iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 9 **Issue:** II **Month of publication:** February 2021

DOI: https://doi.org/10.22214/ijraset.2021.33120

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue II Feb 2021- Available at www.ijraset.com

A Review on AMBA AHB Lite Protocol and Verification using UVM Methodology

Someshvar S[1], Ohileshwari MS[2]
[1]*Electronics and communication, RNSIT, India*
[2]*Electronics and communication, RNSIT, India*

Abstract: ***AMBA are the most widely used protocols in soc designs and microcontrollers, they support the interface of modules on a microcontroller for intercommunication, different types of systems based on their performance are supported by AMBA protocols and there are various types of AMBA protocol based on their features and advancements. and verification is a important aspect in the VLSI domain to verify whether the designed circuit works according to the specification and requirement it plays a major role in the productivity of a semiconductor industry, so in this paper we have reviewed various AMBA protocols its features, specifications to have a idea of their usage in microcontrollers and the design of AMBA AHB-lite protocol which describes its operation and features. We have also described about the need of Verification its evolution and the Universal verification methodology (UVM) it's features and advantages over conventional verification methods.***
Keywords: AMBA, AHB-lite, UVM, Functional Verification

I. INTRODUCTION

Communication protocols in microcontrollers have evolved a long way, right from UART to AMBA protocols, Communication protocols are a medium for communication for the systems present in a microcontroller with some set of assigned rules. Advanced microcontroller bus architecture AMBA is a chip interconnect which connects and manages systems or functional blocks in a microcontroller [1].

AMBA protocols are the most widely used protocols in the microcontrollers nowadays, as they have many features and different versions based on their specifications.

They are even used in nowadays smartphones, gadgets and a wide range of system on chips. AMBA family consists of many versions of AMBA protocol they will be classified in this paper further. AMBA AHB- lite is one of them which is a high performance bus in the AMBA family hence the name Advanced high performance bus, here lite is one of the version of AHB which has only one master and multiple slaves.

AMBA is widely adopted by many of the industries because of its variety and different versions which can be used across various microcontrollers [2], AMBA is more compact and flexible then compared to other protocols hence it is more widely used. Verification is the most important aspect in the VLSI domain [8], all the soc designs which have been designed has to be verified in order to check its functionality, weather it performs according to the given specifications or not.

Verification increases the productivity of a semiconductor industry but the verification process should be fast and efficient, therefore to obtain efficient verification we use UVM (universal verification methodology) UVM is a standard methodology which has to be followed in the verification process it is fast, reusable, efficient and also portable compared to other verification process. In this paper we give a overview on the types of AMBA protocols and discuss AHB-lite in detail. And discuss about verification of designs, evolution of verification and importance of UVM in industries.

II. BACKGROUND

A. AMBA Classification

AMBA protocols are classified into various types based on their specifications, By these number of classifications AMBA protocols are used across various microcontrollers and SOC's making them platform independent and to be used with any processor architecture [9].

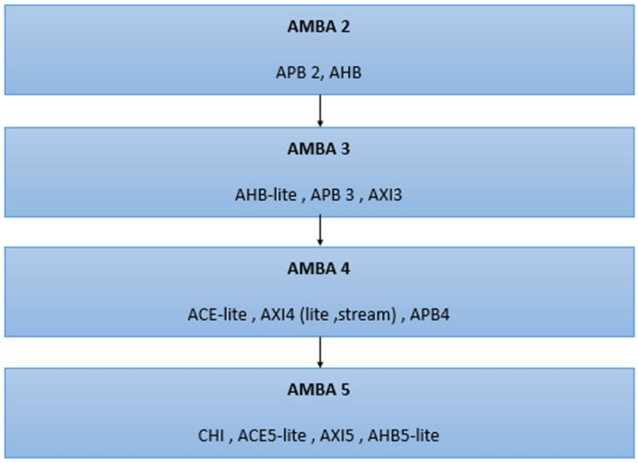


Figure 1: AMBA classification

Figure 1 shows the different types of AMBA protocols, ARM introduced AMBA and is a trade mark of ARM ltd. [9] it was introduced in 1996 the first one to roll out of AMBA family was the ASB (Advanced system bus) and the APB (Advanced peripheral bus) in its AMBA 2 in 1999 later ARM introduced AHB(Advanced high performance bus), Then in 2003 third generation AMBA ie: AMBA 3 was introduced including AXI (Advanced extensible interface) and the ATB which is the (Advanced trace bus) , then AMBA 4 was introduced in 2010 which included AXI4, then in 2011 ACE(Advanced coherency extensions) was introduced ,later in 2015 AMBA 5 was introduced which included CHI(coherent hub interface) [9].

