

Design and Verification of Serial Peripheral Interface

M.Sandya¹, K.Rajasekhar²

1. Department of E.C.E, Akula Sree Ramulu Engineering College, Tetali

2. H.O.D, Department of E.C.E, Akula Sree Ramulu Engineering College, Tetali

Abstract— The objective of this project is to design SPI Master Core using Verilog and verify the code using system verilog. Serial Peripheral Interface (SPI) is an interface that facilitates the transfer of synchronous serial data. SPI (Serial Peripheral Interface) is a synchronous serial data link that operates in full duplex mode. It communicates in master/slave mode where the master device initiates the data frame. Multiple slave devices are allowed with individual slave select line. Serial Peripheral Interface of symmetrical structure can be synthesized using Xilinx 12.3, and then can be simulated using Questa 10.0b. It is a popular interface used for connecting peripherals to each other and to microprocessors. The verification of the project is done using system verilog.

Keywords— SPI interface; serial; Verilog HDL; system Verilog

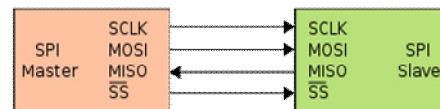
I. INTRODUCTION

In our days microcontroller embedded systems are widespread in a large area of products from industrial, medical, educational to domestic domains, there are often many modules in the designs. For example, nearly every system includes: some intelligent control, usually a microcontroller core; General-purpose circuits like LCD drivers, remote I/O ports, RAM, EEPROM, or data converters; Application-oriented circuits for communication interfaces and/or computation intensive task. So the communication between these modules is very important, in many applications the interface is still the bottleneck of system performance. In such scenario the reuse of intellectual property (IP) macrocells is becoming the center of gravity for design productivity and the key for being able to produce chips that really work. All the integrated components must be connected each other and every SoC must be linked each other in an efficient way that allows a fast and error-free communication. The communication among SoC is the key to grant high performances: the most used solution for interconnecting SoC is a serial bus which presents great advantage in terms of costs. Using few wires to link different devices means few occupied area and consequently minor costs for the producer. A large number of serial protocols are actually used, so if a designer wants to create a SoC that can communicate using various protocols he must equip his system with a large number of interfaces, despite the fact that when the SoC is on the field usually only few of the supported protocols will be used. In this work the design and verification of SPI interface is presented.

II. SPI BUS PRINCIPLE

SPI is a synchronous serial bus protocol developed by Motorola and integrated in many of their microcontrollers. SPI bus consists of four signals: master out slave in (MOSI), master in slave out (MISO), serial clock (SCLK), and active-low chip select (CS). As a multi-master/slave protocol, communications between the master and selected slave use the unidirectional MISO and MOSI lines, to achieve data rates in full duplex mode. With SPI we can connect as many devices as many pins we have on the main microcontroller. The speed of the communication between the ICs is much faster. Full Duplex communication is done using SPI.

Figure 1. SPI Block diagram



III. DESIGN PRINCIPLES

Verilog HDL is a hardware description language, it can be used for different levels of logic design, can be used for digital system logic simulation, timing analysis and logic synthesis, a wide range of applications. In this paper, a SPI Interface Module is designed using verilog, achieve a common structure which can realize IP reused. According to the SPI bus principle, can achieve the bidirectional data transmission between the microprocessor and slave devices using several functional modules.

A. System architecture design

According to the SPI bus principle, the design of the SPI Master is compatible with the SPI protocol, the design of host side is equivalent to the slave devices of wishbone bus specification-compliant, and the overall structure can be divided into the following three functional modules: Clock generator, Serial interface, Wishbone interface.

