IARJSET

ISSN (Online) 2393-802
ISSN (Print) 2394- 588

International Advanced Research Journal in Science, Engineering and Technology

ISO 3297:2007 Certified

Vol. 3, Issue 8, August 2016

Abstract, Correct By Construction and Faster Register Modeling of AMBA APB Bus

Kiran J P [1], Dr. R Jayagowri[2]

Master of Technology, VLSI Design and Embedded Systems, BMSCE, Bangalore, India[1]

Associate Professor, Electronics Department, BMSCE, Bangalore, India[2]

Abstract: The SoC (System on Chip) uses AMBA APB as an on chip bus.APB is low bandwidth and low performance bus used to connect the peripherals like UART, Keypad, Timer and other peripheral devices to the bus architecture. This paper describes the design generation of AMBA APB (Advanced Peripheral Bus) protocol using Perl scripting language. Here main aim is to reduce human interface in design part so we can reduce common syntax errors. Per generates the Verilog design code of APB slave and its corresponding test bench, where all its specifications are there in XML script. This code is simulated in QuestaSim. Finally wave forms and code coverage reports are analyzed.

Keywords: AMBA, APB, Perl, SoC, XML, XLS

I. INTRODUCTION

Earlier AMBA buses were mainly used for in microcontroller devices but now it is widely used in large range of ASICs and SoC parts including the application processors used in modern portable mobile devices like smartphones. AMBA is an open standard, on-chip interconnect specification for the purpose of connecting and managing functional blocks in a System-on-Chip (SoC).

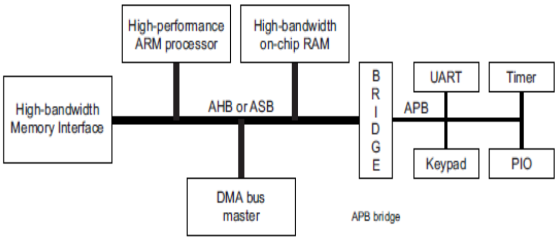

High-performance ARM processor
High-bandwidth on-chip RAM
High-bandwidth Memory Interface
AHB or ASB
B R I D G E
UART
Timer
APB
Keypad
PIO
DMA bus master
APB bridge

FIG. AMBA Bus Architecture

So, for APB the bridge acts as the master and all the devices connected on the APB bus acts as the slave. The component on the high performance bus initiates the transactions and transfer them to the peripherals connected on the APB. So, at a time the bridge is used for communication between the high performance bus and the peripheral devices.

As seen in the Figure. , AMBA bus architecture consists of three components, namely Advanced High Performance Bus (AHB), Advanced System Bus (ASB), Advanced Peripheral Bus (APB). AMBA AHB or ASB is high performance bus and has higher bandwidth. So the components requiring higher bandwidth like High Bandwidth on chip RAM high-performance ARM processor are connected to AHB/ASB. So, the components requiring lower bandwidth like the peripheral devices such as UART, Keypad, Timer and PIO (Peripheral Input Output) devices are connected to the APB.

II. APB DESIGN

The APB is the member of the AMBA 3 protocol famil which implements a low cost interface which minimizes the power consumption and reduces the interface complexity. Since APB has unpipelined protocol. Therefore, it interfaces to the low bandwidth peripherals that do not demand the high performance of the pipelined bus interface. All the signal transitions are associated with the rising edge of the clock which makes it simple to integrate APB peripherals into any design flow.