B. Details of AMBA Version

1) *AMBA:* The ASB and APB are the first bus of the amba family ASB (advanced system bus) is an high performing bus for 16 and 32 bit systems. Many same signals on an AHB are used for the ASB also. APB (advanced peripheral bus) is a bus used for low performing systems on a microcontroller, the low performing bus are the peripheral devices in a microcontroller
2) *AMBA 2:* Amba 2 was introduced in earlier days it is quite older generation of the chip interconnect which included AMBA AHB which is most commonly used across Arm7, Arm9 and Arm Cortex-M designs [3], it also includes ASB which is a high performance bus used for 16 and 32 bit systems, it also incudes APB2 a low performance bus used for peripherals
3) *AMBA 3:* AMBA 3 is also a older generation system on chip interconnect which included the AMBA AXI (Advanced extensible Interface) protocol and also introduced ATB (Advanced trace bus) and gave extended improvements on APB and AHB protocols [9]. AXI provides high-performance high frequency designs, AXI suits for high-bandwidth, low-latency system designs it enables high-frequency operation without using any of the complex bridges, it also interfaces the requirements of various range of interconnects and system [4]. It supports five unidirectional channels and enables simultaneous read and write transactions.[9], ATB transfers the trace information between components of a trace system they are in parallel with the peripherals which provides visibility for debug purposes. AMBA has advanced the AHB and APB in in AMBA 3 which introduced the lite version of the AMBA AHB ie: AHB-lite which operates only one master and multiple slaves, and the APB which includes wait and error report and supports low bandwidth transactions for low performing systems or peripherals in a microcontroller interconnect.
4) *AMBA 4:* AMBA 4 gives more advancements to AMBA 3 these include ACE (Advanced coherency extensions), which is the AXI with additional signalling which is system wide coherency. And some other protocols with is intended to fulfil the requirements of high bandwidth, high clock frequency components, which makes them a high speed interconnect. [9]. All these features makes them. Suitable for mobile and consumer electronics. ACE and ACE-lite comes under AMBA 4 ACE is a protocol which has three additional channels for sharing the data between ACE master cache and cache maintenance control of hardware [4]. ACE-lite is the lite version of the ACE in which small set of signals that offer input and output. Or in one way coherency in which ACE masters the cache coherency of ACE-lite masters. They can still snoop ACE master caches but other masters cannot snoop [9]. AMBA 4 also includes AXI 4 and AXI 4-lite AXI 4 is the advancement to AXI 3 which supports 256 burst lengths and supports multiple region interfaces. AXI 4-lite is the smaller version of AXI4 which is smaller and lite with burst length of one and some exclusive access are not supported. AMBA AXI stream is also a part of AMBA 4 interconnects

which transfers data unidirectionally from master to slave and it also supports single and multiple data streams in the same shared wires AMBA AXI stream is ideal for implementation in FPGA. And there are some versions of AMBA ATB which introduces trace synchronization signalling and synchronization request signalling, AMBA 4 APB version which includes transaction protection and sparse data transfer to the interface and the AMBA LPI this is a protocol of Q-channel and P-channel which manages clock and power of SOC components.

