Development of a functional verification enviroment based on UVM Express for a NoC 2x2

Edgar J. Domínguez Quintana, Valentín de Armas Sosa, Félix B. Tobajas Guerrero
IUMA, Institute for Applied Microelectronics
University of Las Palmas de GC, Spain

Abstract—**In this paper, the design and development of a functional verification environment to verify AMBA AXI4 interface protocol, is presented. The development is based on UVM Express methodology and was applied to a NoC 2x2 of Arteris over Questa SIM platform, and the results of the simulations set demonstrate that the developed environment (specially his graphical capabilities) makes easy the exploration of functional behavior of an AMBA AXI compliance component.**

Index Terms; Questa SIM, AMBA AXI4, NOC, UVM, UVM Express, BFM, driver, sequences.

I. INTRODUCTION

This paper describes the implementation of a functional verification environment to verify AMBA AXI4 interfaces [1]. The development is based in the Universal Verification Methodology (UVM) [2]- [4] and in the Questa SIM platform from Mentor Graphics [4]. Correctness of the environment was checked by simulating the behavior of a Network on Chip (NoC) 2x2. This NoC has AMBA AXI4 interfaces and was implemented with the FlexNoC tool from Arteris company [6].

II. GENERIC UVM TEST ARCHITECTURE

UVM verification test are built with a specific architecture based on reusable and configurable components verification (RCCV). The relationship between those components is established by the UVM class library hierarchy (Figure 1).

Test class is at the top level of the hierarchy. The next level corresponds to the Testbench component which can instances other verification modules at this level, called Universal Verification Components (UVCs). This level also includes other elements like the scoreboard or the bus monitor. UVM clearly defines the architecture of the UVC, which includes:

- Transactions. These are the lowest level component. They correspond to the information sent to and received from the DUV (Device Under Verification).
- Sequencer. This component generates and sends transactions to the driver.
- Driver. This component transforms transactions from the sequencer into RTL signal. It uses a bus functional model (BFM) to perform the DUV stimulation.
- Monitor. It´s a passive component to perform the checking to capture data and verifies the protocol specification.
- Agent. This is the top level component in the UVC hierarchy that encapsulates the mentioned other components like drivers, sequencers, etc.

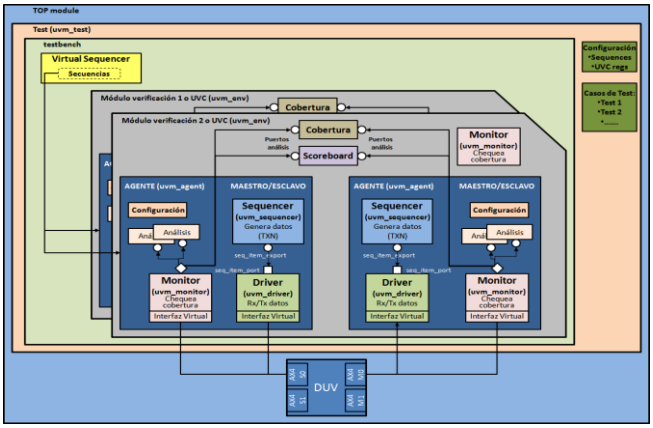

Figure 1. Full architecture of an UVM Verification Environment.

III. UVM EXPRESS

Due to the complexity of the UVM, it´s highly recommended in order to adopt the full UVM methodology, to start with a simplified version of UVM named UVM Express [4]. UVM Express is a subset of techniques, coding styles and UVM usages. It's composed of four phases (Figure 2).

The objective of the first three phases is to create the basic infrastructure of a typical UVM verification environment, which is composed of BFMs, monitors and random stimulus generators.

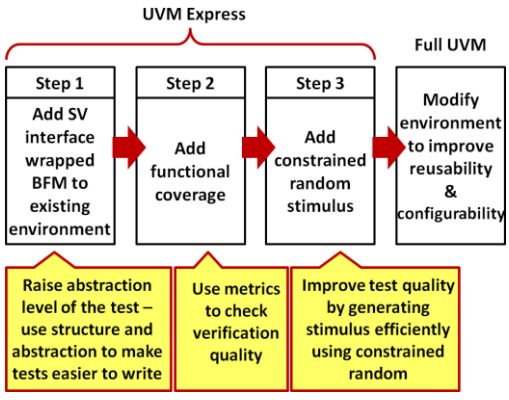

Figure 2. Phases of UMV Express.

Last phase involves the definitions of other components such as scoreboards, coverage modules, etc., which improve performance and configurability.

IV. Environment Implementation

Each component of the verification environment was implemented in a UVM Express phases. Firstlu, BFMs and monitors were developed, and secondly the random stimulus agent composed of transactions, sequences, sequencers and drivers was implemented, and finally, the coverage modules and global scoreboard were developed to complete the whole enviroment.

Due to AMBA AXI channels are concurrents, a library of sequences was developed to execute BFMs tasks in both sequential and concurrent mode. These sequences are directly executed by a unique driver which controls five AMBA AXI channels, as shown in figure 3.

