Design and Functional Verification of A SPI Master Slave Core Using System Verilog

K.Aditya,M.Sivakumar, Fazal Noorbasha,T.Praveen Blessington

Abstract:- **Synchronous serial interfaces are widely used to provide economical board level interfaces between different devices such as microcontrollers, DACs ADCs and other. Many IC manufacturers produce components that are compatible with SPI and Microwire/plus. The SPI Master core is compatible with both protocols as master with some additional functionality. At the hosts side,the core acts like a Wishbone compliant slave device. The SPI master core consists of three parts, Serial interface, clock generator and Wishbone interface. The SPI core has five 32-bit registers through the Wishbone compatible interface. The serial interface consists of slave select lines, serial clock lines, as well as input and output data lines. All transfers are full duplex transfers of a programmable number of bits per transfer(upto 64 bits).It has 8 slave select lines but only one is selected at a time. We design the SPI Master-Slave core design using system verilog and do functional verification for our design in modelsim.**

KEY WORDS:- SPI,Wishbone,coverage.

I. INTRODUCTION

In our days nearly every system includes some intelligent control, usually a Microcontroller Core. General-purpose circuits like LCD drivers, remote I/0 ports, RAM, EEPROM, or data converters. Application-oriented circuits for communication interfaces and/or computation intensive task. So the communication between these modules are very important.In such aspect 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. With the development of the IC manufacturing, the communication between hardware devices became particularly important. The Currently widely used protocols such as WISHBONE bus protocol, I2C bus protocol, ARM bus protocol and so on, let hardware devices to communicate through the appointment of the rules and match the timing for achieving the purpose of exchanging data. SPI is a serial interface protocol, compared to other protocols, it has high transmission speed, simple to use and little pins advantages. The four interfaces are required by standard SPI protocol at least. Usually, the devices which based on SPI protocol are divided into master-device and slave-device for transmitting the data. The chip select signal and clock signal have be generated by the master-device when the data exchange has been processed. Therefore, the master device must have

multiple chip select interfaces for slave devices when it controls multiple slave devices. That will no longer meet the standard SPI protocol. The standard SPI communication is a single-master communication, that means all the communications are only have one master device. Thus the devices which based on SPI protocol are limited by the both aspects. This design which (adopts the parameterization method, identify the master/slave devices automatically and use TSM(Time Sharing Multiplex) technology to control the same slave device at the same time) is implemented for aiming at the defects. The design is fully complied with the four interfaces of standard SPI protocol.

II. SPI INTERFACE DIAGRAM

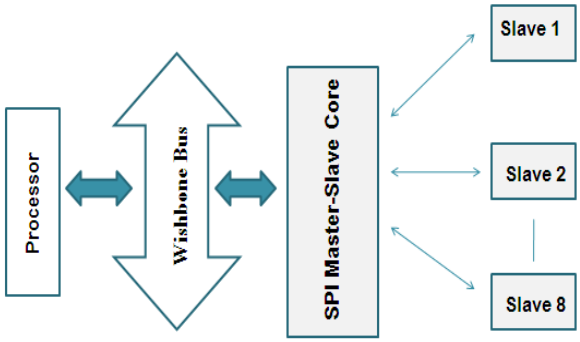


Figure 1: interface diagram of SPI

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 (SCK), 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 over 1Mbps in full duplex mode. A disadvantage to SPI is the requirement to have separate CS lines for each slave. 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 thanks for the Full Duplex proper of the SPI communication.

III. WISHBONE BUS INTERFACE

The WISHBONE System-on-Chip (SoC) Interconnection Architecture for Portable IP Cores is a flexible design methodology for use with semiconductor IP cores. Its purpose is to faster design reuse by alleviating

System-on-Chip integration problems. The main objectives of this specification are to create a flexible interconnection means for use with semiconductor IP cores. This allows various IP cores to be connected together to form a System-on-Chip and to make WISHBONE interfaces independent of logic signaling levels. WISHBONE uses a MASTER/SLAVE architecture. That means that functional modules with MASTER interfaces initiate data transactions to participating SLAVE interfaces. The MASTERs and SLAVEs communicate through an interconnection interface called the INTERCON. The INTERCON is best thought of as a 'cloud' that contains circuits. These circuits allow MASTERs to communicate with SLAVEs. Mentioned below are the wishbone interface signals used for our Serial Peripheral Interface communication.