5) *AMBA 5:* AMBA 5 is the latest version of AMBA protocol which includes the CHI (coherent hub interface) , AXI5, ACE5 and AHB5 which extends its features and offers more performance to align and interface CHI [9], AMBA CHI interfaces the coherent processors like the cortex A-65, cortex A-76 and some dynamic memory controllers like core link DMC-620 to high performance non-blocking interconnects such as the core link CMN-600, AMBA CHI is widely used in networking, mobile and data centres It also supports high frequency nonblocking data transfers, supports end to end data protection, AMBA 5 includes AMBA AHB in this version 5 it includes features like secure and nonsecure signalling in address phase and it also supports more complex systems, it gives improvement to AHB-lite as they have more properties to adopt them widely across different platforms and provides features like multiple slave select and single and multiple copy which enables scaling in multiple cores as AMBA 5 has more interconnects it also includes AMBA CXS which is a nonblocking streaming interface protocol which is used in point to point packetized communications,[9], additionally it also has the AMBA ATP (adaptive traffic profiles) which is a synthetic traffic framework used for modelling systems and the AMBA distributed translation interface DTIwhich aligns with the MMU architecture of ARM.

These protocols together work in a microcontroller below figure shows a generic AMBA based microcontroller in which both AHB and APB buses contribute interconnects to a microcontroller.

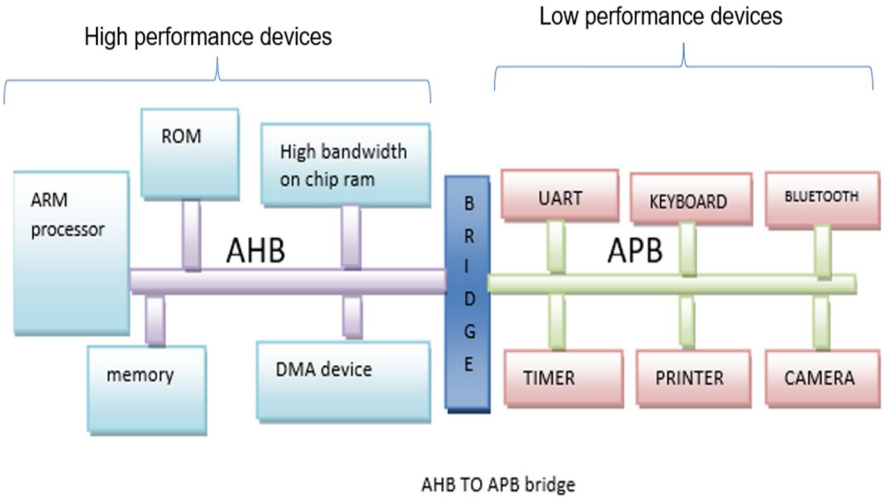


Figure 2 : Generic AMBA based microcontroller

From the figure we can see that all the high performing systems in a microcontroller such as ram, rom, processor dma device are interfaced by AHB the high performing bus, and the low perofrming systems ie: the peripherals in a microcontroller which are interfaced by APB the low performing bus, there is a bridge which interfaces these both systems here bridge is considered as a master and the systems are slaves when designing. The above shown high and low performing devices all together contribute a microcontroller interconnect

C. Overview of AMBA AHB-lite

AMBA AHB - lite is used in high performance synthesizable designs which supports a single master and multiple slaves operation [4] AHB-lite has many featurs for its high performing, and high clocking frequency systems which include

1) Single clock edge operations
2) Burst transfers
3) Non tristate operation
4) Wide data bus configuration such as 64, 128, 256, 512 and 1024 bits

Master slave configuration of AMBA AHB – lite protocol

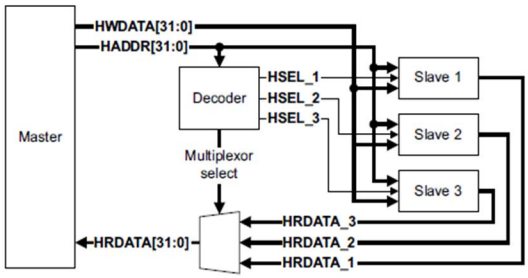


Figure 3 : AHB-lite block diagram

Figure 3 shows the block diagram of AHB-lite which has a single master and multiple slaves configuration system. The interconnect logic of the bus consists of address decoder, slave-to-master multiplexer. The decoder is used to monitor the address from the master so that required slave is selected and the multiplexer routes the corresponding slave output data back to the master this is how the operation of AHB-lite takes place