B. Module design

1). Design of Clock Generation module – *clk_gen*

The clock signal of the `clk_gen` is root from the external system clock—`wb_clk`, the module accordance with different frequency factor of the clock register to produce the output signal `s_clk`. Because there is no response mechanism for SPI, in order to ensure the reliability of timing, design a module can generate reliable serial clock of transmission whether for the odd or even Frequency Division. Divider is the efficient and basic part of the digital integrated circuits and FPGA design, also an important component of the communication system. Its basic idea is according to relevant algorithm produce the frequency wanted. This module focus on the odd frequency division, in order to save resources, changes the odd frequency situation can achieve the goal of even frequency. The `s_clk` of Master core systems is generated by divider of `wb_clk`, by changing the value of the divider can achieve arbitrary clock output frequency. The expression of `wb_clk` and `s_clk` is as follows:

$$f_{sclk} = \frac{f_{wb_clk}}{(DIVIDER+1)*2}$$

Using verilog language to describe Clock Generation module, synthesis with ISE, it generate circuit shown in Figure 2.

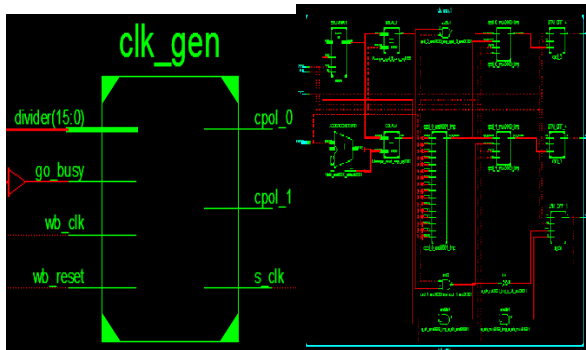


Figure 2. Clock Generation Circuit Module

2). Serial interface module design-- spi-shift

SPI data transfer is the core module. This module is responsible for the data parallel into serial out and serial data into parallel out. In this paper, the design of the shift is different with the usual shift SPI module, the reusing of register can use less hardware resources and increase the time to transmit data, thereby enhancing the overall rate of data transmission. Parallel data into bit longer, for example, data 128bit, in order to increase the speed of transmission, the

design enhance the expense of the resources of the previous conservative SPI module. SPI Master in host side as a slave device to receive data, at the same time as the master device to send data. Verilog code for this module is synthesised by ISE as shown in Figure 3.

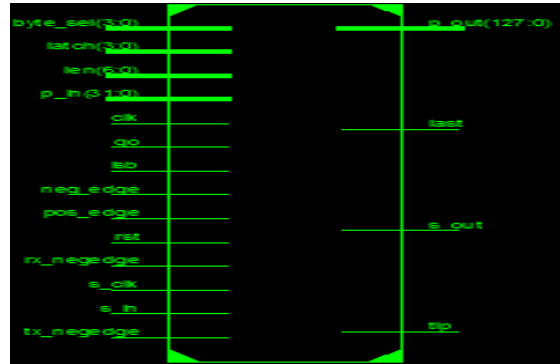


Figure 3. Serial interface module circuit

3). Top-level module --TOP

Based on the analysis of the SPI protocol get the basic structure of high-speed reusable SPI bus, including: the clock generator module, data transfer module. The important role of the top-level module is to ensure sub-module work smoothly. Therefore, the top of the SPI module needs the control word, the normal operation of clock generation module and data transmission module. Module synthesized by ISE as shown in Figure 4.

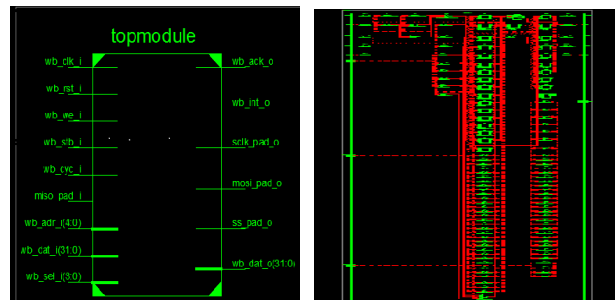


Figure 4. Top-level module circuit

IV. SIMULATION AND VERIFICATION

Using verilog language description design the SPI interface circuit, synthesis with ISE, and then use Questa sim to simulate. In order to establish the test platform, first establish the master module to simulate the Wishbone protocol, and establish the slave module to simulate SPI protocol at the same time. Then to compare and check the receive/send data and addresses. The test bench is designed using system verilog language. Verification is also the major work of this paper

because as we are using many modules and having many inputs we are going to use System verilog for verifying the design. Here the Design is considered as design under test (DUT). Through verification we can send the stimulus according to our wish. Here we can even constraint the unwanted stimulus. Components of Verification Environment are as follows : RTL module, Interface module to connect DUT with TB, Transaction ,Generator, Driver, Receiver, Scoreboard, TOP level module which encapsulates the above all with RTL and Test Cases.