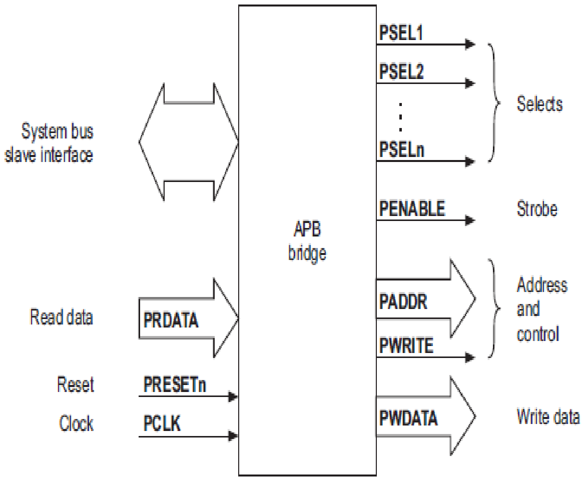


FIG. 3 APB Slave

III. PERL

Perl a scripting language is used here to extract the design specification from XML where it is in specified formate to XLS spreadsheet. This XLS Spreadsheet is accessed to generate APB slave design code and its corresponding test bench. Here we need to develop two Perl script one to extract specification from XML to XLS and another to generate Verilog design file by accessing XLS.

DOI 10.17148/IARJSET.2016.3818

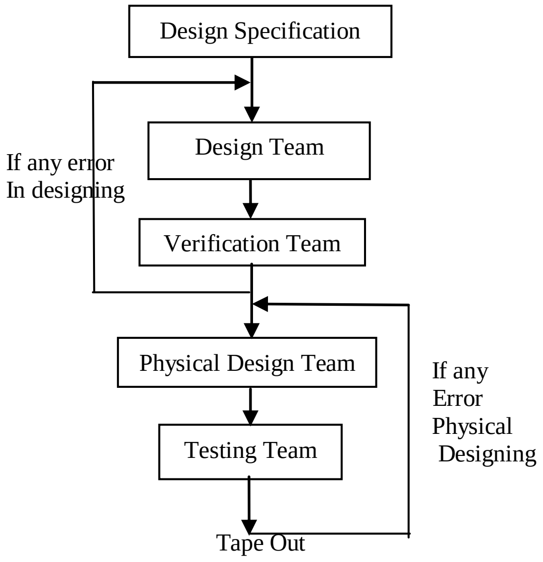
IARJSET

ISSN (Online) 2393-802
ISSN (Print) 2394- 588

International Advanced Research Journal in Science, Engineering and Technology

ISO 3297:2007 Certified

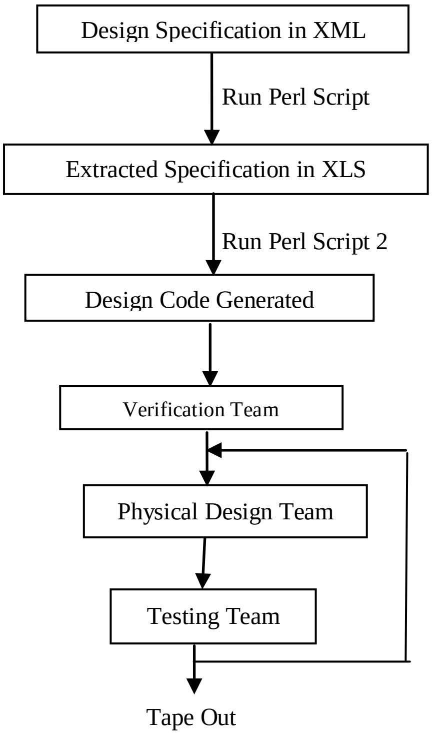
Vol. 3, Issue 8, August 2016



Design Flow Before

We have used three Perl modules to do this operation. They are
XML: Simple: This module helps in reading XML script to extract the specification for our design.
Spreadsheet: Write Excel: This module helps in writing the extracted spec. into XLS spreadsheet in required format.
Spreadsheet: Parser Excel: This module helps in accessing the XLS sheet to generate APB verilog design file and test bench file.

By using proposed method



Design Flow After

Here we are not getting any error after the design is done because it it generated from Perl script not by manually. At first we need to write Perl script for this APB slave design once it is done no need to write the same code again. We can generate the design code without any common syntax errors.

