RESEARCH ARTICLE OPEN ACCESS

Generation of Graph for APB via SPI Verification Using Trek

Gowtham R*, Govindaraj V**
*(Department of Electronics and Communication Engineering, KPR Institute of Engineering and Technology, Coimbatore,)
** (Department of Electronics and Communication Engineering, KPR Institute of Engineering and Technology, Coimbatore,)

----------**********----------

Abstract:

In VLSI Technologies, according to Moore's law the density of a semiconducting material increases rapidly. The verification technology in this industry is complex. For verification process, manually generating test cases is difficulty and time consuming process for verifying complex problem in design. For reducing complexity of the verification process is to generate a graph scenario model for each Intellectual Property (IP) blocks in system on chip (SOC) design. The main objective is to verify the IP block of AMBA (Advanced Microcontroller Bus Architecture) APB (Advanced Peripheral Bus) via SPI (Serial Peripheral Interface) using graph based scenario model. This concept includes generation of graph of APB_WB (Advanced Peripheral Bus Wishbone) and integration of test cases generated by the graph to APB. The graph is generated by the software called trek by Breker verification systems. This software automatically generates test cases. These test cases are self-verifying. Trek takes input information from scenario models describing the desired outcomes, developed by the user which is integrated with DUT (Device Under Test) and outcome is verified by Trek mailbox. For more accuracy of SPI protocol, the test cases has to be generated for transmit register using trek.

Keywords — AMBA_APB, SPI, Signal, Transfers, Graph Scenario model, Trek.

----------**********----------

I. INTRODUCTION

The role of graph-based scenario models can be explained by example digital camera SOC design. The images are captured from lens in the camera block. The images can be displayed for the user, driven by in the serial process of photo processor, transmitted via a USB port, and saved to an SD card. A series of such images may be treated as a video stream and handled similarly by the video processor. Each block consist an individual IP in the SOC. The verification team must understand all of the data flows and all possible interactions for each block. if it is to develop a testbench environment. By creating testbench using hand written C test cases is time consuming process. So the graph scenario model is to generate scenario model for each block in the IP design is for verification. One of the major blocks is APB_SPI bus for fast communication between each part in the design.

II. AMBA APB SIGNALS

PCLK signal is the clock indicates that times all transfers on the APB. PRESET signal is reset which indicates that the APB reset signal is active low. This signal is normally connected directly to the system bus reset signal. PADDR is an address signal which represents 32 bits driven by bus peripheral unit. PPROT is a protection unit which indicates the normal, privileged, or secure protection level of the transaction and whether the transaction is a data access or an instruction access. PSEL is a select signal which generates this signal to each peripheral bus slave it indicates that the slave device is selected and that a data transfer is required. There is a PSEL signal for each slave. PENABLE is an enable signal indicates the second and subsequent cycles of an APB transfer. PWRITE indicates direction. An APB writes access when high and an APB. PWDATA is a write data signal. This bus is driven by the peripheral bus bridge unit during write cycles when PWRITE is high. This bus can be up to 32 bits wide. PSTRB is write strobe

signal indicates which byte lanes to update during a write transfer. PREADY is Ready signal. The slave uses this signal to extend an APB transfer. PRDATA is data signal which drives this bus during read cycles when PWRITE is low. This bus can be up to 32-bits wide PSLVERR is an error signal which indicates a transfer failure. APB peripherals should not support the PSLVERR pin. It is true functionality for both existing and new APB peripheral designs. For a case of peripheral does not include this pin then the appropriate input to the APB Bridge is combined with low.

III. AMBA APB SIGNAL TRANSFERS

To begin with the transfer of signal contains read and write. The read transfer has two states which is wait state and no wait state. The write transfer also has two states which is wait state and no wait state.

A. Write Transfer with No Wait State

When PENABLE asserted, indicates the start of the Access phase of the transfer. When PREADY asserted indicates that the slave can complete the transfer at the next rising edge of PCLK**.** The address PADDR, write data PWDATA, and control signals all remain valid until the transfer completes at the end of the Access phase. The enable signal PENABLE, is de-asserted at the end of the transfer. The select signal PSEL is also de-asserted unless the transfer is to be followed immediately by another transfer to the same peripheral.

B. Write transfer with Wait States

During an Access phase, when PENABLE is high, the slave extends the transfer by driving PREADY low. The following signals remain unchanged while PREADY remains low. When PENABLE is low, this ensures that peripherals have a fixed two cycle access can tie PREADY is high. PREADY can take any value.

C. Read Transfer with No Wait States

The timing of the address, writes, select, and enable signal. The slave must provide the data before the end of the read transfer.