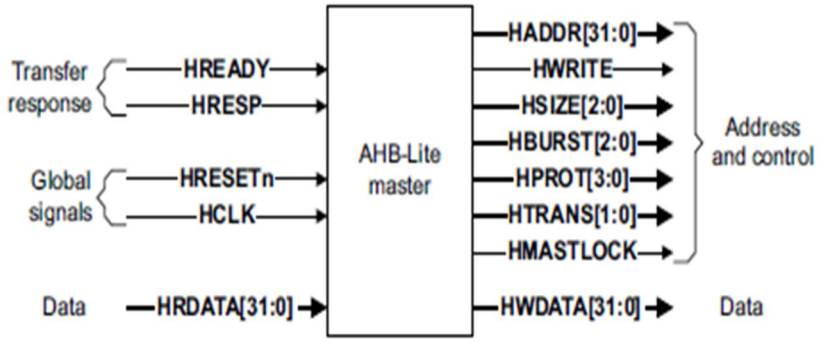


Figure 4: AHB-lite master

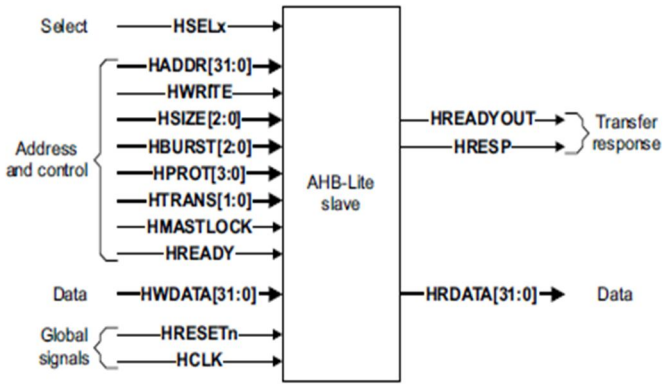


Figure 5: AHB-lite slave

Figure 4 indicates AHB-lite master which is responsible for read and write operations in the protocol, it consists of transfer response signals which maintains transfers across the master and slave , the global signals are the clock and reset as they are carried out in the entire process hence they are referred as global signals, address and controls are the one which address the transfers and maintains controllability over the transfers and response, and there are data signals which rolls data in and out of the specified transfers
Figure 5 is a AHB-lite slave which consists same signals as the master but the output of the master is the input to slave, and there exists a select signal which selects the required slave for the operation

D. Operation

Master sends transfers by driving the control signals and address these signals carry information about the address direction transfer width they even indicate the transfers which perform burst operation [3]

There can be single transfers also incrementing bursts do not wrap at the address boundaries and the Wrapping bursts which wrap at a particular address boundaries, Data is moved by the write data bus from the master to slave similarly read data bus is moved from slave to master every transfer consists of data phase and address phase, considering slave cannot request address phase all the slaves must be able to sample the address at this time a single slave can request a master which extends the data phase by the signal H-READY. when the signal is at low it causes wait states to be inserted to the transfer and it also enables the slaves to have some extra time which is to provide or sample the data, the slave uses H -RESP signal which is used to indicate the failure or success of a transfer [3]

E. Verification Background

Due to increase in the complexity of the designs and circuits nowadays verification is a must to avoid failure of chips [5]. But how good is the verification is it efficient or not, there are many methods in verification domain but many industries adopt UVM as their verification method because of its efficiency and reusable nature [6].

RTL designs and circuits were verified using the same design language ie Verilog language by using test bench, but later on when the circuit complexity increased these conventional methods were difficult to use as they had many problems like delay mismatch and slow process, Then later on they proposed a language which can be used for both design and advanced verification called as System Verilog

System Verilog was introduced for both design and verification, they are more advanced than Verilog language and more flexible than the hardware description language [7], system Verilog is standardized as IEEE 1800, system Verilog is based on Verilog with some advancements and extensions it is a evolution to Verilog

System Verilog's verification environment is flexible and the architecture can be developed on the ideas of the verification engineer on his basis, all the blocks in the architecture are coded separately and interfaced which is more faster and delay mismatch can be avoided.