IV. APB OPERATING CYCLE

Figure.3 shows the basic state machine that represents operation of the peripheral bus. There are three states namely, IDLE, SETUP and ACCESS state

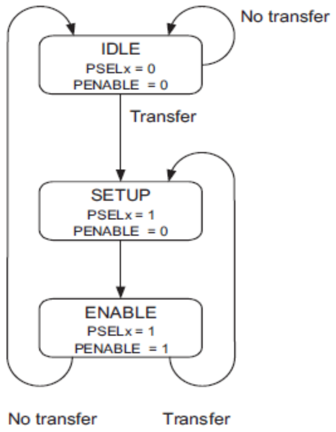


Fig 3: State Diagram

IDLE state is the default state in which no operation is being performed. The assertion of the PSEL signal indicates the beginning of the SETUP phase. The bus enters into the SETUP phase when the data transfer is required. The PWRITE, PADDR and PWDATA are also provided during this phase.

The bus remains in the SETUP phase for one clock cycle and on the next rising edge of the clock, the bus will move to the ACCESS state. The assertion of the PENABLE signal indicates the start of the ACCESS phase. All the control signals, address, and the data signals remains stable during the transition from the SETUP phase to the ACCESS phase.

In case of read operation the PRDATA is present on the bus during this phase. PENABLE signal also remain high for one clock cycle. If no further data transfer is required, the bus will move the IDLE state. But, if further data transfer is required then the bus will move to the SETUP phase.

Write Cycle
During the write transfer operation, the PSEL, PWRITE, PADDR and PWDATA signals are asserted at the T clock edge which is called the SETUP cycle. At the next rising edge of the clock T2, the PENABLE signal and PREADY signal are asserted. This is called the ACCESS cycle. At the clock edge T3, PENABLE signal is disabled and if further data transfer is required, a high to low transition occurs on the PREADY signal.

DOI 10.17148/IARJSET.2016.3818

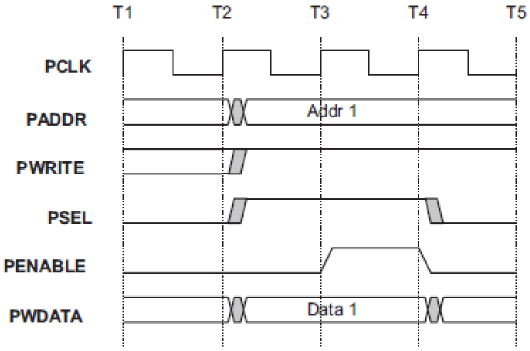


Fig. 4 Write Transfer

Read Cycle

During the read operation, the PSEL, PENABLE, PWRITE, PADDR signals are asserted at the clock edge T (SETUP cycle). At the clock edge T2, (ACCESS cycle), the PENABLE, PREADY are asserted and PRDATA is also read during this phase.

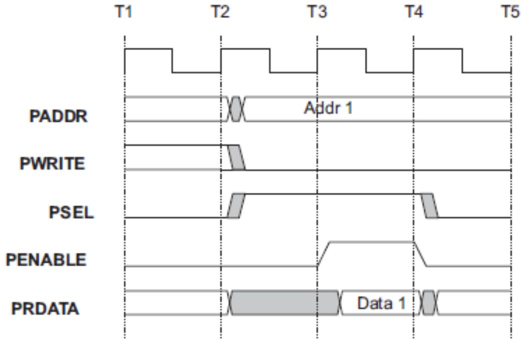
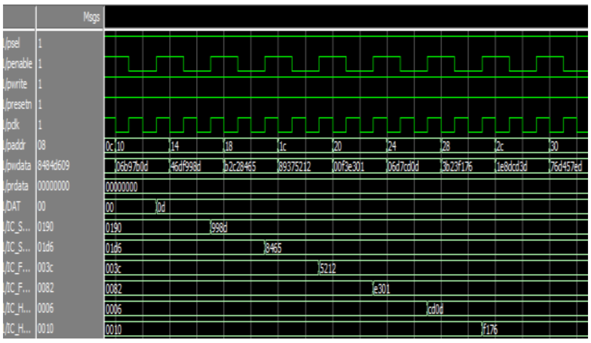