D. Read transfer with wait states

Read transfer with wait transfer has PREADY signal can extend the transfer. The transfer is extended if PREADY is driven low during an Access phase.

IV. GENERATION OF GRAPH

A. Amba _Apb Graph

For the generation of graph, the operating states of APB has three states which is idle, setup, access. Idle state is a default state of APB. When a transfer is required the bus moves into the SETUP state, where the appropriate select signal, PSEL, is asserted. The bus only remains in the SETUP state for one clock cycle and always moves to the ACCESS state on the next rising edge of the clock. The enable signal, PENABLE, is asserted in the ACCESS state. The address write, select, and write data signals must remain stable during the transition from the SETUP to ACCESS state. The graph contains three states for each process of test cases. The test cases are generated for read and write transfer. The read transfer is again has the test cases via wait and no wait states. Similarly the write transfer is again wait and no wait states. In operating states, the idle state has both PSEL and PENABLE is zero. The setup state has PSEL value is one and PENABLE value is zero. The access phase has both PSEL and PENABLE value is one.

```
trek: info: Created 'trek_work/Makefile'
trek: info: Running 'make  -f trek_work/Makefile'
trek: compile: make[1]: Entering directory `/home/gowtham/Trek/apb/run'
trek: compile: /home/gowtham/tools/trek-4.0.16_20140415_32b/bin/trekcc -I ../lib
 -o trek_work/sm001.treko ../graph/apb_graph.trek
trek: compile: /home/gowtham/tools/trek-4.0.16_20140415_32b/bin/trekcc -I ../lib
 -o trek_work/sm002.treko ../graph/apb_top.trek
trek: compile: make[1]: Leaving directory `/home/gowtham/Trek/apb/run'
trek: info: Compile successful
trek: info: /home/gowtham/tools/trek-4.0.16_20140415_32b/libraries/trek/io_servi
ces/graph/io_services.vso: found 19 goals
trek: info: trek_work/sm001.treko: found 0 goals
trek: info: trek_work/sm002.treko: found 36 goals
trek: info: ts0.iteration_count = 1
trek: info: ts0: switching to thread_id 1
trek: info: new scenario on thread_id 1
trek: info: apb0.error: PSEL Value 0x1
trek: info: apb0.error: PENABLE Value 0x1
trek: info: apb0.error: PSLVERR Value 0x1
trek: info: apb0.error: PREADY Value 0x1
trek: info: ts0: switching to thread_id 1
trek: info: ts0: completed thread_id 1 after iteration 1
trek: info: ts0: completed all threads
```

Fig 1. Values from APB Graph

Fig 1 shows that values are obtained from APB graph. The values can be used further for integration to the design of APB. APB has two transfers which is read and write goal in the phase of select goals. The read has wait and no wait state goals which have operating states and error signals. The write has also wait and no wait states. These wait state ends with operating state and error signals. Fig 2 shows the APB Graph is generated from trek.

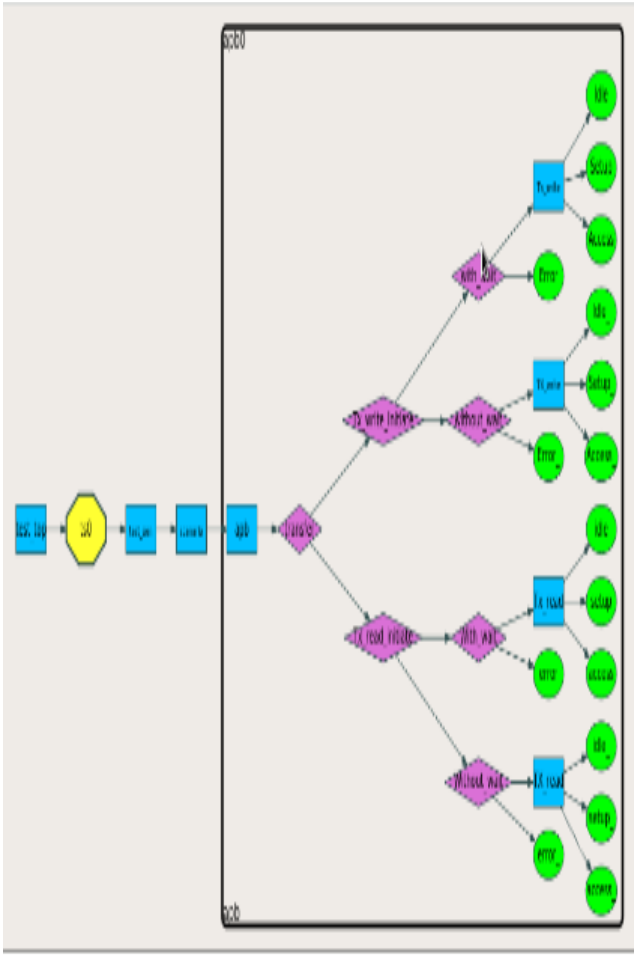

Fig 2. Graph for APB

B. Transfer Register for SPI

For SPI, The transfer register has to select the transmitting value from the properties like Character length, Endian values, Interrupt values, Clock edge values, Slave select mode.

