

Verification of AMBA AHB Bus Protocol by using Different Methodologies - A Review

¹Prathamesh M Soni, ² Prof. Dr. R. M. Deshmukh

¹PG Student Department of EXTC, DRGITR, Amravati

²Prof & Head of Department of EXTC, DRGITR, Amravati

Abstract:- Increasing technology increases the amount of logic that can be placed in a silicon chip driving highly integrated SoC design development. The most relevant factor for a SoC is how well they are interconnected. The SoC uses an on-chip bus architecture. AMBA (Advanced Microcontroller Bus Architecture) bus is most widely used the on-chip bus which is introduced by the ARM. Almost 85-90% of on-chip bus used in any SoC is AMBA (Advanced Microcontroller Bus Architecture). In this project, an AMBA AHB bus Verification environment is built which is verified by using System Verilog Assertions and functional coverage. AMBA protocol (AHB) is verified by achieving successful read & write operations for incrementing burst feature. In this paper we are presenting different comparative methods for IP based Verification of AMBA AHB Protocol by implementing some techniques like Assertions, Constraint random Verification & Coverage report. We are also presenting system Verilog based & Universal Verification based methodologies for verification of AHB protocol. Finally, we have concluded that by observing different methodologies were used and what are the challenges they have encountered and remedies/advancement to overcome/reduced those efforts that we have explained in this review paper.

Keywords. AHB, UVM, Assertions, coverage, CRV, IP, SOC, Arbiter, MUX.

1. INTRODUCTION

An AMBA-based microcontroller typically consists of a high-performance system backbone bus (AMBA AHB or AMBA ASB), able to sustain the external memory bandwidth, on which the CPU, on-chip memory and other Direct Memory Access (DMA) devices reside. This bus provides a high-bandwidth interface between the elements that are involved in the majority of transfers. Also located on the high-performance bus is a bridge to the lower bandwidth APB, where most of the peripheral devices in the system are located (see Figure 1-1).

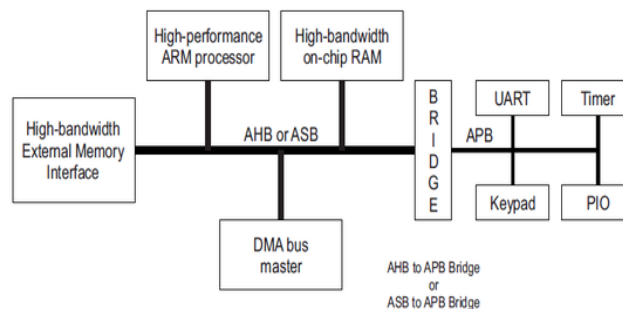


Figure 1.1 A typical AMBA Based Microcontroller

AHB is a new generation of AMBA bus which is intended to address the requirements of high-performance synthesizable designs. It is a high-performance system bus that supports multiple bus masters and provides high-bandwidth operation. AMBA AHB implements the features required for high-performance, high clock frequency systems including:

- burst transfers
- split transactions
- single-cycle bus master handover
- single-clock edge operation
- non-tristate implementation
- wider data bus configurations (64/128 bits).

Bridging between this higher level of bus and the current ASB/APB can be done efficiently to ensure that any existing designs can be easily integrated. An AMBA AHB design may contain one or more bus masters, typically a system would contain at least the processor and test interface. However, it would also be common for a Direct Memory Access (DMA) or Digital Signal Processor (DSP) to be included as bus masters. The external memory interface, APB bridge and any internal memory are the most common AHB slaves. Any other peripheral in the system could also be included as an AHB slave. However, low-bandwidth peripherals typically reside on the APB. A typical AMBA AHB system design contains the following components:

AHB master- A bus master is able to initiate read and write operations by providing an address and control information. Only one bus master is allowed to actively use the bus at any one time.

AHB slave- A bus slave responds to a read or write operation within a given address-space range. The bus slave signals back to the active master the success, failure or waiting of the data transfer.

AHB arbiter- the bus arbiter ensures that only one bus master at a time is allowed to initiate data transfers. Even though the arbitration protocol is fixed, any arbitration algorithm, such as highest priority or fair access can be implemented depending on the application requirements. An AHB would include only one arbiter, although this would be trivial in single bus master systems.

AHB decoder- The AHB decoder is used to decode the address of each transfer and provide a select signal for the slave that is involved in the transfer. A single centralized decoder is required in all AHB implementations.

I.I What is Universal Verification Methodology (UVM):

The **Universal Verification Methodology (UVM)** is a standardized methodology for verifying Integrated Circuit designs. UVM is derived mainly from the OVM (Open Verification Methodology) which was, to a large part, based on the eRM (e Reuse Methodology) for the e verification Language developed by Verisity Design in 2001. The UVM class library brings much automation to the system Verilog language such as sequences and data automation features (packing, copy, compare) etc., and unlike the previous methodologies developed independently by the simulator vendors, is an Accellera standard with support from multiple vendors: Aldec, Cadence, Mentor Graphics, Synopsys, Xilinx Simulator (XSIM).

I.II Why do we need UVM Methodology:

The primary advantage is that the methodology specifies and lays out a set of guidelines to be followed for creation of verification of testbench. This will ensure testbench uniformity between different verification teams, cross compatibility, between IP and standalone environment integration, flexibility and ease of maintaining testbenches. For examples there can be many different ways to implement display messages and verbosity, with different settings such as warning, error & debug. In UVM the underlying reporting mechanism has been standardized and made available so that engineers can focused on most important part of their jobs which is design Verification.

I.III How does UVM help:

Every Verification testbench has a few key components like Driver, Monitor, Stimulus Generator & Scoreboard. UVM provides a base class for each of these components with standardize functions, to instantiate, connect and build the testbench environment. These are the static entities called components in verification environment that exist throughout the simulation just like buildings in a city. These components operate and process on some kind of data that flows around the environment similar to people and vehicles in the city. The data or transaction are called as object or sequence items, since they appear & disappear at various times in the simulation and is more dynamic in nature.

II. DIFFERENT METHODOLOGIES APPLIED FOR VERIFICATION OF AMBA AHB PROTOCOL

The problem of detecting and Verifying Bugs in AHB Protocol verification has been going for a very long time. It has been one of the major concerns of the Semiconductor industry. Normally, IC design verification engineer, special control and protection software systems are responsible for detecting the fault occurrences and isolating the error portion from the system. Thus, it is necessary for the Bugs/error to be detected quickly and precisely. Here are some System Verilog and UVM as verification methodologies were applied by various researchers for Verification of AMBA AHB.

II.I Verification of AHB bus arbiter using system Verilog assertions

The time required for verifying the design is becoming monotonous as the complicity of the chip design is increasing exponentially. Nowadays, about 70% of the design time is needed for developing the verification environment. Hence in this condition, it is important to reduce the verification time in order to speed up the whole development process [1]. Therefore, to improve the design observability and to detect its faults, Assertion Based Verification (ABV) is introduced. ABV is a technique in which assertions are used to grab specific design behavior either through simulation, formal verification or emulation of these assertions.

Here in this paper, they were using round- robin algorithm for designing of arbiter. A round robin arbiter ensures competence among masters and allows vacant time slot to be shared to a master whose round robin turn is later but who is ready now. This simulation result shows that at any one clock cycle only one request can access the grant signal. They were implemented assertions-based technology on arbiter Implementation.

S.No.	PROPERTY DESCRIPTION	STATUS
1.	Master_1 sends a request signal to the arbiter.	PASS

Table 1.1 Test status

Assertion name	Assertion type	Language	Enable	Fail count	Pass count	Active count	Peak memo	Peak Memo time	Cumulative threads
Check for Valid Gnt	concurrent	SVA	on	4	1	1	1448	300 ns	6
Invalid Hmaster	concurrent	SVA	on	7	3	0	968	500 ns	10
Invalid Grant	concurrent	SVA	on	7	2	1	968	800 ns	10
Success Valid Gnt	concurrent	SVA	on	9	1	0	968	500 ns	10

Table 1.2 Concurrent assertions status

II.II VERIFICATION OF AHB PROTOCOL USING SYSTEM VERILOG ASSERTIONS

In this paper the Assertion based verification environment is builder for the AMBA AHB protocol. One Observation is made that assertion verification is faster than the conventional. The idea of this project is to understand and check the working of the protocol. The proposed verification environment is done in system Verilog language construct. The basic feature of AHB like simple read/write, incremental burst read write and multi beat read/write operations are implemented and simulated [2]. And with the help of Scoreboard the depth of functional verification is measured. All the work and simulation are performed with the help of Questa-Sim 10.4e tool (by mentor Graphics).

-----ASSERTION SCOREBOARD-----			
TYPE OF CHECK	TOTAL COUNT	PASS COUNT	FAIL COUNT
error_check	12	12	0
read_only_error_check	2	2	0
basic_write_error	1	0	1
basic_read_error_check	0	0	0
basic_burst_write_check	1	0	1
basic_burst_read_check	0	0	0
HREADY_check	11	10	1
bursts_count_check4	2	2	0
bursts_count_check8	0	0	0
bursts_count_check16	0	0	0
address_change4	5	5	0
address_change8	2	2	0
address_change16	2	2	0

Table 1.3 Scoreboard Output of test pass Vs test failed

II.III THE DESIGN AND VERIFICATION OF AMBA AHB PROTOCOL BY USING SYSTEM VERILOG

In this paper, design of AMBA AHB Protocol using single master single slave, single master multi slave, multi master single slave and mutli master mutli slave was successfully designed using verilog and simulated using Modelsim 10.b software [3]. The proposed method is verification of entire design of AMBA AHB was created using system Verilog. The proposed verification environment comprised of base packet,tx_gen, driver, interface, DUT, monitor, score board were implemented by using OOP concepts. The functionality of the design verified by the system verilog verification environment.

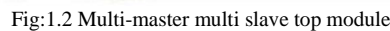
TYPE OF ARBITRATION ALGORITHM	NUMBER OF LUTs USED	DELAY/CYCLE (ns)
Static Fixed Priority Algorithm	4	8.123
Round Robin Algorithm	6	11.798
Modified Round Robin Algorithm	15	18.617

Table 1.4 Comparison of Arbitration algorithm

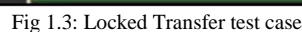
TYPE OF ARBITRATION ALGORITHM	NUMBER OF CLOCK CYCLES TAKEN	DELAY (ns)
Static Fixed Priority Algorithm	Mater 4 gets the grant signal only when the Master 2 disable the request signal	
Round Robin Algorithm	4	47.192
Modified Round Robin Algorithm	2	37.617

Table 1.5 Comparison on delay arbitration algorithm

The design and verification of AMBA multi master AHB using system verilog is successfully completed. Universal verification method (UVM) is used for verification. First step of our design is to construct separate system verilog code for simplest transfer and complex master-slave transfer. From this step it is observed that simplest transfer has no need of arbitration mechanism, but multi master multi slave operation followed by a "First-in, first out"(FIFO) arbitration scheme with split and wait signal [4]. Waveform simulation take place at 100ns this is because of advantage of system verilog language. The second stage of design is Verification and functional coverage.



This project is aimed at performing Verification of multi master amba ahh protocol using System Verilog and UVM methodology [5]. Test cases were coded to check each of the features of AHB including multi master operation, single and all types of burst operations both read and write, locked transfer operation, error case like invalid address. Coverage model was also implemented to analyse the functional coverage report. Overall average functional coverage value was obtained as 82.95%. This highlights few limitations of DUT and calls for improvement of the DUT. Here we have 3 masters requesting the bus with priorities in the order master 0 > master 1 > master 2. Even during a transaction by a master of lower priority if higher priority master requests for the bus the transaction will be stopped temporarily and grant will be assigned to higher priority master. The lower priority master will regain the access of the bus only after this master transaction gets completed.



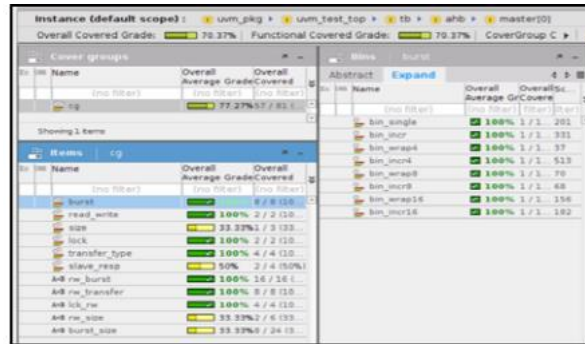


Fig1.4: cover points analysis report

II.V VERIFICATION OF AHB PROTOCOL USING UVM

The results of verification components such as Master Agent and the Slave Agent of the UVM Environment are presented. According to Test Plan, the test cases are verified by developing the Verification IP for the AHB Protocol. The Test Cases are written in the form of sequences in the Sequencer using System Verilog. The sequencer drives the sequences to the driver and thereby to Score Board. Verification of AHB Protocol for Single Master-Single Slave has been verified by developing the Verification IP using the UVM methodology [6]. UVM used System Verilog implementation of standard TLM interfaces for modular communication between Verification Components. Coding of AHB components such as Master Agent and Slave Agent has been done in System Verilog using UVM. The Verification IP is developed using tool Questa and it is sure that developed VIP is also compatible with other tools such as Cadence, Synopsys etc. The developed Verification IP can be used in the verification of SoCs.



Fig1.5: R/W transaction of incrementing Burst 16

III. OBSERVATION

As per the survey, there were so many researchers were put their efforts in the domain of VLSI Verification by Universal Verification Methodology (UVM) and System Verilog Verification methodology in last few years. We may observe that, most of the researchers focused on AHB2.0, arbitration interface, verification using round robin fashion, and priority-based algorithm. Some of research work found in Verification of Single/multi-Master-Single/multi-Slave Verification by System Verilog and UVM also. Some methods have its own advantages and disadvantages, we found that somewhere it is required to implement more features for Protocol efficiency, latency, maximum throughput etc. we also observed that if we considered some specific standards while doing design and verification it would get a better accuracy, precision and efficiency. By considering:

Choice of system- bus Both AMBA AHB and ASB are available for use as the main system bus. Typically, the choice of system bus will depend on the interface provided by the system modules required. The AHB is recommended for all new designs, not only because it provides a higher bandwidth solution, but also because the single-clock-edge protocol results in a smoother integration with design automation tools used during a typical ASIC development.

When to use AMBA AHB/ASB or APB A full AHB or ASB interface is used for:

- bus masters
- on-chip memory blocks
- external memory interfaces
- high-bandwidth peripherals with FIFO interfaces
- DMA slave peripherals.

A simple APB interface is recommended for:

- simple register-mapped slave devices
- very low power interfaces where clocks cannot be globally routed

- grouping narrow-bus peripherals to avoid loading the system bus.

IV. DISCUSSION

By going through the different scenarios, we may conclude that, by adding some features, and few points taken into consideration as we have discussed in observation, we can give a better observability, more efficiency, Bugs free codes, good coverage report. Verification by Universal verification methodology (UVM) gets best results over System Verilog as they have some advanced features like concept of Factory, TLM ports, resource db, config db etc.

So, by considering all above factors we will be able to implement an AHB Protocol Verification system having some additional features and also, we are focusing to improve its capacity of transferring data in more efficient manner. By implementing:

- 1) AHB 5.0 version implementation
- 2) Split and Retry response functionality
- 3) Verification of AHB interconnect (arbiter, decoder and mux)
- 4) Verification of AHB Universal verification component (UVC's)
- 5) 16 master and 16 slaves can be connected over AHB Bus
- 6) This verification environment can be fitted in any AMBA based parallel processing protocol eg. AXI.

REFERENCES

- [1] Verification of Advanced High Performance Bus Arbiter using System Verilog Assertion, International Journal of Engineering Trends and Technology (IJETT) – Volume 37 Number 5 - July 2016. By Prince Gurha , Dr.(Mrs.) R. R. Khandelwal Dept. of Electronics R.C.O.E.M, Nagpur.
- [2] VERIFICATION OF AHB PROTOCOL USING SYSTEM VERILOG ASSERTIONS, Vol-4 Issue-5 2018, IJARIIE-ISSN(O)-2395-4396 by Nikhil B. Gaikwad , Vijay N. Patil.
- [3] The Design and Verification of AMBA AHB Protocol by using System Verilog, International Journal of Electronics, Electrical and Computational System IJEECS ISSN 2348-117X Volume 7, Issue 4 April 2018 by Keerthana Keerthu.
- [4] PERFORMANCE VERIFICATION OF AMBA MULTI-MASTER AHB BUS USING SYSTEM VERILOG, International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056 Volume: 06 Issue: 06 | June 2019 www.irjet.net p-ISSN: 2395-0072. By Deepthi V. D M. Tech student, VLSI&ES, Govt. Engineering College Idukki, Kerala, India
- [5] MULTI-MASTER AMBA AHB PROTOCOL VERIFICATION USING TLM BASED UVM ENVIRONMENT, Int. J. Chem. Sci.: 14(S3), 2016, 769-776 ISSN 0972-768X www.sadgurupublications.com. By SINNY P. SUNNY, S. SIVANANTHAM* and C. PRALINE RAJABAI Department of Micro and Nanoelectronics, School of Electronics Engineering, VIT University, VELLORE (T.N.) INDIA.
- [6] Verification of AHB Protocol using UVM, International Journal of Engineering Research & Technology (IJERT) ISSN: 2278-0181 Vol. 3 Issue 6, June – 2014. By Tejaswini H N 1 Asst. Prof. Dept. of ECE Revati Bothe2 Sambhram Inst of Technology, Team leader, Ravishankar C V 3 Bangalore, India SmartPlay Technologies Prof. Dept. of ECE Bangalore, India Sambhram Inst of Technology, Bangalore, India.