Here all internal WISHBONE logic is registered to the rising edge of the [clk_i] clock
input. So master clock decides the data to send or not. The active low asynchronous reset input [rst_i] forces the core to restart. All internal registers are preset and all state-machines are set to an initial state. The interrupt request output is asserted when the core needs service from the host
system. When asserted, the cycle input [cyc_i] indicates that a valid bus cycle is in progress. The logical AND function of [cyc_i] and [stb_i] indicates a valid transfer cycle to/from the core.So when only the cycle input is high strobe signal is high. The strobe input [stb_i] is asserted when the core is being addressed. The core only responds to WISHBONE cycles when [stb_i] is asserted, except for the [rst_i], which always receive a response. The address array input [adr_i] is used to pass a binary coded address to the core. The most significant bit is at the higher number of the array. When asserted, the write enable input [we_i] indicates that the current bus cycle is a write cycle. When negated, it indicates that the current bus cycle is a read cycle. The data array input [dat_i] is used to pass binary data from the current WISHBONE

Port	Width	Direction	Description
wb_clk_i	1	Input	Master clock
wb_rst_i	1	Input	Synchronous reset, active high
wb_adr_i	5	Input	Lower address bits
wb_dat_i	32	Input	Data towards the core
wb_dat_o	32	Output	Data from the core
wb_sel_i	4	Input	Byte select signals
wb_we_i	1	Input	Write enable input
wb_stb_i	1	Input	Strobe signal/Core select input
wb_cyc_i	1	Input	Valid bus cycle input
wb_ack_o	1	Output	Bus cycle acknowledge output
wb_err_o	1	Output	Bus cycle error output
wb_int_o	1	Output	Interrupt signal output

Figure 2: Wishbone Interface signals

Master to the core. All data transfers are 8 bit wide. The data array output [dat_o] is used to pass binary data from the core to the current WISHBONE Master. All data transfers are 8 bit wide. When asserted, the acknowledge output [ack_o] indicates the normal termination of a valid bus cycle.

IV. SPI MASTER-SLAVE OPERATION

In the standard SPI communication, the clock signal (SCLK) through the bus from the master to send to each slave. All the SPI signals are synchronized by this clock signal. How many slave devices have, the master-device will have how many selection signal pins (CSn), which used for selecting slave devices. Recently widely used SPI devices are customized by one-to-one or one-to-more. In standard SPI, the master device must have many CS signals in the Multi-device communication. The communication is a single-master communication that means more than one master device connect to the slave device is strictly prohibited.

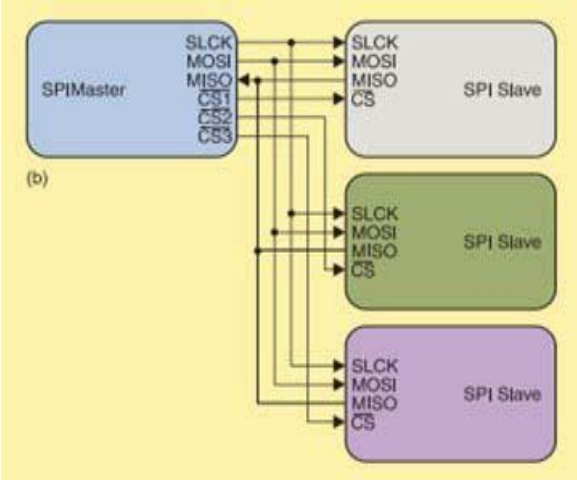


Figure 3: Standard SPI Multi-device communication Schematic

In the work mode, a master device connect with one or more slave device, it can not be changed at this time. Other devices want to control the slave device which has been controlled by another master device, it must stop master device's working first and disconnect with it, then connect to the new devices. In this way, a master device can select multiple slave devices do not have to add extra chip select pins and multiple master devices will use the TSM(Time Sharing Multiplex) through the level of communication priority to control the same slave device without to stop working. So universal multiple devices SPI interface IP is more flexible and effective.

