iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 9 **Issue:** VII **Month of publication:** July 2021

DOI: https://doi.org/10.22214/ijraset.2021.36282

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 VII July 2021- Available at www.ijraset.com

Verification of Wishbone Bus Interface for SoC using System Verilog and UVM

Pooja. D. R[1], Nanda. B. S[2]
[1, 2] *ECE Department (VLSI Design and Embedded systems), PES College of Engineering Mandya, Karnataka (India)*

Abstract: ***The Verification phase carries important role in design cycle of a system on chip. Verification gives with the actual enactment and functionality of a DUT and to verify the design meets the system requirements. This paper present wishbone bus interface for soc integration to interconnect architecture for portable IP cores and test bench is developed in system Verilog and verification is done by both system Verilog verification methodology and universal verification methodology which includes scoreboard, functional coverage and assertion. This paper based on two application to integrate IP cores that is single master with single slave interconnection and single master with multiple slave interconnections where master is test bench and slave will be a core.***

Keywords: System Verilog, environment, coverage, assertion, UVM.

I. INTRODUCTION

Integration of multimillion transistors in single chip is the demand of semiconductor industries. With rapidly increasing number of transistor integration in single chip greatly increase the performance of the system with an increase in the complexity also. The system on chip design is proposed to this problem. The SoC integration can design large system at acceptable cost and quality by the use of IP blocks. System-on-Chip (SOC) design is proposed as an extended methodology to this problem where intellectual property (IP) cores of embedded processors, interface blocks, analog blocks and memory blocks are combined on a single chip for specific applications. In this project wishbone interconnect is use as common interface, providing a standard data exchange between IP core to create a custom system on chip. Wishbone allow cores to be integrated more easily and quickly alleviating soc integration problem moreover its IP core can be easily available and most of its cores are free to users. The wishbone design was strongly influenced by three factors. First, there was a requirement for a good, reliable system-on-chip integration for a good, reliable system-on-chip integration solution. Second, there was a need for a common interface specification to facilitate structures design methodologies on large project teams. Third, they were impressed by the traditional system integration solutions afforded by microcomputer buses such as PCI bus and VME bus. Previously, IP cores used non-standard interconnection schemes that made them difficult to integrate. That can't be used in different application. So, this required the creation of custom logic to connect each of the cores together. With the use of standard interconnection scheme, the cores may be integrated more easily and quickly by the end user.

II. WISHBONE BASICS

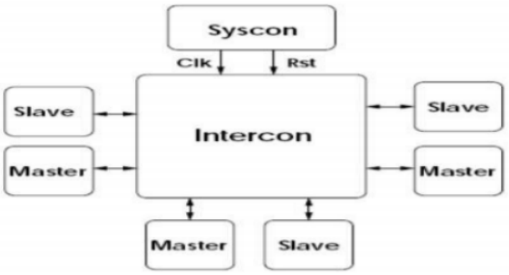


Fig.1. Block diagram wishbone basics

Wishbone utilizes master and slave architectures which are connected to each other through an interface called intercom. Master is an IP core that initiates the data transaction to the slave IP core. Master starts transaction providing an address and control signal to slave. Slave in turn responds to the data transaction with master wit specified address range. The intercom is the medium consists of wires and logics which help in data transfer between master and slave. The intercom also requires a SYSON module which generates wishbone reset and clock signal for the proper functioning of the system.

III. PROPOSED SYSTEM ARCHITECTURE

A. Point-To-Point Interconnection

Point-to-point interconnection includes DMA, MEMORY and SYSCON Cores.

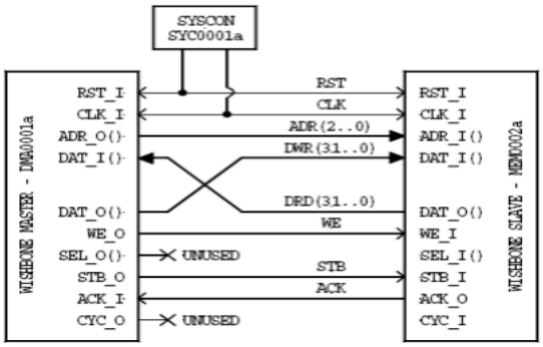


Fig.2.Point-to-point interconnection

Point-to-point is the simplest way of connecting two IP cores to each other where a single master interface is connected to multiple slave interfaces. For example, a microprocessor as master can be connected to serial input output port slave interface.