System Verilog in addition supports many features for Verification including concepts like oops, coverage, assertions etc, which makes it a single language which is suitable for both design and verification. System Verilog is mostly used for design verification and also the coverage analysis, which we cannot achieve with such efficiency using Verilog HDL [7]. Object oriented feature of system Verilog makes it a powerful language to verify the corner cases of the logic designs which is not possible using HDL. One more most important feature of the system Verilog is that it has RANDOMIZE feature using which we can randomize the test vectors (stimuli) instead of generating all the values, this saves time and has a great efficiency compared to Verilog hence industries prefer system Verilog for verification.

F. System Verilog Verification Architecture

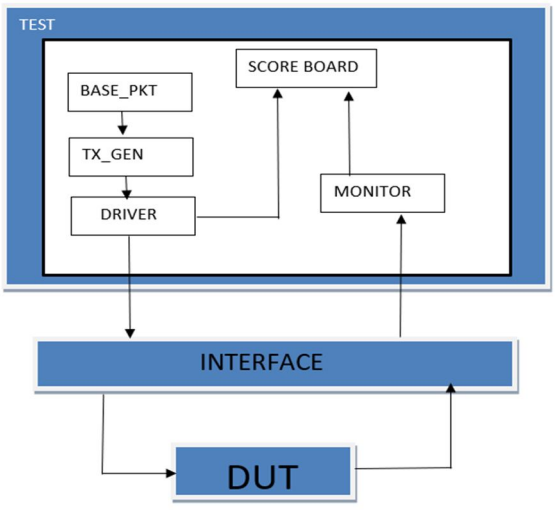


Figure 6: system Verilog verification architecture

Verification architecture is developed to check the correctness of a design which is under test, by generating stimulus or testbenches and driving a user defined correct input sequence to the design, which captures the design output and compares with the expected output. Verification environment is a group of blocks which performs specific operation generating stimulus and driving them to the respective blocks and also monitoring them. and the classes will be named based on the operation in figure 6 base packet is used to generate test vectors to generate these test vectors we need variables. all the test bench variables will be generated here in this block

G. Functionality of Blocks in the Verification Architecture

1) *Trans Generator:* all the variables generated in base packet will be given to TX_GEN here the variables will be converted to the test vectors
2) *Driver:* driver drives the test vectors to the respective blocks one to the interface and to the score board
3) *Monitor:* monitor captures the DUT output and sends it to the SCOREBOARD for verification monitor is for observability purpose
4) *Interface:* interface is a communication medium between the test and the designed circuit which is under test (DUT) interface establishes connectivity, direction, timing, functionality
5) *Scoreboard:* score board compares the generated test vectors and the DUT to see weather the designed circuit is correct
6) *Mailbox:* It is a medium which is responsible for transferring of data between blocks the data generated from one block is not directly transferred to another block rather it is transferred to the mail box from the mail box another blocks takes the data and processes This operation is done by PUT and GET option
7) *DUT:* and the DUT output (reference and DUT) and gives the verification result matched or not matched by this system verilog verification is achieved but system Verilog also has some drawbacks.

H. Issues of System Verilog

System Verilog language is not portable, one developer who works on its verification architecture and a different person wants to continue should have to learn the architecture framed by that developer ,System Verilog does not have features like reusability Although system Verilog has some issues industries started developing their own architecture, but one employee who quits and works for another company has to study their architecture first and get familiarized in which he cannot work instantly. Therefore to overcome these problems many industries came forward to develop a common universal methodology and the result in which UVM (universal verification methodology) was introduced.

I. Introduction to UVM

As we know verification contributes about 60 to 70% in VLSI flow all those discussed above methods did not meet the requirements hence to obtain a fast, reusable, efficient verification UVM was introduced [5]. UVM is not a separate language rather it is a methodology the language is system Verilog with some additional features and advancements. UVM is a standard verification methodology to verify integrated circuits [10]. UVM was developed from OVM later upgraded to VMM then further upgraded to VMM 1.2 and then UVM was introduced, UVM was made a standard by accellera, aldec, cadence, mentor graphics and synopsis provide support to UVM