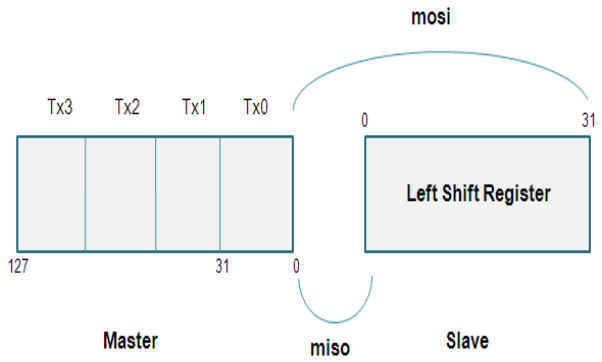


Figure 4: Operation between master and slave.

When we want to send data of 32 bit, the LSB bit is selected and sent from master to the slave, the bit goes to the MSB position and at the same time the slave acts as a left shift register. This is called as MOSI (Master Out Slave In). At the same time the LSB bit in the slave goes to the LSB bit of the master. This is called as MISO(Master In Slave Out).

Port	Width	Direction	Description
/ss_pad_o	8	Output	Slave select output signals
sclk_pad_o	1	Output	Serial clock output
mosi_pad_o	1	Output	Master out slave in data signal output
miso_pad_i	1	Input	Master in slave out data signal input

Figure 5 : SPI external connections

Here, SCK [sck_o] is generated by the master device and synchronizes data movement in and out of the device through the MOSI [mosi_o] and MISO [miso_o] lines. The SPI clock is generated by dividing the WISHBONE clock [clk_i]. The division factor is software programmable. The Master Out Slave In line is a unidirectional serial data signal. It is an output from a master device and an input to a slave device. The Master In Slave Out line is a unidirectional serial data signal. It is an output from a slave device and an input to a master device. The signal ss_pad_o is of output signal with 8 bit and it selects the slave output signals. Based on this only the slave is selected.

V. ARCHITECTURE

We have already known about wishbone interface signals from above. The timing generator generates the clock. We use different types of registers like SPCR(Serial Peripheral Control Register),SPER(Serial Peripheral Extension Register),SPSR(Serial Peripheral Status Register) and SPDR(Serial Peripheral Data Register).

Data register has 2 registers, Transmitter register and Receiver register.Write buffer which writes the data into the shift register and read buffer which reads the data from the shift register.Status register shows the status fifo whether it is full or empty or upto how much it is filled.Control register controls all the signals from the wishbone interface.It has registers like ASS ,IE ,LSB ,Tx_NEG ,Rx_NEG ,GO_BSY, CHAR_LEN.

In ASS if this bit is set, ss_pad_o signals are generated automatically. This means that slave select signal, which is selected in SS register is asserted by the SPI controller, when transfer is started by setting CTRL[GO_BSY] and is de-asserted after transfer is finished. If this bit is cleared, slave select signals are asserted and de-aserted by writing and clearing bits in SS register. In IE if this bit is set, the interrupt output is set active after a transfer is finished.