The DMA core is a simple 32-bit DMA unit with a WISHBONE Master interface. Two methods of data transfers are supported such as, single read/write cycles and block read/write cycles. The type of cycle is selected by a non-WISHBONE signal input DMODE. If DMODE input is negated, the DMA generates single read/write cycles, if it is asserted the DMA generates block read/write cycles. In block read write mode, the DMA initiates eight phases of block write cycle, and then DMA generates a similar kind of block read cycle. The Memory is a simple, 8x32-bit size memory module with WISHBONE Slave interface. When data input is given to memory block. First it will check for WE and STB signals if both the signals are asserted then only it will write the data on memory otherwise memory's data remain unchanged. DMA access the memory block and if strobe is asserted then it indicates slave is selected. A slave should give response other wishbone Signals only when strobe signal asserted. When reset is asserted then slave should not take input data. DMODE, WE and STB signals are asserted then only input data can be writing on memory. It supports single read/write, block read/write and RMW cycles.

SYSCON: The SYSCON also called as system controller is used to generate WISHBONE compatible clock and reset signals for the system. The clock output is fed directly from an external clock signal called EXTCLK. The reset generator produces a single reset signal RST in accordance with the WISHBONE reset timing.

IV. VERIFICATION METHODOLOGIES

A. System Verilog Verification methodology

System Verilog is a kind of standard hardware description language, it is well known as hardware description verification language. By making use of System Verilog, it is possible to design, simulate, test, and finally we can implement electronic circuits.

The System Verilog verification hierarchy consists of components explained below:

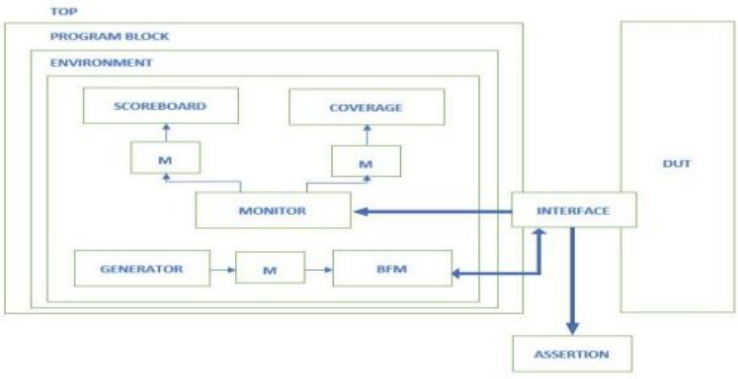


Fig.3. System Verilog verification environment of wishbone bus interface for SOC

1) *Top:* Top module contains environment and test bench.
2) *Program Block:* Program blocks are mainly used to avoid the race condition problems.
3) *Environment:* It mainly consisting of generator, BFM, monitor and coverage, all are performing automatic in nature.
4) *Transaction:* An interface in system Verilog is a collection of wires, and optional tasks and functions to manage those wires.
5) *Generator:* The generator consists of the basic stimulus data and generates the actual packets that are later sequenced and packed by agent. The stimulus generator is governed by strict constraints.
6) *Bus Functional Model:* BFMs apply stimuli to the design under verification via complex waveforms and protocols.
7) *Monitor:* Monitors DUT interface and forward transaction to higher layer.
8) *Score Board:* It verifies the function of design against a reference model.
9) *Functional Coverage:* Measures how much % of scenario is covered.
10) *Assertion:* Assertions are used to validate the behaviour of a system defined as properties

B. *Universal Verification Methodology*

UVM is the widely used verification methodology because it provides reusability property. Due to many reasons like unpredictable designs of products, time to market and workload on verification process requires reusable test benches. UVM have well-structured class libraries which are used while designing the well-constructed, reusable test bench.