Fig. 5 Read Transfer

V. SIMULATION RESULTS FOR THE GENERATED DESIGN

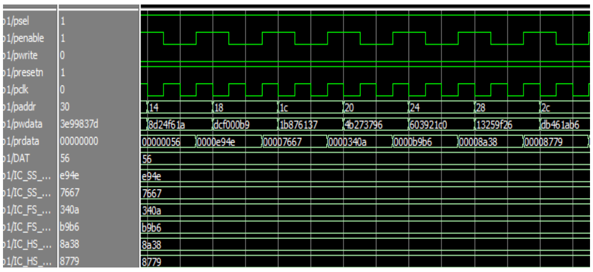
The design and the test bench generated using Perl in Verilog has been compiled using QuestaSim. The results show the write and read operation. It is evident from the figure that when PWRITE= , the address and data are written at the same clock edge. When PWRITE =0, the address is sent on a given clock edge and the data is read on the following clock edge.

Write Cycle



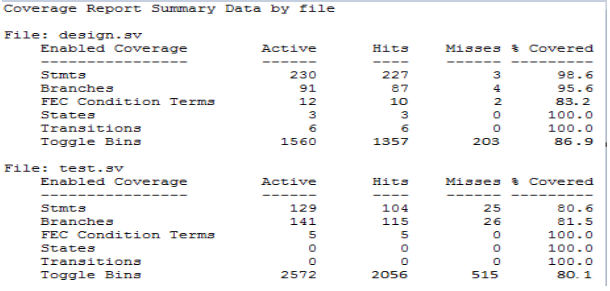
Simulation result for Write Cycle

Read Cycle



Simulation result for Read Cycle

Coverage Report



```
Coverage Report Summary Data by file

File: design.sv
    Enabled Coverage          Active      Hits    Misses % Covered
    ----------------          ------      ----    ------ ---------
    Stmts                        230       227         3      98.6
    Branches                      91        87         4      95.6
    FEC Condition Terms           12        10         2      83.2
    States                         3         3         0     100.0
    Transitions                    6         6         0     100.0
    Toggle Bins                 1560      1357       203      86.9

File: test.sv
    Enabled Coverage          Active      Hits    Misses % Covered
    ----------------          ------      ----    ------ ---------
    Stmts                        129       104        25      80.6
    Branches                     141       115        26      81.5
    FEC Condition Terms            5         5         0     100.0
    States                         0         0         0     100.0
    Transitions                    0         0         0     100.0
    Toggle Bins                 2572      2056       515      80.1
```

Coverage report for design and test file

VI. CONCLUSION AND FUTURE WORK

The advantage of this method is we can develop APB slave of any number of register with in less time. That too without having common syntax errors. Meaning re-designing time of the same code reduces considerably .Also by generating suitable test bench the design code has been verified and validated. We can also check the code coverage of the same design code for its quality analysis. Till now we have used XLS sheet as intermediate between Perl script and Verilog code but now we making script to generate design code directly from using XML script without the interference of XLS sheet.

ACKNOWLEDGEMENT

I would like to thank **Mr. Prabhu Bhairi** from Sevitech Systems, who had been guiding through out to complete the work successfully, and would also like to thank **Dr. R Jayagowri,** Associate Professor, ECE Department for extending help & support in giving technical ideas about the paper and motivating to complete the work effectively & successfully.

REFERENCES

. AMBA Specification (Rev2.0). http://www.arm.com

2. Yi-Ting Lin, Chien-Chou Wang, "AMBA APB bus potocol checker with efficient debugging mechanism", IEEE International Symposium on Huang Circuits and Systems, Page(s). 928 – 93, 2008.

3. M. Dubois, Y. Savaria, "A generic APB bus for implementing high-speed locally synchronous islands", Bois, G. Southeast Conference, Proceedings. IEEE, Page(s). – 6, 2005.

4. ASB Example AMBA System Technical Reference Manual Copyright © 998- 999 ARM Limited.