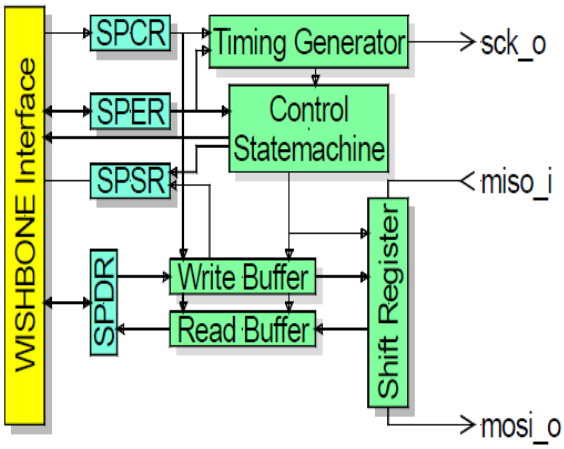


Figure 6 : Architecture of Serial Peripheral Interface

The Interrupt signal is deasserted after a Read or Write to any register.In LSB If this bit is set, the LSB is sent first on the line (bit TxL[0]), and the first bit received from the line will be put in the LSB position in the Rx register (bit RxL[0]). If this bit is cleared, the MSB is transmitted/received first (which bit in TxX/RxX register that is depends on the CHAR_LEN field in the CTRL register).In Tx_NEG if this bit is set, the mosi_pad_o signal is changed on the falling edge of a sclk_pad_o clock signal, or otherwise the mosi_pad_o signal is changed on the rising edge of sclk_pad_o.In Rx_NEG if this bit is set, the miso_pad_i signal is latched on the falling edge of a sclk_pad_o clock signal, or otherwise the miso_pad_i signal is latched on the rising edge of sclk_pad_o.In GO_BSY Writing 1 to this bit starts the transfer. This bit remains set during the transfer and is automatically cleared after the transfer finished. Writing 0 to this bit has no effect. CHAR_LEN specifies how many bits are transmitted in one transfer. Up to 64 bits can be transmitted.

We also have one divider register DIVIDER. The value in this field is the frequency divider of the system clock wb_clk_i to generate the serial clock on the output sclk_pad_o. The desired frequency is obtained according to the following equation:

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

VI. VERIFICATION PLAN

Simulation and verification is a very important part of IP core design, because it directly related to the availability of the IP. We simulate our design in modelsim and do verification using system verilog language. To verify SPI data transmission the randomized signal of wishbone interface , control register , divider register and slave register are generated by calling system inbuilt randomize function.It is done by Generator.

The driver part of verification deals with the configuration of several registers which are used in data transmission. Firstly, Slave select register is configured to select slave registers of receiver. Secondly, divider register is configured to generate serial clk from wb_clk_in of wishbone interface.Thirdly, control register of SPI core is configured to transmit data depending on character length, LSB,Tx_Neg etc parameters affecting data transmission.For data transmission, the 8th bit of control register "Go_busy" is to set '1' after register configuration.Two events are triggered one after ss configuration and one after data transmission.

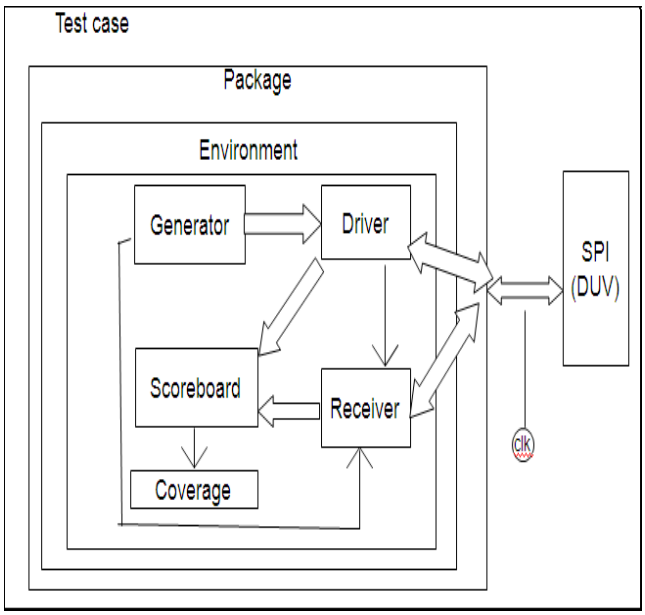


Figure 7: Verification methodology in system verilog

The receiver part of verification deals with driving of MISO depending on different events. The scoreboard part of verification deals with comparision of master data and slave data. The coverage model of our design consists of several cover points and cross cover points which checks all the possible scenarios of input parameter which are affecting data transmission.The cover points are divider,ass,lsb,Tx_Neg,Rx_Neg,data_in, character length also above cover points are crossed together to get maximum coverage report.