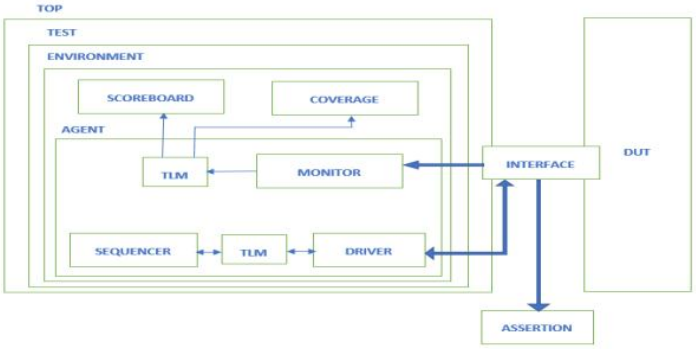


Fig.4. UVM verification environment of wishbone bus interface for SOC

1) *Top/Test Bench:* The UVM Test bench typically instantiates the Design under Test (DUT) module and the UVM Test class, and configures the connections between them.
2) *Test:* The UVM Test typically performs three main functions: Instantiates the top-level environment, configures the environment, and applies stimulus by invoking UVM Sequences through the environment to the DUT.
3) *Environment:* The UVM Environment is a hierarchical component that groups together other verification components that are interrelated. Typical components that are usually instantiated inside the UVM Environment are UVM Agents, UVM Scoreboards, or even other UVM Environments.
4) *Scoreboard:* The scoreboard receives the transaction from the driver and performs the same task on the model. For the given memory DUT an associative array is used as model. The scoreboard also receives the output from the DUT and compares them with the expected output.
5) *Functional Coverage:* Measures how much % of Stimuli Scenarios are covered.
6) *Agent:* The UVM Agent is a hierarchical component that groups together other verification components that are dealing with a specific DUT interface.
7) *TLM:* TLM, transaction-level modelling, is a modelling style for building highly abstract models of components and systems.
8) *Sequencer:* The UVM Sequencer serves as an arbiter for controlling transaction flow from multiple stimulus sequences.
9) *Driver:* Driver class converts data inside a series of sequence –items into pin level transaction. It takes input from sequencer and passes it to DUT through an interface.
10) *Monitor:* The UVM Monitor is responsible for captures the signal activity from the design interfaces and translates it into transactions level data objects that can be sent to other components.
11) *Sequence:* A UVM sequence is an object contains behaviour for generating stimulus.
12) *Assertion:* Validates the protocol behaviour at DUT interface

V. RESULTS AND DISCUSSIONS

This section gives the result analysis of Wishbone bus interface for soc in system Verilog and UVM. The simulation is carried out using Mentor Graphics Questasim tool for a data bit size of 32- bites. Various test scenarios were created in order to verify the functionalities of the wishbone bus interface thoroughly. Here the single read/write, Block read/ write and read modify write were utilized in order to carry out verification analysis.

A. Single Master With Multiple Slaves

SYSTEM VERILOG: - SINGLE READ/WRITE

Single master and multiple slaves hear slave increased into 4 slaves every single salve works as single slave on that time every single slave will create one memory and Master will always be test bench. For 4 Slaves four particular memory will create so to specify a particular memory all address will be same. But write data will be change, while in read operation also for particular address different different address will get different slaves as different memory.VIF1 is 1st slave all signal will not work parallel all slaves works as concurrent. First operation have 5 write operation, where WE is high is write operation when it is low it is read operation and STB CYC and ACK this all the signals are high and it will be low once slave is terminated and 2nd slave vif2 like this it will continued until the 4th slave

