UVM Verification of DMA Controller

Neal Israni, Akshay Tamboli

Department of Electrical Engineering, San Jose State university, San Jose, California 95192.

Introduction

This paper demonstrates a verification plan to verify DMA controller core using Universal Verification Methodology (UVM). Functionality of the DMA controller can be verified by using the verification IP described in this paper. This paper incorporated a new approach to verify functionality, and Universal Verification Methodology was used to verify functionality of DMA controller. In this project, functional verification was done by applying test cases from test bench to the design under test. The test bench was created by using Universal Verification Methodology (UVM) standard libraries that were available in System Verilog. This paper also demonstrates how to build verification environment using Universal Verification Methodology (UVM) from scratch to verify the given device under test. In the semiconductor industry, verification has become one of the most important task. Any product should work according to the given specification. And therefore, verification plays a very important role in this industry. In this project, the design used is the DMA Controller.

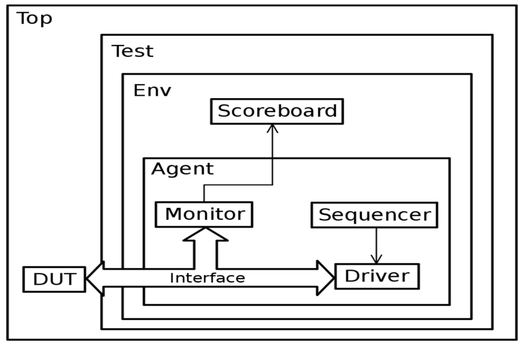
All the above components are the basic and very essential components of UVM. The use of all the above mentioned components makes the verification IP available for reuse . UVM helps in generating constrained random test cases for the purpose of verification. Using all such features of UVM, the design can thus be tested extensively.

Methodology

Universal Verification Methodology

The methodology one be used to verify functionality is Universal Verification Methodology and System Verilog for assertions. One will use Synopsys VCS simulator for the purpose of simulation.
Several components contributes to a UVM test bench. Diagram of a UVM test bench is shown below. It contains several components. The most important component of a UVM test bench is an agent. All the components that are needed for a specific protocol are contained in an agent. A monitor, a driver and a sequencer are the typical elements of a UVM agent. The main part of a sequencer is to create sequences. Sequences are the stimulus which are created and given to the design under test.
These sequences can either be directed data or constrained random. The flow of UVM shows that the sequence or the stimulus being generated by the driver is given to the driver. The interface is then driven by the driver.

Methodology



Driver

Driver is a device that matches the logic which is given to design under test. By sampling and driving the signals of design under test, driver receives data continuously, it then drives the data to design under test.

Sequencer

A sequencer is a device that generates the stimulus, which has a control over items being given to driver. Whenever requested, it returns a random data from driver.

Monitor

A monitor is a device that does not drive the signals of design under test but it samples them. It performs checking after it collects the coverage information.

Agent

Drivers, monitors and sequencers are all independent of each other and they can be reused. But they need an environment integrating device so that the names and roles of each of them can be hooked.

Advantages of UVM

- Backward compatibility with OVM.
- Flexible, configurable and layered testbench.
- Constrained random & coverage driven environment is created.
- Code reuse.

Verifying different modes

- Single read and write are tested by sending single address and data to the DUT.
- Burst mode is tested by sending a sequence of 8 address and data items at a time. The device has a buffer which stores 8 instructions.
- Sometimes, more than 8 data and address are also sent in burst mode just to verify whether the DUT sends the result as expected or not.

Directed Tests

•Directed test cases are written.

- Hardcoded values of address, data and response type are sent. These values are generated in sequencer as a part of sequence item.
- Sequence of operation and data content are known. Here the known data is send so that it is easy to verify the output from the DUT. Because, it can be compared with the expected result.
- The output of the DUT is monitored.

Random Tests

- Random test cases are written.
- Sequence of operation and address, data and instructions sent are random. This is done in order to increase the complexity of the tests.
- Constraints are added and randomized data packets are sent.
- Invalid data and address are sent and output of DUT is checked. This will actually tell whether the device is still throwing the output even after giving invalid set of input data.

Results

The DMA Controller is used as a design in this project. The verification environment is created for the verification purpose. This environment is created in system Verilog using standard libraries of Universal Verification Methodology (UVM). All the components that are required for the verification environment are created.
These components include a monitor, a sequencer, an agent, a sequencer. The test cases are then written in UVM and send it to the device under test which is the DMA Controller. It has been observed that all the signals that are being generated in the test bench are reaching to the DMA Controller.

Test cases are generated and the functionality of the design of DMA controller is tested. There are in all 27 instructions given in the specification of the DMA Controller to be tested. For this project, test cases are written for arithmetic and the logical instructions.

Number of instructions	Number of tests run	Directed tests	Random tests
15	736	300	436

Summary

The aim is to design the verification environment for the functional verification of DMA Controller using all the necessary components like a driver, a sequencer, a driver. Reusable IP has been designed in this project for the verification of the design. The design used in this project is the DMA Controller. This is the main advantage of this project. Any verification engineer can use this verification IP and verify any DMA Controller according to the specification. All the components which are needed for the verification environment are designed in this project.

Key References

[1] TMS320VC5501/5502 DSP Direct Memory Access (DMA) Controller Reference Guide

[2] Getting started with UVM, A beginner’s guide. Vanessa R. Cooper.

[3] A Practical Guide to Adopting the Universal Verification Methodology by Sharon Rosenberg, Kathleen Meade.

[4] SYSTEM VERILOG FOR VERIFICATION: A Guide to Learning the Testbench Language Features by Chris Spear.

[5] An improved DMA Controller for high speed data transfer in MPU based SOC by Hang Yuan, Hongyi Chen and Guogiang Bai

[6] The UVM Primer: A step by step introduction to the Universal Verification Methodology (UVM) by Ray Salemi.

Acknowledgements

Would like to express sincere thanks to Professor Morris Jones for his assistance in this project. His guidance helped me in learning UVM and implement that in this project to verify the functionality of DMA Controller.