To verify SPI Protocol we have used two different types of test cases, they are:Directed Test Case and Regression test Case. In directed test case, we are providing predefined stimulus with required constraint to verify expected output.In regression test case, we are extending base class(transactor class) to different child classes where we are constraining different input parameters to check all possible cases.

VII. SIMULATION RESULTS

When simulated in modelsim we get the output data as the data given as input data.

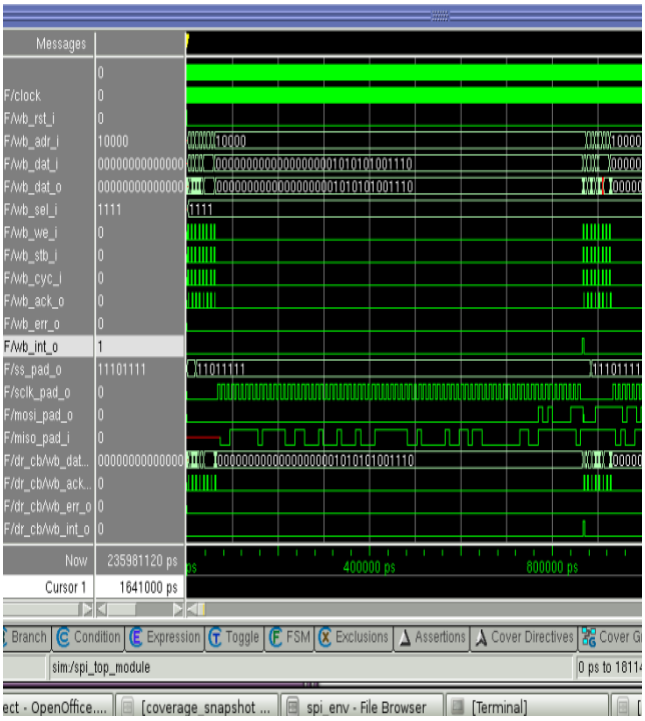
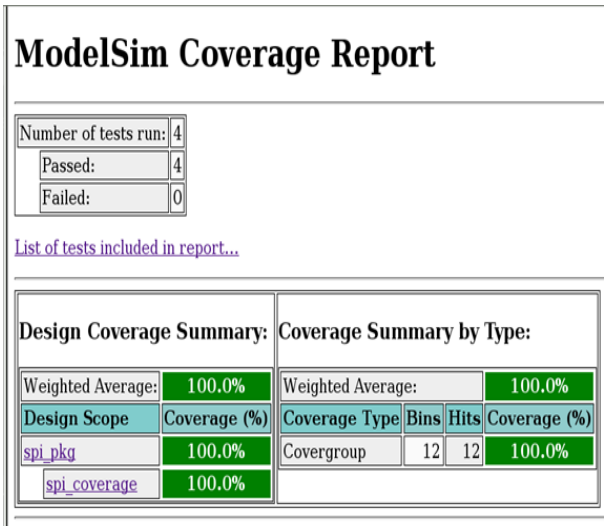


Figure 8: output waveform of SPI master-slave core



ModelSim Coverage Report

Number of tests run:	4
Passed:	4
Failed:	0

List of tests included in report...