The character length values vary from 0, 2, 4, 8, , , , 128. The endian goal has to select whether little or big. Interrupt goal is to select whether enable or not. Clock edge goal is to select the edge values vary between positive and negative or both positive and negative. Slave select mode goal is to select mode which is either auto or manual.

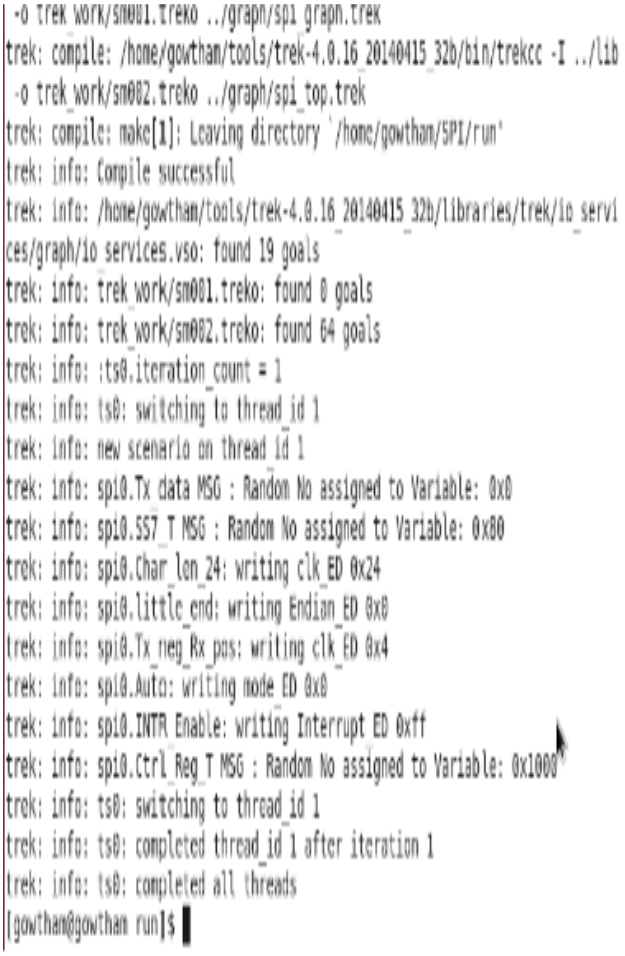

Fig 3. Values from SPI Graph

Fig 3 represents that values are obtained from APB graph. The values can be used further for integration to the design of SPI. This transfer graph has the value of control register and slave select register according to register specification. The transfer graph has to merge with SPI graph for Verifying APB_SPI verification. For generating more accurate test cases it is necessary to have transfer register value test cases.

V. CONCLUSION

APB_SPI protocol has high transmission speed, ease of implementation and with pins advantages. The APB_SPI can connect as many devices as many pins we have on the main microcontroller. Basically the function and operation of SPI and description of registers, signals, pin are discussed. For APB_SPI model, the graph is generated with s transmit register values and writing register values for slave select, control and status and enable clock using Trek software. The APB_SPI protocol graph gives more accurate test cases. The main objective of using Trek software is to generate test cases. So these test cases for APB_SPI can be useful for SOC (which has SPI protocol) verification.

REFERENCES

[1]. Roychoudhery .A, Mitra .T, Karri .S.R, " Using Formal Techiques To Debug AMBA System-On-Chip Bus Protocol",(IEEE) ISSN: 1530 1591, Page 828-833, 2003.

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

[3]. Oudijida AK, Berrandjia ML, LiachanA, Tiar R, "Design and test of general purpose SPI Master/Slave IPs on OPB bus",ISBN:978-1-4244-7532-2, Pages :1-6.

[4]. Zhili zhou, Zhang xie, Xinan wang, Teng wang, "Development of verification environment for SPI master interface using System Verilog"(IEEE) ISSN:2164 5221, Pages 2188-2192.

[5]. Simon Srot "SPI Master Core Specification", Rev.0.6. March 15, 2004.

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