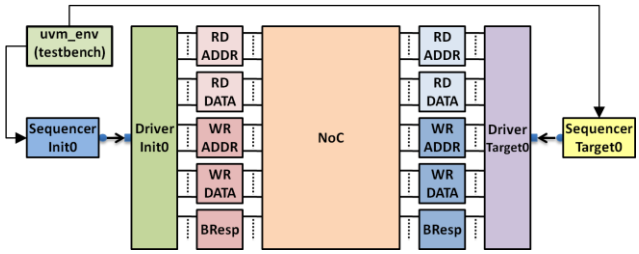

Figure 3. Driver throws BFM tasks in parallel.

Communication between sequence and driver follow a handshake protocol as illustrated in Figure 4. This protocol enables to exchange sequences of transactions between these components.

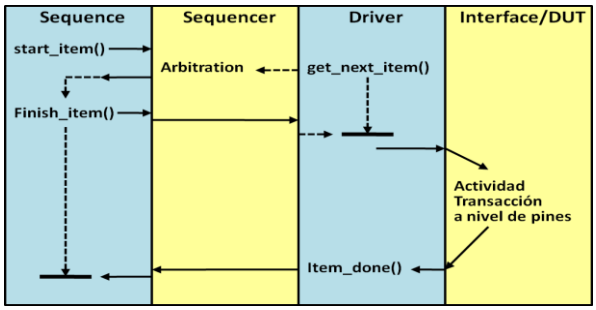

Figure 4. Handshake protocol between drive and sequencer components.

The main goal of the monitor is to detect transactions and to pass these transactions to the coverage module and scoreboard. This last uses this information to establish, by comparing and at the end of the simulation, the correctness of transactions processing.

V. The Simulation and Results

Simulations were done on Questa SIM platform. From these simulations, it has been proved that Arteris´s NoC is compatible with AMBA AXI4, but doesn´t support the following AMBA AXI4 features:

- Neither length transactions up 256 bytes nor transfers different of 4 bytes are supported.
- FIXED, RESERVED and WRAP burst. Although NoC accepts 2/4 bytes WRAP request, the NoC doesn´t process them right.
- Neither exclusive nor locked access transactions, nor narrow transfers, are supported.
- Neither AMBA AXI3 nor Write Data Interleaving (WDI), nor cache support and access protection are developed.
- Neither quality of Service, regions and user signals (AXQOS, AXREGION, AXUSER, XUSER y BUSER) are implemented.

VI. Conclusions and Future Work

The developed verification Environment is suitable to be used on according any DUV with AMBA AXI4 interfaces, and it´s easily modifiable to adjust coverage modules and generation stimulus process, according to the particular features of the other DUVs.

One of the most important contributions of this Final Master Thesis is the developed library of sequences, because it´s able to stimulate AMBA AXI channel in concurrent mode, and also the auto verification carries out by the scoreboard.

Finally, a special interesting part of the developed code is the class that allows a hierarchical display of transactions and their phases over the Questa SIM platform. The hierarchical displaying dramatically increases the detection capability of singularities and anomalies in DUV´s. Indeed, this class has allowed the discovery of the *Read Over Write* strategy implementation, and the abnormalities when using the WRAP transactions.

The developed environment leaves open the option of incorporating new functionalities, such as:

- To modify driver/sequencer behavior to support out-of-order operations, pipeline [7], and full independency in the stimulation of AMBA AXI channels.
- To modify components to support AMBA AXI3 and other protocols, and create a full protocol checker, implemented as a separate component.
- To invoke sequences from command line and text file, using the *uvm_cmdline_processor* class, to facilitate performing the test.

References

[1] ARM Holdings Plc. "*AMBA AXI and ACE Protocol Specification*" (on line). Available on http://www.arm.com/products/system-ip/amba/amba-open-specifications.php, visited November 3, 2013.

[2] Accellera, "*Universal Verification Methodology (UVM) 1.1 User's Guide*", Accellera Organization. Napa, CA, May 2011.

[3] Mentor Graphics, "*Mentor Graphics UVM/OVM Documentation. Verification Methodology Online Cookbook*" (on line). Available on: http://www.mentor.com/products/fv/techpubs, visited June 10, 2013.

[4] Mentor Graphics, "*Overview UMV Express. Cookbook.*" (on line). Available on: http://verificationacademy.com, visited June 10, 2013.

[5] Mentor Graphics, "*Functional Verification. Questa Advanced Simulator*", (on line). Available on: http://www.mentor.com/products/fv/questa, visited August 15, 2013.

[6] Arteris, "*NoC Interconnect Technology for SoCs*" (on line). Available on: http://www.arteris.com/technology.php, visited November 3, 2013.

[7] Rich Edelman and Raghu Ardeishar, "*Sequence, Sequence on the Wall – Who's the Fairest of Them All? Using SystemVerilog UVM Sequences for Fun and Profit*", *DV-CON 2013,* (on line). Available on: https://s3.amazonaws.com/verificationacademynews/DVCon2013/Papers/MGC_DVCon_13_Sequence_on_the_Wa..pdf, visited Nov. 15, 2013.