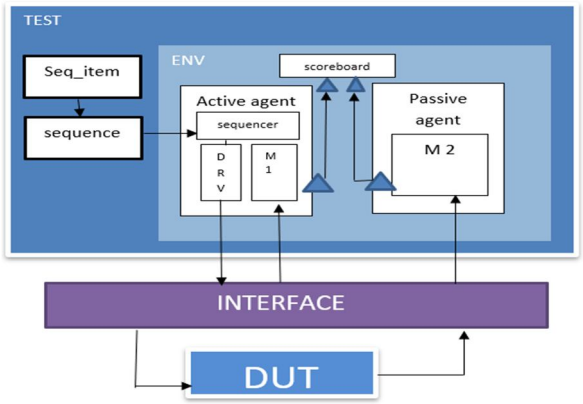


Figure : 7 UVM verification environment

Figure 7 shows verification architecture of UVM from the diagram we can see that there are several blocks in the test environment such as sequence item, sequencer, active, passive agent and a scoreboard and a common interface to link the test design model or the test vectors to the DUT and perform verification same as the System verilog environment but this environment is a standard architecture which should be followed by every verification engineers and cannot skip therfore to avoid portable issues . including UVM is more flexible and have more features than system verilog [6]. From the architecture we can see that there are two diferent blocks active and passive agent in which active agent maintains input transactions and passive agent maintains output transactions. hence additionally UVM has two monitors M1 and M2, here sequencer acts as a controller in which it receives response from driver wether to send data to the driver or not by this we can achieve more controlabilty and efficiency

J. Some Important Components of UVM Verification Architecture Are

1) *Library:* UVM consists of class libraries which are needed for well constructed reusable environment which improves interperability and rewriting the IP's using libraries we can use components to use instantlly.
2) *Factory:* Commmonly factory is used in oops , object used to instantiating other objects, we can also register objects in this factory, it also provides variety of create methods and to register to us it with a particular instance
3) *Sequence:* Sequencer initilazes the verification environment which configures the Verification environment and the DUT, and the memory is loadd with any types of initial conditions it also registers the settings
4) *Scoreboard:* Score board is the main component for final results of a verification environment, scoreboard can be implemented in different way. Scoreboard also takes the inputs and outputs from DUT to determine the result with the reference model and compares.
5) *Agent:* DUT may have diferrrent interfaces in which each has different objects associated with them the monitor and scoreboard or even a PCI interface will be different from the Ethernet interface. Various objects of UVM can be organized as members of a wrapper class which is known as an agent, there are two types of agents active and passive, active agent maintains input transactions and passive agents maintains output transactions
6) *Driver:* Driver is the unit which drives the data from one block ro the respective block , these may be bits of data or information the drivers duty here is to take the sequence item and provide proper stimulus to the DUT

By all these above mentioned features UVM is more adoped across verification industries in which we can obtain more efficient and resuable verification to increase productivity.

K. Advantages of UVM over System Verilog

UVM is a fixed architecture in which we cannot change the standard procedure hence the verification environment is protable
UVM is a reusable environment
We can separate the tests from the testbenches hence we can reuse the stimulus
Using factory feature we can simpilify and modify components easily
UVM helps to configure the mechanisms of different testbench components based on the use of which verification environment is used
UVM is also simulator independent ie : it is is supported by all the simulators and not dependent on any particular simulator.