Design Coverage Summary:		Coverage Summary by Type:			
Weighted Average:	100.0%	Weighted Average:			100.0%
Design Scope	Coverage (%)	Coverage Type	Bins	Hits	Coverage (%)
spi_pkg	100.0%	Covergroup	12	12	100.0%
spi_coverage	100.0%				

Figure 9: Total functional coverage report

Covergroup type: spi_fcov				100.0%
Coverpoints / Bins	**At Least**	**Hits**	**Goal**	**Coverage**
Coverpoint: TX_RX (covered 2 of 2 bins)			100.0%	100.0%
bin low	1	2014		Covered
bin high	1	2014		Covered
Coverpoint: LSB (covered 2 of 2 bins)			100.0%	100.0%
bin low	1	2000		Covered
bin high	1	2054		Covered
Coverpoint: CHAR_LEN (covered 3 of 3 bins)			100.0%	100.0%
bin low	1	34		Covered
bin mid	1	3981		Covered
bin high	1	39		Covered
Coverpoint: WB_DATA (covered 3 of 3 bins)			100.0%	100.0%
bin low	1	9		Covered
bin mid	1	4045		Covered
bin high	1	3		Covered
Coverpoint: Divider (covered 2 of 2 bins)			100.0%	100.0%
bin low	1	1056		Covered
bin mid	1	2998		Covered

Figure 10 : Individual functional coverage report

VIII. CONCLUSION

In this paper, we have designed the SPI Master-Slave core based upon design-reuse methodology. We simulated this design in ModelSim using Systemverilog. We got the output data same as the input data. Further, we have also done functional verification and achieved 100 percent functional code coverage for our design.

REFERENCES

[1] www.opencore.org.Simon Srot. "SPI Master Core Specification",Rev.0.6. May 16,2007.
[2] "Design and Implementation of a Reused Interface" 978-0-7695-3887-7/09/$26.00 ©2009 IEEE.
[3] Wikipedia, the free encyclopedia, "Serial Peripheral Interface Bus",
Available http://en.wikipedia.org/wiki/Serial_Peripheral_Interface_Bus.
[4] Tianxiang Liu "IP Design of Universal Multiple Devices SPI Interface" 978-1-61284-632-3/11/$26.00 ©2011 IEEE.
[5] *Specification for the:"WISHBONE System-on-Chip (SoC) Interconnection Architecture for Portable IP Cores"*Revision: B.3, Released: September 7, 2002.
[6] Chris spear "System verilog for verification" second edition.

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

K.Aditya was born in vijayawada,krishna(Dist.), AP, India. He received B.Tech in E.C.E from JNTUniversity, Anantapur,A.P,india. Pursuing M.Tech in VLSI from KL University. His research interest includes Low Power design of VLSI circuits.Functional verification of Digital circuits.

M.Sivakumar was born in Vijayawada, Krishna(Dist.,), AP, India. He received B.E in E.C.E from Periyar University Dharmapuri, Tamilnadu, India, M.Tech from Bharath University, Chennai, Tamilnadu. His research interest includes LowPower and design of fault model circuits.

Dr. Fazal Noorbasha, Presently working as an Assistant Professor, Department of Electronics and Communication Engineering, KL University, Guntur, Andhra Pradesh, India, where he has been engaged in teaching and research, VLSI Research Group Head, Department Curriculum Committee (DCC) Member. His interest of research and development is Low-power VLSI, High-speed CMOS VLSI SoC, Memory Processors LSI's, Digital Image Processing, Embedded Systems and Nanotechnology. He has published and presented over 35 Science and Technical papers in various International and National reputed journals and conferences. He is a Scientific and Technical Committee & Editorial Review Board Member in Engineering and Applied Sciences of World Academy of Science Engineering and Technology (WASET), Advisory Board Member of International Journal of Advances Engineering & Technology (IJAET), Life Member of Indian Society for Technical Education (ISTE-India), Member of International Association of Engineers (IAENG-China) and Senior Member of International Association of Computer Science and Information Technology (IACSIT-Singapore).

T. Praveen Blessington, Presently working as an Associate Professor & research scholar in Department of ECE, KL University, Guntur, Andhra Pradesh, India, where he has been engaged in teaching and research in VLSI & embedded designs. He is a member of VLSI Research Group, Department Curriculum Committee (DCC) Member. His interest is research and development in SOC, NOC Architectures, Low-Power VLSI & Embedded Systems. He has published and presented various International and National reputed journals and conferences. He is a life member of IETE, ISTE and SCIEI.