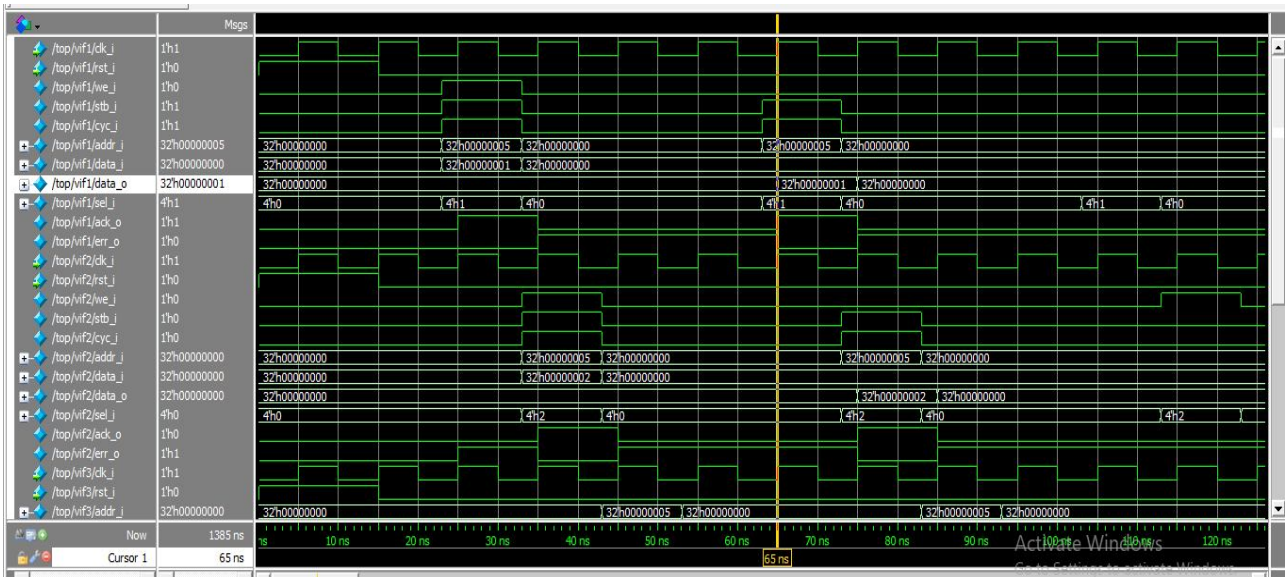


Fig.5. Waveform of Single read/write

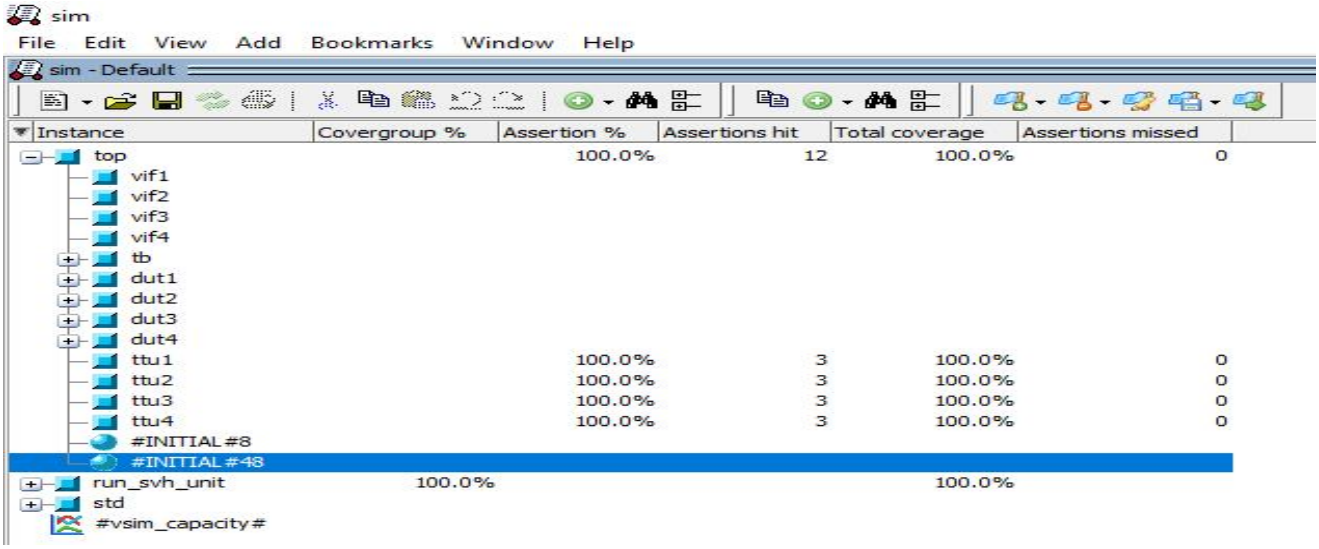


Fig.6. Assertion Output

B. Single Master With Single Slave

UVM: - BLOCK READ/WRITE

Block read/write block transfer cycle perform multiple data transfers. However, the BLOCK cycles are modified so that these individual cycles are combined together to form a single block cycle.