III. COMPARASION

A. AMBA comparison

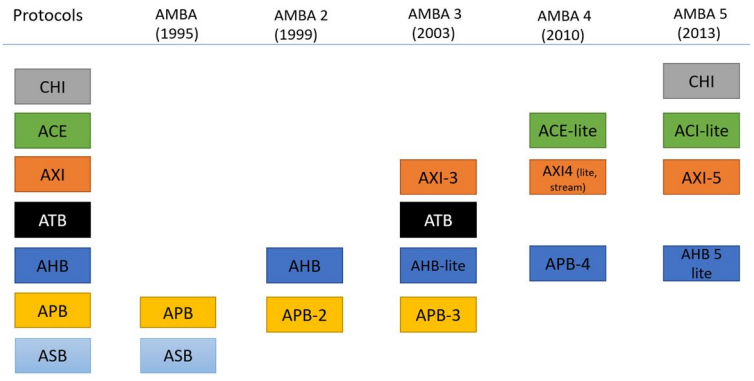


Figure 8: Comparison of AMBA

AMBA has introduced its first bus in 1995 and its AMBA 5 the latest version in 2013, AMBA protocols has gained a wide popularity and standards in industries they are used in many interconnects and SOC designs Figure 8 shows various types of AMBA protocols based on their AMBA family and their year of introduction. All these protocols are classified based on their operating nature, performance, clocking, transfers etc, In addition there are more new protocols like CXS which is a non blocking streaming interface protocol, The AMBA Distributed Translation Interface the DTI which aligns with the architecture of MNU of the ARM which is suitable for scalable distributed messaging protocol, The AMBA LTI Local Translation Interface ie: LTI protocol specification which aligns with the Arm MMU system architecture which provides higher performance and achieves efficient translation services. And the LPI protocol which has Q channel and P channel to manage clock and power [9]. By all these factors AMBA protocols have been classified.

B. Verification Comparison

Verilog vs System Verilog

VERILOG	SYSTEM VERILOG
Most used for design	Used for design and advanced verification
Verilog do not Support user defined types	Supports user defined types
There is no Randomization feature	Variables or test vectors can be randomized
In Verilog Verification is limited and it is not suitable for advanced verification	Supports advanced verification and allows many features like polymorphism, encapsulation, inheritance etc….
Verilog is not compact and flexible for verification	System Verilog is suitable for advanced verification, it is compact and flexible
Verification environment of Verilog causes delay mismatch	There is no delay mismatch is System Verilog verification
Verilog language is not suitable to verify complex logics and not that efficient	Object oriented nature of system Verilog makes it to verify even the corner cases of the logic

On the other hand let us compare system Verilog verification and UVM

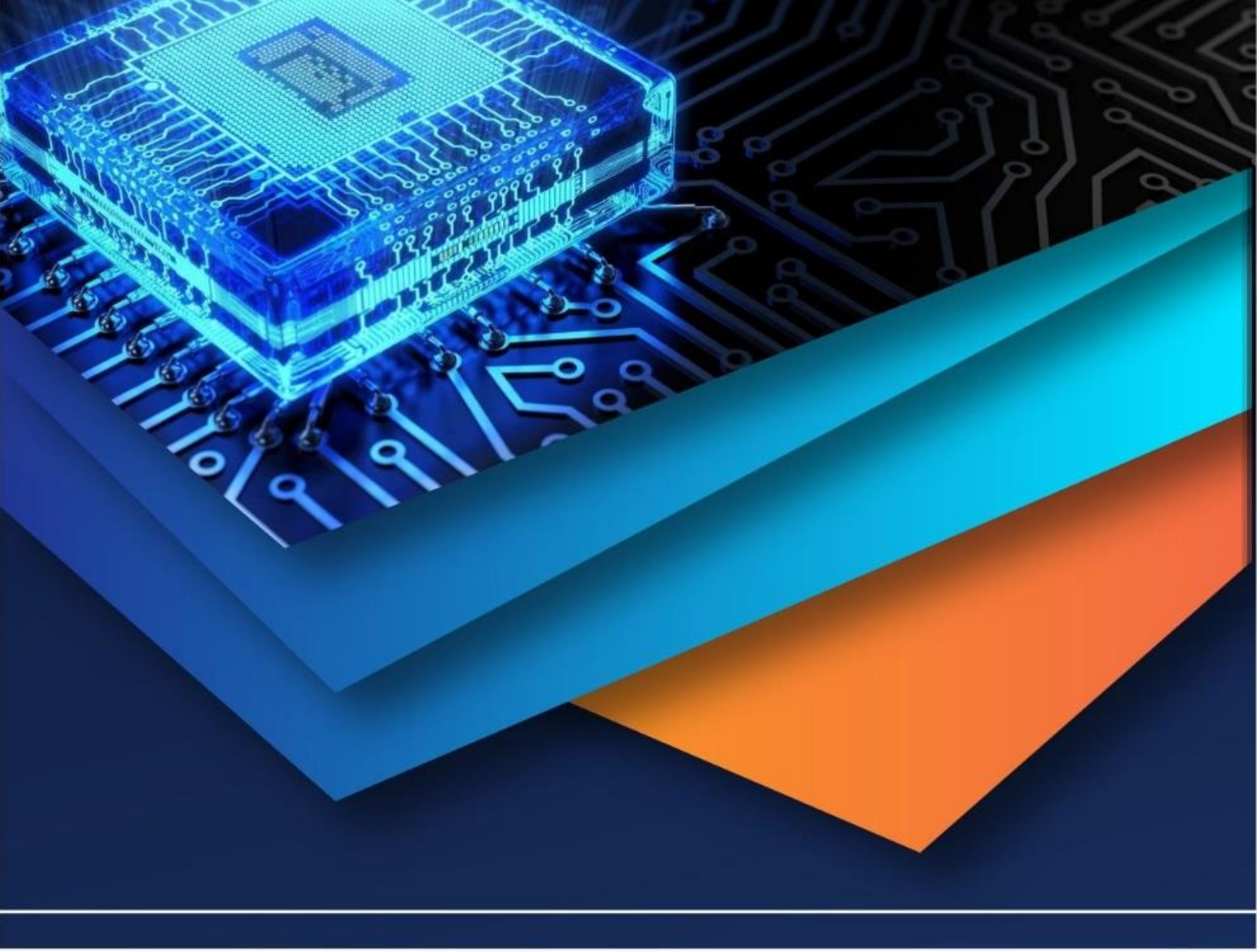
SYSTEM VERILOG	UVM
Verification language	Verification methodology
Not efficient as UVM	Efficient verification can be achieved using UVM
Verification architecture is not portable	verification architecture is portable
Stimuli generation cannot be controlled	Stimuli generation can be controlled and is more flexible
System Verilog verification is not reusable	UVM offers reusability
Verification methodology is not standardized it can be changed according to the developer's convenience	UVM is a fixed standard methodology that every verification engineer should follow
Testbenches in Verilog are not structured	Testbenches are structured in UVM

IV. CONCLUSION

This paper shows the different versions of AMBA protocols their nature of operation and their contribution to microcontrollers, and we have discussed about AMBA AHB-lite in detail and their working. and as we know complexity of VLSI chips and their logic is increasing very rapidly we need a fast reliable and efficient verification of designs so we have discussed need of verification its evolution and about system Verilog and UVM and compared them based on their various features.

REFERENCES

[1] P. Giridhar and P. Choudhury, "Design and Verification of AMBA AHB," 2019 1st International Conference on Advanced Technologies in Intelligent Control, Environment, Computing & Communication Engineering (ICATIECE), Bangalore, India, 2019, pp. 310-315, doi: 10.1109/ICATIECE45860.2019.9063856.
[2] Milica Mitić and Mile Stojčev "A Survey of Three System-on-Chip Buses AMBA CoreConnect and Wishbone" university of Niš
[3] Sravya Kante, Hari KishoreKakarla and Avinash Yadlapati "Design and Verification of AMBA AHBLite protocol using Verilog HDL" , Vol 8 No 2 Apr-May 2016, e-ISSN: 0975-4024 International Journal of Engineering and Technology (IJET)
[4] Anurag Shrivastava, G.S. Tomar and Kamal K. Kalra "Performance Comparison of AMBA Bus-Based 2011 IEEE", DOI 10.1109/CSNT.2011.98, 2011 International Conference on Communication Systems and Network Technologies.
[5] W. Ni and J. Zhang, "Research of reusability based on UVM verification," 2015 IEEE 11th International Conference on ASIC (ASICON), Chengdu, 2015, pp. 1-4, doi: 10.1109/ASICON.2015.7517189.
[6] Khaled Salah "A UVM-based smart functional verification platform: Concepts, pros, cons, and opportunities" research gate DOI: 10.1109/IDT.2014.7038594, February 2015
[7] Jonathan Bromley Verilab Ltd , Edinburgh Scotland If SystemVerilog Is So Good, Why Do We Need the UVM.
[8] M. S. Ohileshwari and A. B. Gudi, "Detection of stability faults in sub-threshold SRAM cell using IDDT waveform," 2018 2nd International Conference on Inventive Systems and Control (ICISC), Coimbatore, 2018, pp. 774-780, doi: 10.1109/ICISC.2018.8398904.
[9] www.arm.com
[10] www.wikipedia.com



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY