International Journal on Recent and Innovation Trends in Computing and Communication ISSN: 2321-8169
Volume: 4 Issue: 5 188 - 191

Design of SPI(Serial Peripheral Interface) Protocol with DO-254 Compliance

Koushik M[1]
M.Tech Dept. of E.C.E,
SJB Institute of Technology
Bengaluru, INDIA
mkoushik43@gmail.com

Anushree R[2]
Dept. of E.C.E
SJB Institute of Technology
Bengaluru, INDIA
anurgowda@gmail.com

Kumaraswamy K V[3]
Technical Manager,
Trident Tech Labs Pvt. Ltd.,
Bengaluru, INDIA
kuvarji@gmail.com

***Abstract*—** The objective is to design the SPI protocol compliant with avionic hardware design standards (D0-254). This is to make sure that the protocol follows the most stringent standards so as to develop a fool proof design that would be highly reliable and would not fail under conventional circumstances. The significance of this project is designing the Serial Peripheral Interface (SPI) protocol in the DO-254 flow which is an industry standard used for the hardware process. Using DO-254 standards SPI is coded in Verilog. Single Master-Single Slave, Independent Slave configuration, Daisy Chain configuration all the three designs are designed following the rule sets of DO254. It is verified using the assertion based verification (ABV) or System Verilog Assertions (SVA).

Keywords- *DO-254 approach, Assertion Based Verification(ABV), Verilog*

*****

I. INTRODUCTION

In serial communication protocol there are two types and are synchronous serial communication and asynchronous serial communication.

Serial synchronous communication means the transmitting and receiving of data which is depending on the clocks and is synchronized. Serial asynchronous communication means the transmission and reception of data is not only dependent on clocks but also depends on start and stop signals. SPI is the Serial Peripheral Interface which is a serial synchronous communication protocol. It is a de-facto standard. SPI is designed in such a way that it communicates between two devices or systems based on particular application. Today, at the low end of the Communication Protocols, there are mainly two protocols: Inter- Integrated Circuit (I2C) and the Serial Peripheral Interface (SPI) Protocols. Both the protocols are well suited for communications between Integrated Circuits for communication with On-Board Peripherals. SPI is the commonly used protocol for the low/medium speed data transfer. It is also used for connecting the peripherals.

DO-254 is deriving of requirements for a design flow along with a very strict process design assurance. The standards for the design to be followed in this approach. In DO-254 flow, the coding standards must be documented, followed, and reviewed.. The DO-254 mainly focus to ensure that the product performs its intended function (via design assurance process).

II. SPI PROTOCOL

The SPI Protocol is of two configurations Independent Slave and Daisy Configuration. In general SPI protocol is designed as a Single Master Single Slave.

A. SINGLE MASTER SINGLE SLAVE

Serial Peripheral Interface (SPI) is a serial synchronous communication protocol which is a de-facto standard used in the industries. It is a full duplex communication protocol. SPI can be four-wire or three-wire or two-wire interface bus. Basically a four-wire SPI is used for all the applications. SPI consists of Master and Slave. Fig 1 shows Master-Slave diagram where it has four signals for communication and are :-

* SCLK- Serial clock generated upon dividing the Master clock.

* SS- Slave select. This signal should be active low for the Slave to respond with Master and also it selects the slave if there is more than one slave for the communication.

* MOSI: - Master out Slave in.

* MISO: - Master in Slave out.

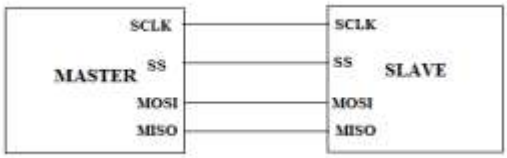

Fig 1: SPI busSingle Master Single Slave

B. INDEPENDENT SLAVE CONFIGURATION

In this configuration, there are more than one slaves. Consider three slaves, all these three slaves have their own independent slave select signal (ss1, ss2, ss3) as shown in Fig2. To avoid unnecessary cross-talk between Master and Slave a pull up resistor is used between the power source and slave select line. Initially this line will be active high. The Master makes any one of the slave select ss1, ss2, ss3 line to low so that it starts communicating. Since, MISO pins are connected it has to be a tri-state pin (can be high, low or high-impedance).

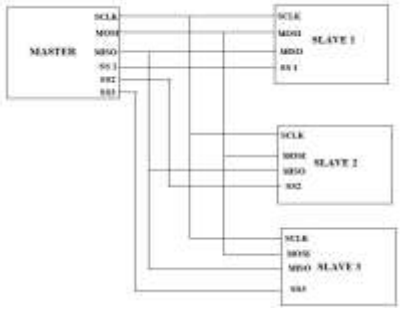

Fig 2: Independent Slave Configuration

C. DAISY CHAIN CONFIGURATION

Some applications of SPI require daisy chain configuration of SPI as shown in Fig3. Consider three slaves, the main

Master sends the serial data to the first slave. The input to the second slave will be the output of first slave. The input to the third slave will be the output of the second slave. The output of the third slave will be again given back to the main Master through the MISO line of the third slave. In this configuration only single slave select (SS) line is required.

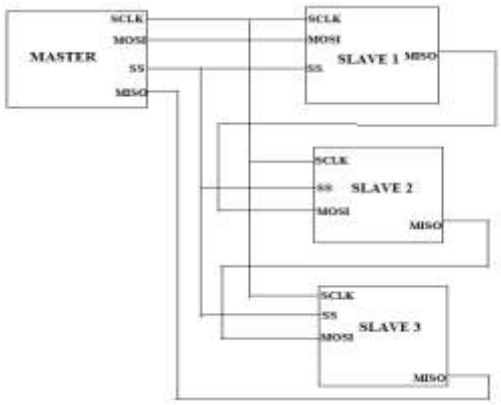

Fig 3: Daisy Chain Configuration

III. Implementation of SPI with DO-254 Approach

DO-254 is the deriving the requirements for a design Flow and also with a very strict process design assurance. DO-254 discusses the need for the standards of a design. In DO-254 programs, HDL coding standards must be documented, followed, and reviewed. While reviews can be done manually, an automated approach (whenever possible) guarantees a more consistent RTL code quality assessment.

DO-254 is an avionics industry document called "Design Assurance Guidance for Airborne Electronic Hardware". Focus of DO-254 is to ensure product performs its intended function (via design assurance process). Invoked as law by the FAA/EASA* in 2005. Now a worldwide standard for aviation design and spreading into military applications. It has influenced other safety-critical industries. DO-254 compliance can be expensive. Expense can be greatly reduced through understanding, proper preparation, targeted tools.

The DO-254 flow for the FPGA design flow is as follows

- Requirements
- Conceptual design
- Detailed Design
- Verification
- Implementation
- Product

A. Requirements

Establishing the requirements for a design flow and managing theses requirements which will be evolving throughout the design will be difficult. The product specifications given by the customers must meet the design specifications designed by the designer. The tool used for this is Mentor Graphics ReqTracer.

B. Conceptual Design

Sometimes conceptual design is also referred to as architectural exploration in the FPGA design process. The tool used for this is Mentor Graphics HDL Designer.

In this, firstly we will do a block diagram with inputs and outputs defined. Once the block diagram is done, we will define the functionality of it by truth table, flow chart, finite state machine (fsm).

C. Detailed Design

The actual development work begins in the this phase, where we will be modeling the design without the physical hardware. HDL coding will occur during the process. The code is written in a Verilog language. Verilog is preferred over VHDL because Verilog is easy to understand. The code written is checked using the DRC (design Rule Check).

The tool used to check DRC is HDL designer. Later the code is simulated by writing the test cases for it. The tool used for simulation is Questa Sim. In this we can also run the code coverage which tells coverage details of the code.

D. Verification

The verification of the design can be done by using more sophisticated methods such as assertion-based design and clock-domain crossing analysis. In this design, assertion based design verification is used.

Assertion based verification (ABV) or System Verilog Assertions (SVA) is defined as a description of a property of the design. It describes the behavior of a design.

E. Implementation

Synthesis and place-and –route can be considered as part of the Implementation phase of the dO-254 cycle. The tool used for synthesis is Precision RTL Plus. This tool ensures that synthesis and subsequent place-and-route are performed as safely and repeatably as possible. For place and route the tool used is Xilinx.

F. Product

The final step in the DO-254 flow is the product. It establishes the baseline that includes design and manufacturing data for consistent production and replication of the FPGA.

IV. Results

The design SPI protocol compliance with DO-254 approach has been designed. In this case the number of slaves is three, which requires the master clock to be divided. We have simulated the design using a Questa Sim and verified the design using SVA. We got the output data same as the input data and further assertion verification is also done.

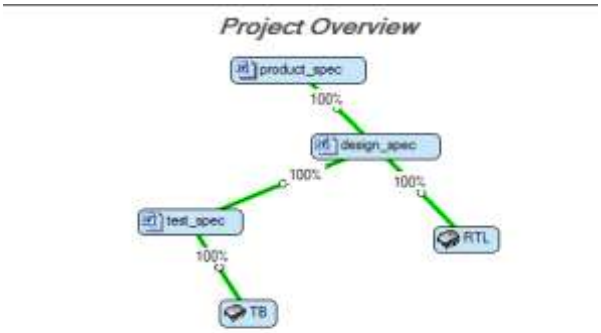

Fig4:- Requirements

 ISSN: 2321-8169
188 - 191

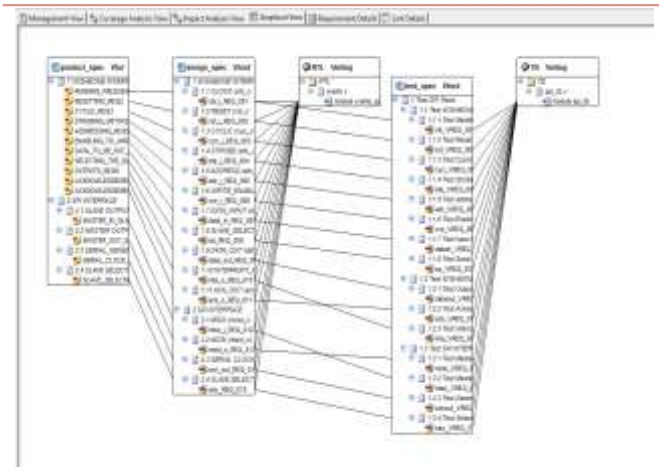

Fig5:Mapping of the Requirements

Fig4 & Fig5 shows the requirements output that is it tells whether the product specification is matching the design specifications and design specifications matching the test specifications.

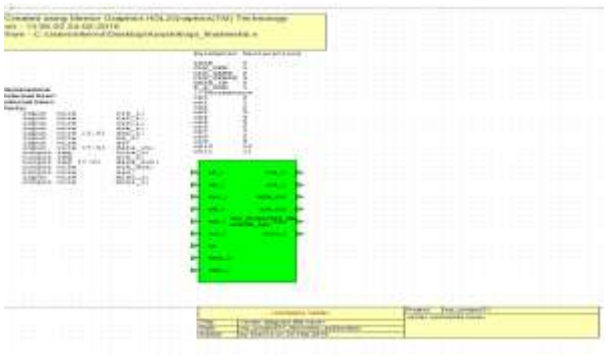

Fig6: Block Diagram of wishbone-SPI inetrafce

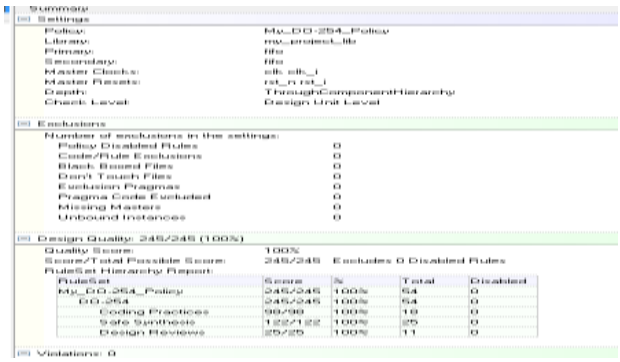

Fig7:-Design Rule Check

Fig6 shows the block diagram of the wishbone SPI interface where the wishbone signals and SPI signals are mentioned. Fig7 shows the Design Rule Check (DRC) where it tells quality of the code which compares with the rule sets of DO-254. The tool used is HDL Designer.