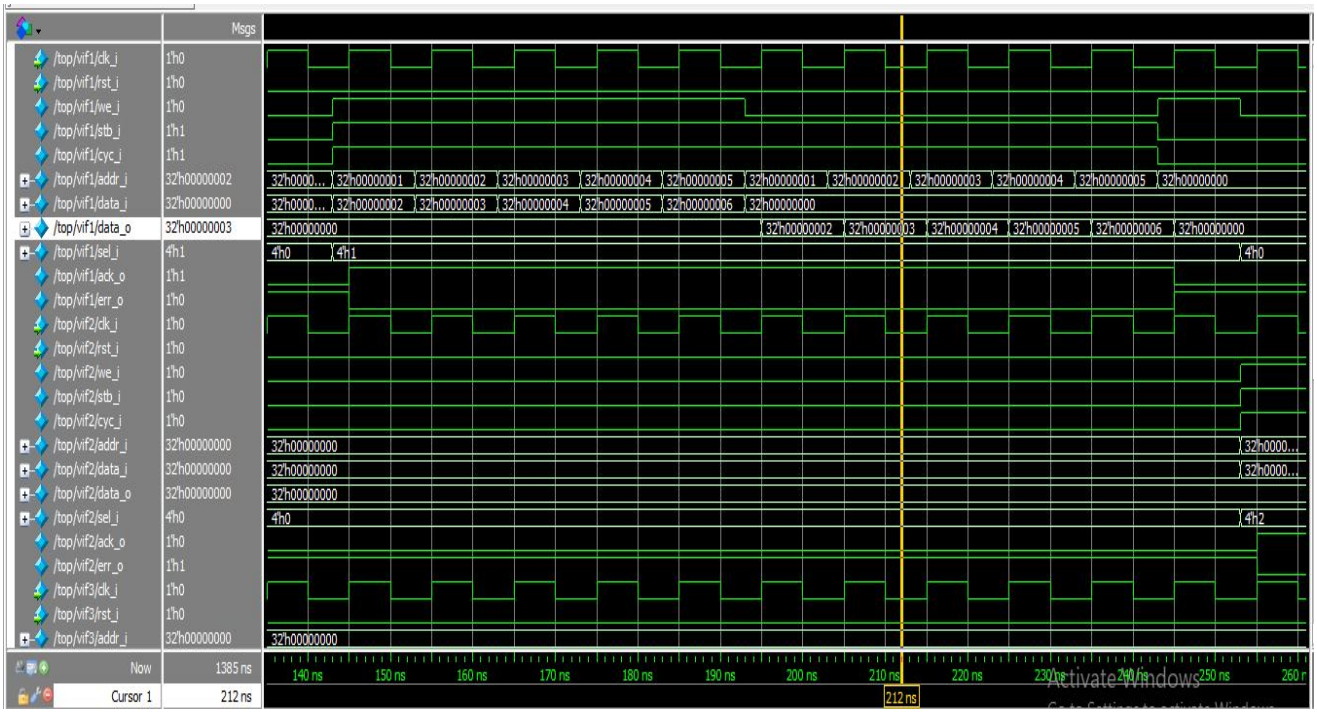


Fig.7. Waveform of Block read/write

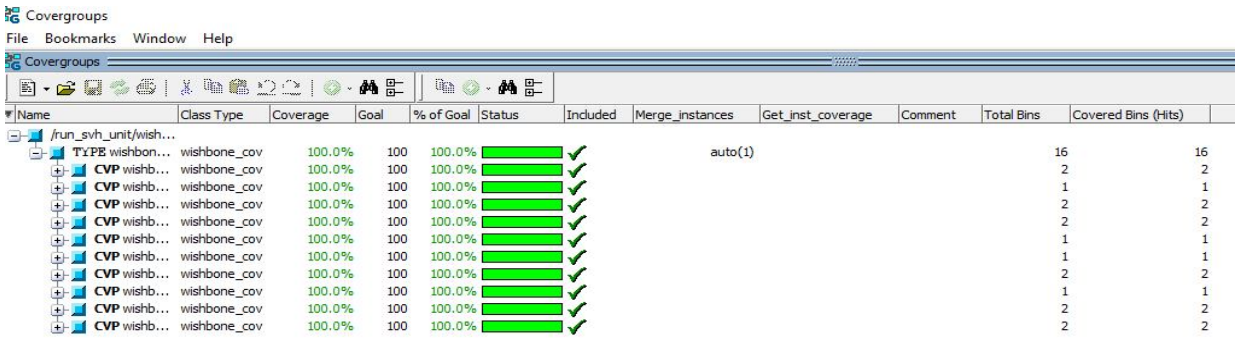


Fig.8. Coverage Output

Assertion and Functional coverage are used to determine the progress of the verification. The functional coverage in used-defined. The comprehensive of the functional coverage depends on the test plan. Assertion