

UVM based Design Verification of FIFO

Apoorva H M¹

Electronics and communication department,
BMS College of Engineering
Bengaluru, India

Dr. Kiran Bailey²

Assistant Professor, Department of ECE
BMS College of Engineering
Bengaluru, India

Abstract—Verification process is important stage in SOC's and FPGA. As the technology is leading towards nano new methodology's are coming up in field of verification. Universal Verification Methodology (UVM) is one of the methodology with advantages robust, scaling and reusable. In this work Synchronous FIFO is designed using Verilog and verified using UVM and simulation is carried out in Questa Sim tool.

Keywords—FIFO; UVM; Questa Sim;

I. INTRODUCTION

Verification process is one of the most important stage in chip designing, where the design has to be verified for different test cases to check its functionality. Verification consumes maximum time in product cycle. UVM is one of the methodology used to reduce the functional verification time [1]. FIFO is the integral part in most of SOC design and FPGA's [2][3][4]. FIFO extensively used as buffers, flow controllers, synchronizers and data storage. This paper deals with verifying the synchronous FIFO for different test scenarios using universal verification methodology.

II. RELATED WORK

Verifications have been done in various hardware description languages such as VHDL, Verilog and System Verilog. VHDL and Verilog lack in high level data types. System Verilog is advanced version of Verilog, which uses OOPS Concept. But, as the technology is scaling down, the occurrence of bugs and time consumption for verification is increasing and there is a need for test bench environment that is scalable, robust and reusable [5].

UVM was used for design verification but, it was replaced by UVM. As it offers advantages such as a sequential library that collects multiple sequences, command line processor and many more that make it a better methodology for design verification [6].

III. PROPOSED WORK

A. Synchronous FIFO Design

In this paper FIFO consists of dual port RAM. This dual port RAM allows simultaneous access of read and write port [7]. Fig. 1 shows synchronous FIFO architecture. The read and write process of FIFO is performed on a same clock. For every positive edge of the clock the data is written in to the FIFO, when the write enable is high and the FIFO is not full. When the read enable is high the data is read out for every positive edge of the clock and FIFO is not empty and all the output signals is set to zero when reset is given.

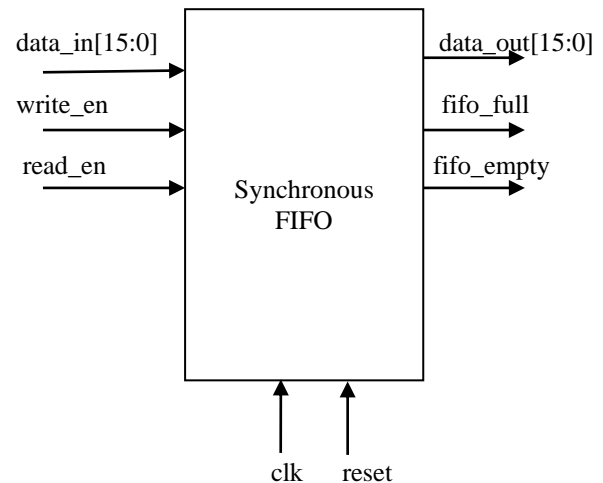


FIG.1: Block diagram of FIFO

B. Verification Plan

Verification plan is written before verifying any project to ease the work of verification engineer. Based on the requirement of the project, verification plan is to be built which includes list of test cases and coverage models. Fig.2 shows the verification flow which explains the types of phases went through the verification process.

C. Universal verification methodology

1) UVM Classes

UVM includes class libraries from which different components for verification can be derived. It has 3 classes.

- a) UVM_Object: It is a base class for UVM data and components which define the random seeding and methods of operation for copy, create, print, record, etc.
- b) UVM_Components: It is a root class for UVM component which defines factory interface and transaction recording. Each UVM component can be addressed through a hierarchical path name.
- c) UVM_Transaction: Transactions are driven by UVM_Transaction. It is transient in nature and inherits UVM_Object features. It gives timing and recording interface.