SPI-top testbench overall structure is shown in figure 5.

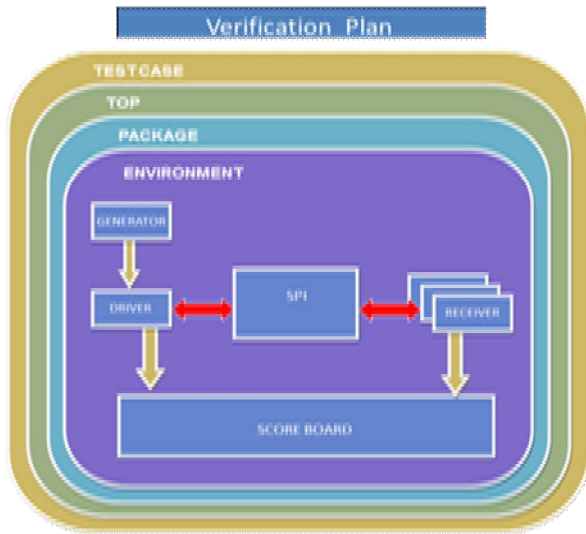


Figure 5.SPI-top testbench overall structure

Simulation results are as shown in Figure 6.

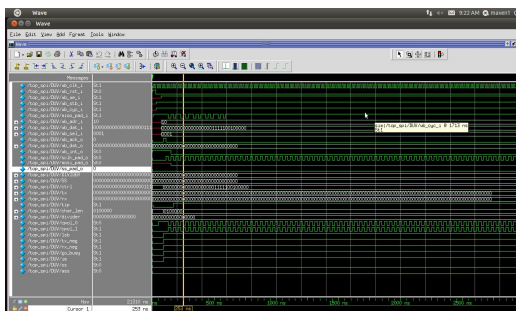


Figure 6.SPI-simulation results

Coverage Report is shown below in Figure 7

Design Coverage Summary:		Coverage Summary by Type:			
Weighted Average:	83.9%	Weighted Average:	83.9%		
Design Scope	Coverage (%)	Coverage Type	Bins	Hits	Coverage (%)
spi_top_module	82.5%	Covergroup	12	12	100.0%
DUV_IF	70.7%	Branch	158	129	81.6%
TEST	70.7%	Condition	87	64	73.6%
DUV	83.8%	Toggle	788	633	80.3%
spi_pkg	78.6%				
spi_sb	100.0%				

Figure 7. Coverage Report

V. CONCLUSION

In this, we have designed the SPI Master-Slave core based upon design-reuse methodology. SPI

transmission timing is very strict, so this paper design a reliable and stable clock generation module, both the case of odd-even frequently were considered. Data transmission module is a simple and the transfer speed is faster. We have also done code coverage and achieved Design coverage of 100 percent. We have verified the data in slave device same as the data in the master device. Further, we have also done functional verification. The complete function of the registers is done in this paper. The innovation of this article: System verilog language is used in order to cover all the functions of the code.

REFERENCES

- [1] www.opencore.org.Simon Srot. SPI Master Core Specification,Rev.0.6. May 16,2007
- [2] Prophet, Graham. Communications IP adds SPI interface to FP-GA. EDN, v 48, n 27, Dec 11, 2003.
- [3] Motorola, "MC68HC II manual,".
- [4] Smart Computing Dictionary, Serial Peripheral Interface (SPI) (online)<http://www.smartcomputing.com/editorial/dictionary/detail.asp?guid=&searchtype=1&DicID=12820&RefType=Dictionary> (access date 28May 2006)
- [5] Frédéric Leens , An Introduction to I2C and SPI Protocols,IEEE Xplore.
- [6] ZHANG Yan-wei, Verilog HDL detailed design procedure, Telecom Press