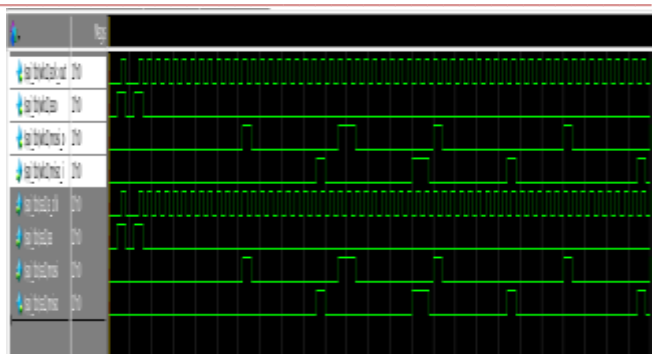

Fig8:- Output waveform of Single Master- Single Slave

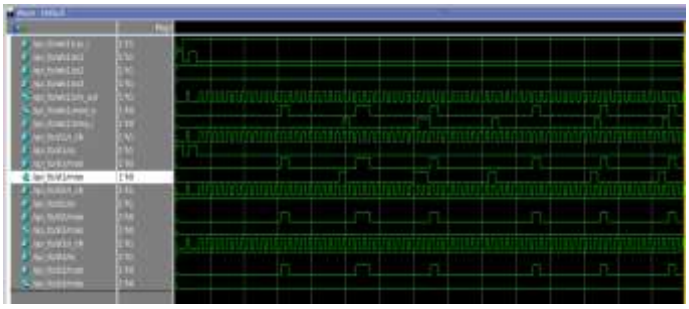

Fig9:-Output waveform of Independent Slave Configuration

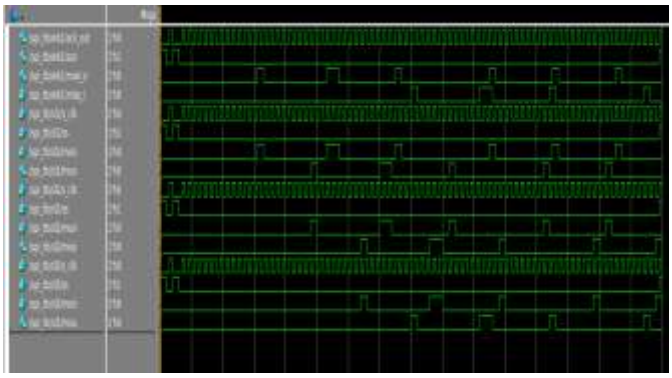

Fig10:- Output waveform of Daisy chain Configuration

Fig8, Fig9, Fig10 shows the output waveform of the Single Master-Single Slave, Independent Slave & Daisy Chain Configuration. The tool used is Questa Sim.

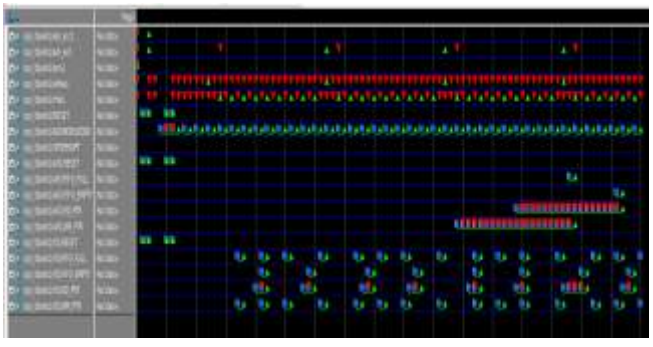

Fig11:- Waveform showing assertions.

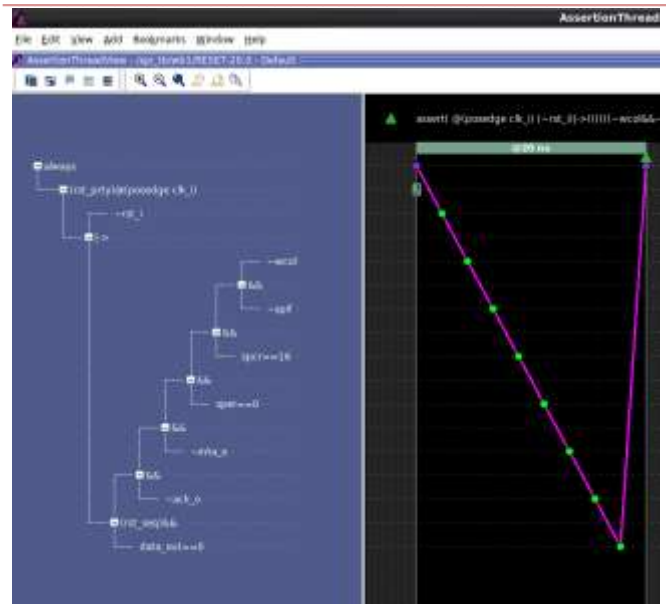

Fig12:- Assertion Thread View(ATV)

Fig11 shows the assertions in the waveform. The aseertion based verification is done in order to check the bahior of a property of a design. Fig12 shows the assertion thread view(ATV) in which the assertions are represented graphically.

Table1:-Power Optimization

Power	Without Optimization	With optimization
Total Power (mW)	193.01	55.55
Dynamic Power (mW)	171.31	35.30
Static Power (mW)	21.71	20.20

Table2:-Area Optimization

Area	Without Optimization	With Optimization
IO's	30	30
LUT's	115	86

V. CONCLUSIONS

The SPI compliant with the DO-254 standards is implemented. The design of SPI is designed using Verilog and verified using the SVA or ABV. The power and area have been optimized and the result obtained with the DO-254 approach suggests that design is safe and can be used in avionics applications.

ACKNOWLEDGMENT

I would like to thank Mr. Shuddhodhan Shetty, Application Engineer, Trident Tech Labs, for guiding me in the project.

REFERENCES

[1] Oruganti D N and Yellampalli S S "Design of power efficient SPI Interface", pp 26022606, Sept. 2014, IEEE

[2] Fareha Naqvi "Design and Implementation of Serial Peripheral Interface (SPI) Protocol Using Verilog HDL", IJEDR, vol.3, 2015.

[3] Saha S, Rahman M A and Thakur A "Design and Implementation of SPI bus protocol with Built-in-self-test capability over FPGA", pp 1-6, April 2014, IEEE.

[4] Donald E Thomas and Philip R Moorby "Verilog Hardware Descriptive Language" Springer Science & Business Media New York

[5] Randall Fulton and Roy Vandermolen "Airborne Electronic Hardware Design Assurance" CRC Press Taylor & Francis Group Boca Raton Londan New York.

[6] Tianxiang Liu "IP Design of Universal Multiple Devices SPI Interface" IEEE.978-1-61284-632-3, 2011 .

[7] Specification for the "WISHBONE System-on-Chip (SoC) Interconnection Architecture for Portable IP Cores" Revision: B.3, Released: September 7, 2002.

[8] K. Aditya, M. Sivakumar, Fazal Noorbasha, T. Praveen Blessington, "Design and Functional Verification of A SPI Master Slave Core Using System Verilog", (IJSCE) ISSN: 2231-2307, Volume-2, Issue-2, May 2012

[9] S.Sarns and J. Woehr, "Exploring I2C," Embedded Systems Programming, vol. 4, p. 46, Sept. 1991.

[10] F.Leens, "An Introduction to I2C and SPI Protocols,"IEEE Instrumentation & Measurement Magazine, pp. 8-13, February 2009.

[11] M.Sandya1, K.Rajasekhar, "Design and Verification of Serial Peripheral Interface", International Journal of Engineering Trends and Technology-Volume3Issue4-2012.