validates the protocol behaviour at DUT interface. The assertion and functional coverage reported by the tool for both SV and UVM.

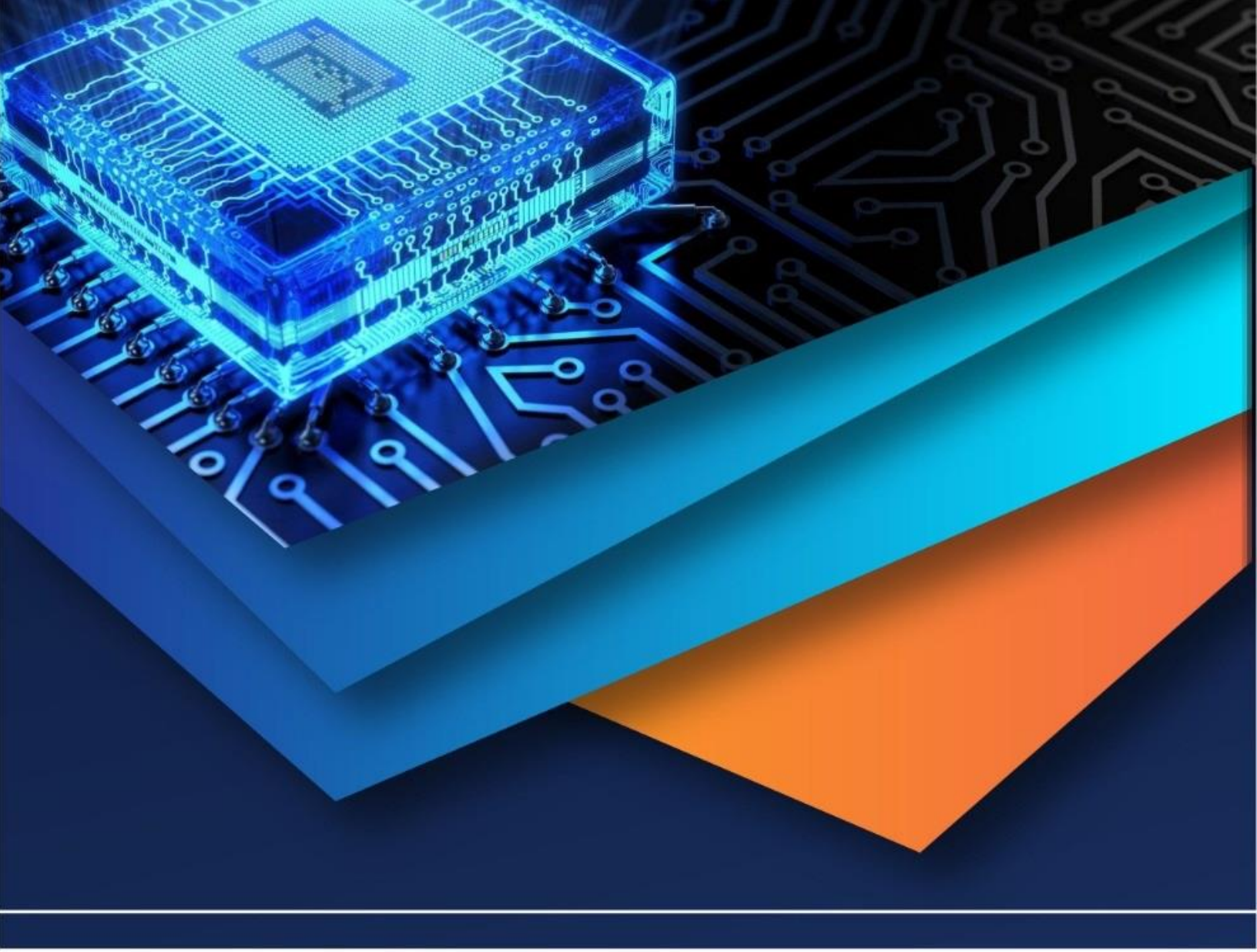
Likewise in system Verilog Block Read/Write and Read modify Write and in UVM Single read/write, Read modify Write all this assertion and functional coverage environment is covered with 100%.