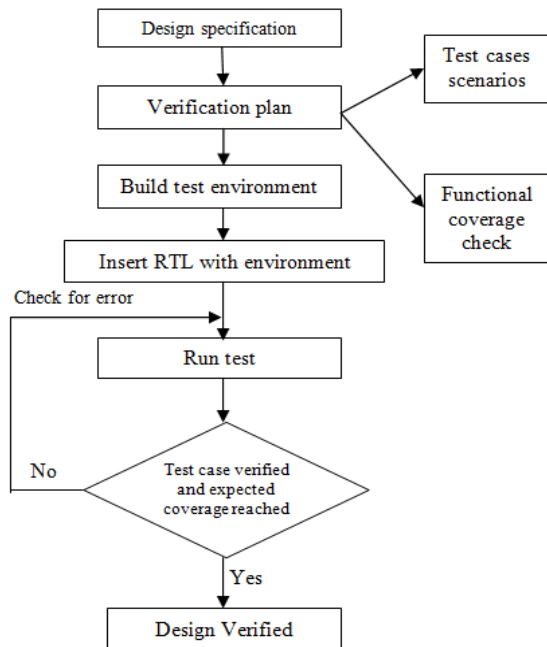


Fig.2:Verification flow

2) UVM Test Bench

UVM test bench architecture can be classified in into three parts

- Generating the stimuli.
- Stimulus as input to DUT.
- Verifying the functionality and measuring the overall coverage.

UVM Test Bench consists of a number of components which will be helpful to build reusable test bench [8]. The functionality of the components are as follows.

- Sequence Item:** It has data field which is used to generate stimulus and represents the communication at the level of abstract.
- Sequence:** It is a collection of sequence item where items are randomized and then sends to driver.
- Sequencer:** It controls the request and response of the sequence item between driver and sequence.
- Driver:** It is used to drive signals to DUT using virtual interface.
- Monitor:** It monitors the signal activity of design interface and converts pin information into the packets and transfers to the scoreboard and coverage collector.
- Agent:** It is encapsulation of sequencer, monitor and driver. It can have an active and passive agent. Passive agent is used only to monitor DUT activity. Whereas active agent drives the signal to DUT and instantiates the components in it.
- Scoreboard:** It will receive the transaction from agents and compares the DUT output with expected values to check the correctness of DUT.
- Coverage:** It measures the verification and finds the area untested in DUT.
- Environment:** It holds the multiple agent and components like scoreboard and functional coverage. A test bench can have numerous environments.

- Test:** It explains the test scenario and hold configuration properties, environment and class overrides, etc.
- Top:** It is the root of the hierarchy and the top contains the verification components, DUT instance, clock and reset generator log, interface instances etc.

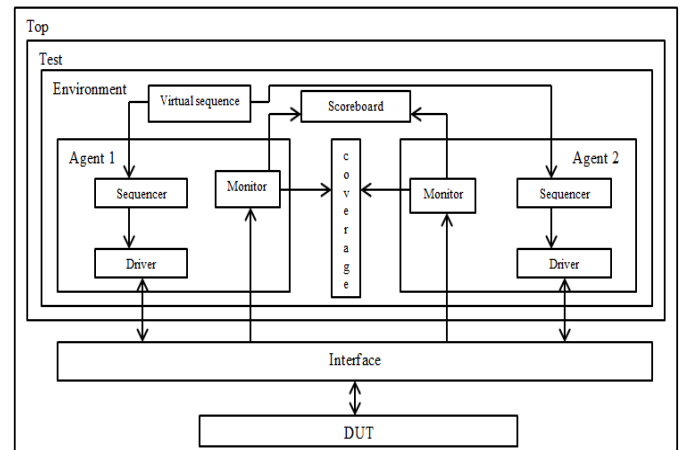


Fig.3:UVM Test Bench

3) UVM Phases

The simulation is carried out in a set of phases to achieve synchronizing mechanism [9]. It is divided into 3 categories

- Build time Phase:** It executes in zero simulation time and runs in top down style.
- Run time phase:** The run phase is a time taking phase, test cases will run in this phase. This phase is performed from start to end of simulation.
- Clean Up Phase:** In this phase results are collected from scoreboard and coverage after run phase.

D. Synchronous FIFO UVM Test bench

- Fig.3 shows verification components where Agent 1 and Agent 2 is data agent and reset agent. Data agent transmits the data from sequencer to DUT and reset agent is used to generate the intermediate reset. Scoreboard will give the results of test scenarios and coverage gives the coverage report.
- Virtual sequence is required to coordinate the stimulus in multiple driving agent.
- Interface consist all input and output signals of synchronous FIFO. It provides communication between driver and DUT.

IV. RESULTS AND SIMULATION

Designed synchronous FIFO is verified using UVM test bench, the simulation is carried out in Questa Sim tool. The scoreboard results are obtained by comparing actual data and expected data, if the data match's, successfully compared will be displayed in transcript or else mismatch will be displayed. Simulation is done for different test scenarios which are

- Reset during FIFO operation, Fig.4.
- Reset applied initially, Fig.5.
- FIFO full condition, Fig.5.
- FIFO empty condition, Fig.5.
- FIFO write and read condition, Fig.6.
- Simultaneous write and read condition, Fig.7.

For all the above scenarios functional coverage is obtained
As in TABLE I.

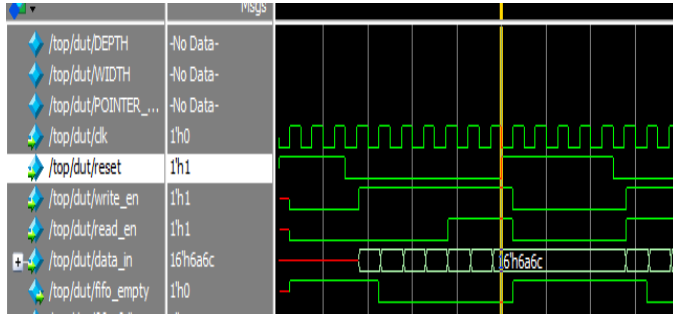


Fig.4:waveform of reset during FIFO operation

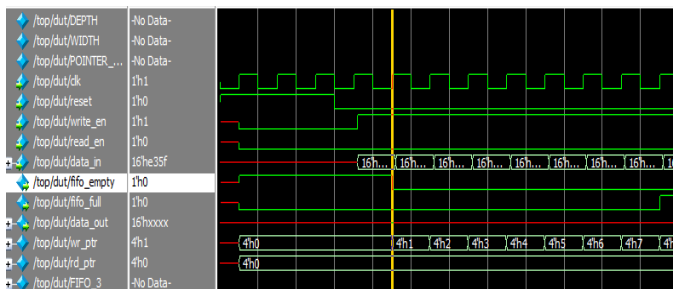


Fig.5:waveform of initial reset,FIFO full and FIFO empty condition

V. CONCLUSION

Synchronous FIFO is verified for possible scenarios using UVM test bench, which have advantage of time reduction with the help of base class, Provides reusable components, define the input stimuli by constraint randomization. The designed synchronous FIFO can be used in the application of SOC and FPGA has it is reliable and consumes less time for verification because of UVM.

TABLE I. RESULTS

Serial number	Test Scenarios	Coverage	Scoreboard
1	Reset during FIFO operation	100%	Successfully compared
2	Reset applied initially	96.7%	Successfully compared
3	FIFO full condition	96.7%	Successfully compared
4	FIFO empty condition	96.7%	Successfully compared
5	FIFO write and read condition	90%	Successfully compared
6	Simultaneous write and read condition	100%	Successfully compared

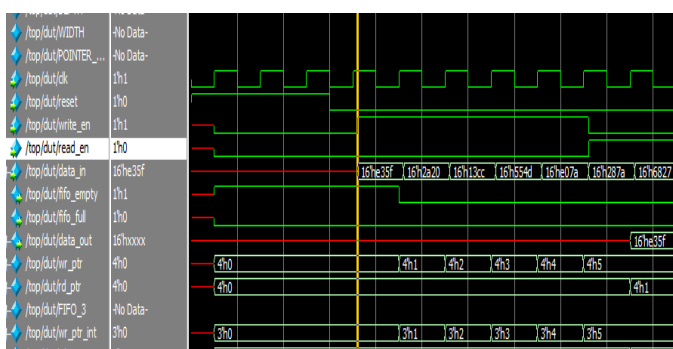


Fig.6:waveform of FIFO write and read condition

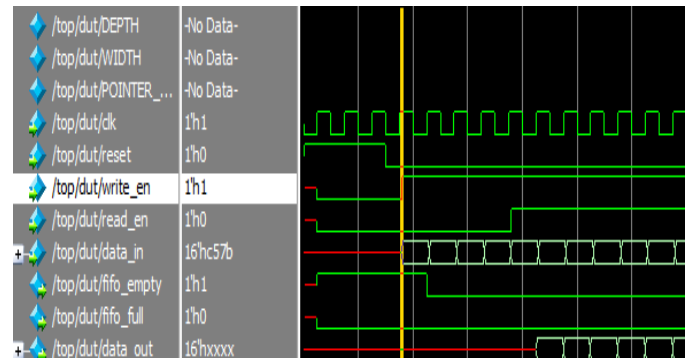


Fig.7:waveform of simultaneous write and read condition

ACKNOWLEDGMENT

I feel fortunate to have been under the of guidance of Dr. Kiran Bailey, Assistant Professor, ECE Department, BMSCE for her valuable pointers and support in this course of work and I would like to express my gratitude towards BMS college of Engineering Bengaluru for providing me an opportunity to carry out this work .

REFERENCES

- [1] Amr Moursi, Romaisaa Samhoud, Yaseen Kamal, Mazen Magdy, Sameh El-Ashry, Ahmed Shalab, "Different Reference Models for UVM Environment to Speed Up the Verification Time," 19th International Workshop on Microprocessor and SOC Test, Security and Verification (MTV), IEEE 2018.
- [2] Ritesh Kumar Agrawal, Vivek Ranjan Mishra, "The Design of High Speed UART," Proceedings of 2013 IEEE Conference on Information and Communication Technologies (ICT 2013).
- [3] Pavan Ambati, Mrudula Singamsetti "A New Approach for RFID Tag Data Reading in FPGA by using UART and FIFO," I.J.Engineering and Manufacturing, Published online march 2018 in MECS.
- [4] Sujata Mallappa Chajagouda Abdullah Gubbi, "Uesing Verilog and System Verilog Design and Verify the Communication Bridge between APB and I2C Protocol," IJSTE - International Journal of Science Technology & Engineering ,Volume 3 , Issue 01 ,July 2016.
- [5] Ms. Deepa Kaith, Dr. Janakkumar B. Patel, Mr. Neeraj Gupta "A Technical Road Map from System Verilog to UVM," International Journal on Recent and Innovation Trrends in Computing and Communication, Volume 3, Issue 3, March 2015.
- [6] Shumaila Qamar, Wasi Haider Butt, Muhammad Waseem Anwar, Farooque Azam and Muhammad Qasim Khan, "A Comprehensive Investigation of Universal Verification Methodology(UVM) Standard for Design Verification," ICSCA 2020, February 18–21, 2020, Langkawi, Malaysia.
- [7] Navaid Z. Rizvi, Rajat Arora and Niraj Agrawal, "Implmentation and Verification of Synchronous FIFO using System Verilog Verification Methodology," IEEE Transaction Journal of Communication Technology and Computer Science, Issue 2, 2015.
- [8] Wei Ni, Jichun Zhang, "Research of Reusability Based on UVM Verification," IEEE 11th International Conference on ASIC, 2015.
- [9] Khaled Salah, "A UVM-Based Smart Functional Verification Platform: Concepts, Pros, Cons, and Opportunities," 9th International Design and Test Symposium, IEEE 2014.