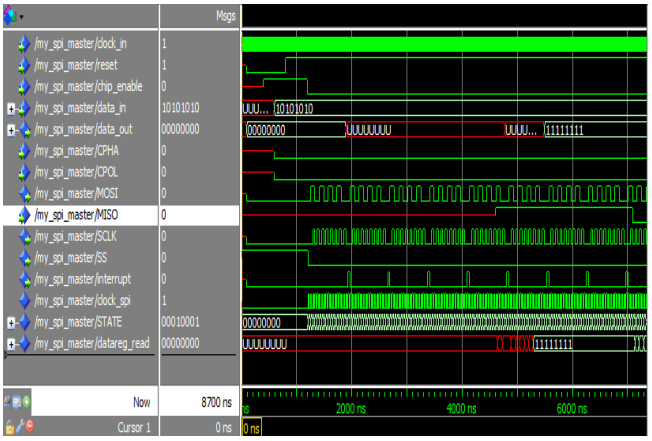

Fig. 4. RTL Simulation of the SPI Master

IV. Implementation of Spi Master on Fpga

FPGA Implementation of the SPI Master Core was done in Altera Cyclone III FPGA. The RTL view of the SPI Master is

 ISSN: 2321-8169
697 – 700

given in Fig. 5. The SPI Master was equally optimized for power, area and performance. The design was successfully synthesized and verified timing for 100 MHz clock. The maximum frequency of operation is obtained as 119 MHz for slow model.

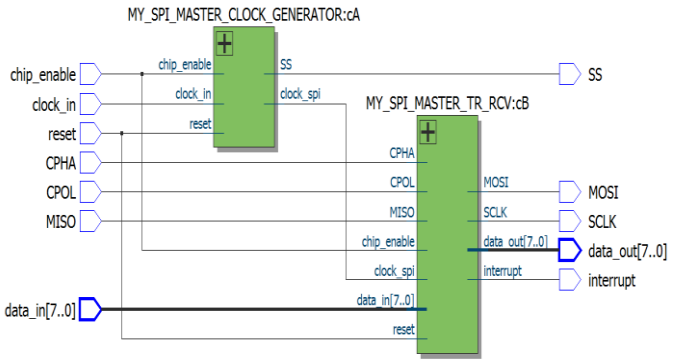

Fig. 5. RTL View of the SPI Master

In the timing report, clk is the clock input, clk_spi is the internal clock, which is divided by two to get the spi clock. The synthesis and timing reports of the design, obtained from Quartus II software is shown in Fig. 7 and Fig. 6 respectively.

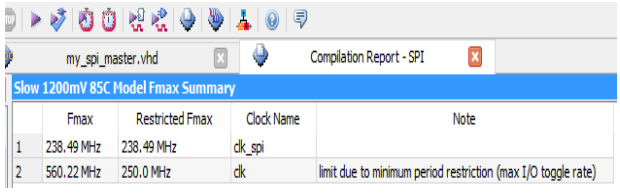

Fig. 6. Timing Report for Slow Model of the SPI Master

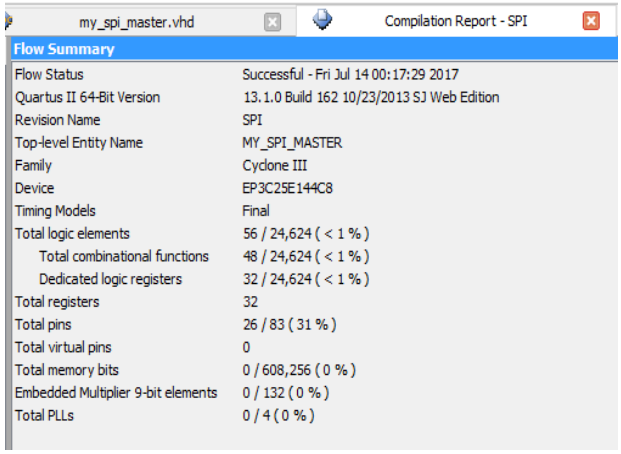

Fig. 7. Flow Summary of the SPI Master

V. Limitations of Implementing High Speed SPI Communication in Digital Systems

Implementing high speed SPI communication have limitations, as it puts constraints on the maximum length of the interconnect between the master and the slave. Any interconnect can be represented using RLC model or simply a distributed/lumped RC model [6]. The parasitic capacitances and resistance of the interconnect limit the rise time/ fall time or speed of propagation of signals on the wire. As a consequence, the maximum distance of communication between master and slave is constrained.

Consider Fig. 8. The timing diagram for Mode 1 SPI communication between master and slave is shown in it, including various propagation delays involved in the communication.

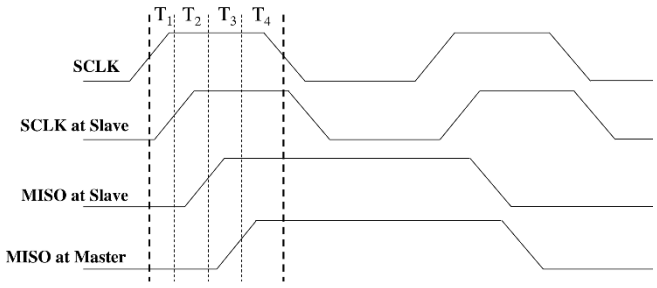

Fig. 8. Timing Diagram for SPI- Mode 1 Operation

T_1: Propagation delay for SCLK, $T_{pp\text{-}clk}$
T_2: Slave clock to output delay, $T_{clk\text{-}q}$
T_3: MISO data propagation delay, $T_{pp\text{-}data}$
T_4: Time available to sample the data, T_{samp}

$$T_{samp} = T_{sclk}/2 - T_{pp\text{-}sclk} - T_{sclk\text{-}q} - T_{pp\text{-}data} \quad (1)$$

Data bit on MISO is written by the slave on the rising edge of the clock. To assure signal integrity and to ensure that the data bit is properly sampled by the master, the data bit sent from the slave has to reach the master before the setup time for the sampling edge (falling edge) of the clock. i.e., the following expression has to be satisfied for proper data sampling:

$$T_{samp} > T_{setup}$$
$$\text{i.e., } T_{sclk} > 2\times (T_{setup} + T_{pp\text{-}sclk} + T_{sclk\text{-}q} + T_{pp\text{-}data}) \quad (2)$$

The above relation limits the maximum frequency of SPI clock for a given interconnect length. For a given SPI clock, interconnect length has to be limited such that $T_{pp\text{-}data}$ satisfies the equation (2). To ensure the proper operation of SPI at high speed, the data path delays have to be reduced. It can be achieved by placing buffers along the data path, or by using faster slaves, or by reducing interconnect length. To understand how buffers reduce the interconnect delay $T_{pp\text{-}data}$, refer to Fig. 9.

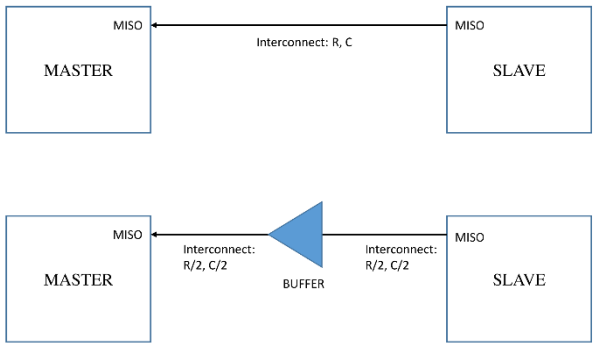

Fig. 9. Flow Summary of the SPI Master

The MISO interconnect between Master and Slave can be represented as RC lumped model. Before adding buffer, the interconnect delay, $T_{pp\text{-}data}$ = RC. After adding buffer, propagation delay is reduced, as $T_{pp\text{-}data}$ = RC/4 + T_{buff} + RC/4, where T_{buff} is the buffer propagation delay.

VI. CONCLUSION

Design and implementation of a high speed SPI Master Core is presented in this work. The design employs FSM approach with minimum no. of states. The design was successfully implemented on FPGA, with timing verified for SCLK frequency = 100 MHz. Challenges and limitations of implementing such a high- speed SPI communication in digital systems have been investigated. It is inferred that the interconnects play a vital role in the tradeoff between speed of communication and maximum distance to keep signal integrity.

REFERENCES

[1] A. K. Oudjida, M. L. Berrandjia, R. Tiar, A. Liacha, K. Tahraoui, "FPGA Implementation of I2C & SPI Protocols", 16th IEEE International Conference on Electronics, Circuits and Systems (ICECS), Yasmine Hammamet, Dec 2009, pp. 507-510.

[2] Alan Berenbaum, Eileen Marando, Richard Wahler, "Point-to-Point Serial Peripheral Interface for Data Communication between Devices Configured in a Daisy-Chain", U.S Patent 0 297 829, Nov 2013.

[3] X. Wang, H. Zhang, L. Zhang, J. Zhang, and Y. Hao, "A Daisy-Chain SPI Interface in a Battery Voltage Monitoring IC for Electric Vehicles, 12th IEEE International Conference on Solid-State and Integrated Circuit Technology (ICSICT), Guilin, May 2014, pp. 1-3.

[4] Gustavo Sutter, Elias Todorovich, Sergio Lopez-Buedo, and Eduardo Boemo, "FSM Decomposition for Low Power in FPGA", Proceedings of the Reconfigurable Computing Is Going Mainstream, 12th International Conference on Field Programmable Logic and Applications, UK, Sep 2002, pp. 350-359.

[5] A. Tiwari and K. A. Tomko, "Enhanced Reliability of Finite-State Machines in FPGA through Efficient Fault Detection and Correction", IEEE Transactions on Reliability, vol. 54, no. 3, pp. 459-467, Sep 2005.

[6] M. Sanaullah and M. H. Chowdhury, "Analysis of RLC Interconnect Delay Model using Second Order Approximation", IEEE International Symposium on Circuits and Systems (ISCAS), Melbourne VIC, June 2014, pp. 2756-2759.