VI.CONCLUSION

The Proposed verification of wishbone bus interface is discussed briefly with single master with single slave and single master with multiple slaves along with specified methods of custom verification in system Verilog and universal verification methodology. The result of verification methodologies indicates that the test bench accomplished complete verification of wishbone and achieved 100% functional coverage and Assertion .The developed test cases is verified by using Mentor Graphics Questasim verification tool and simulated waveforms are achieved as expected.

REFERENCES

[1] Changlei Dongye "Design of On-Chip Bus Based On Wishbone" Shandong University of Science and Technology, IEEE 2011
[2] Vaundhara Patel K.S and Bhavana.R. "Design and Verification of Wishbone I2C Master Device" IEEE 2018.
[3] Chandrla Brijesh A, Mahesh T.Kolte "Design and Verification Point-to Point Architecture of Wishbone Bus for System-on-chip" International Journal of Emerging Engineering Research and Technology Volume 2, Issue 2, May 2014, PP 155-159
[4] Prashant Bachanna, Vivek Jalad and Sharanbasappa Shetkar "Design and analysis of high speed low power reusable on chip bus based on wishbone" IEEE 2013
[5] Gajski, D.D; Wu, A.C-H; Chaiyakul, V.; Mori, S.; Nukiyama, T. and Bricaud, P. "Essential issues for IP reuse," Design Automation Conference 2000 Proceedings of the ASP-DAC, Asia and South Pacific, pp. 111-114, 2000.
[6] Richard Herveille, WISHBONE System-onChip (SoC) Interconnection Architecture for Portable IP Cores, rev. version: B4, 2010. By Open Cores Organization, p.7, 2010.
[7] Mohandeep Sharma, Dilip Kumar "Design and Synthesis of Wishbone Bus Dataflow Interface Architecture for SoC Integration" Annual IEEE India Conference (INDICON) pp.813 - 818 December 2012.
[8] Ayas Kanta Swain, KamalaKanta Mahapatra "Design and Verification of WISHBONE Bus Interface for System-on-Chip Integration" Annual IEEE India Conference (INDICON) pp.1 - 4 December 2010.
[9] Resve Saleh and Steve Wilton, "System-on-chip: Reuse and integration "Proceddings of the IEEE, vol. 94, No6, pp. 1050-1069 june 2006.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY