ijecs open access www.ijecs.in
International Journal Of Engineering And Computer Science ISSN:2319-7242
Volume 6 Issue 6 June 2017, Page No. 21870-21874
Index Copernicus value (2015): 58.10 DOI: 10.18535/ijecs/v6i6.53

A Design of AMBA AXI4-Lite ACE Interconnect Protocol for Transaction-based SoC Design Techniques Integration

Chiranjeet Kumar[1] *Dr. M. Gurunadha Babu*[2]

[1]M.Tech, Department of VLSI system design [2]M.Tech, Ph.D., Head of the Department (ECE)

CMR Institute of Technology, Medchal Road, Hyderabad, Telangana 501401-India

chiranjeet9650530563@gamil.com[1] mgurunadhababu@gmail.com[2]

Abstract-- ***The chip design in the 21st century has undergone various changes due to the increased customer demands and this lead to design complexity in systems-on-chip (SoC), network-on-chip (NoC), application-specific integrated circuit (ASIC), and field-programmable gate array (FPGA) designs. This creates a situation to develop an advanced system to resolve the complexity issue in meantime. The verification step consumes the major portion of the VDHL time and transaction-level modeling (TLM) and Bus Functional modeling (BFM) are used in order to reduce this effort. Transaction-level modeling (TLM) is a technique used to describe the system by using the standard function calls which defines all the transactions which are required to verify the functionality of the system at the architecture level. The usage of the transaction based techniques are designed for the software analysis and for the first time, in this research work it is used for the physical hardware design and its analysis based on the AMBA ace-lite architecture. In past AMBA AXI4 Bus Interconnects is used for the hardware system design but it fails to meet the practical design requirements and the proposed AMBA ace-lite architecture has yielded the desired results with low complexity. With the proposed AMBA ace-lite architectural design for hardware system design, several SoC/NoC subsystems can easily be interconnected in basically the same manner as how transaction-based simulation models are being written. The proposed methodology is useful for the hardware design engineers to deal with the complexity simplification issues by bringing the benefits of transaction-based verification (TBV) to it approach.***

Index Terms: Transaction Level modeling (TLM), Advanced Extensible interface (AXI), Advanced Micro Controller Bus Architecture (AMBA), FPGA, Software, Hardware

1. Introduction

To meet the customer demands and the time to market always designers and the verification engineers look for methods which can reduce the effort as well as the time. Adopting Transaction-level modeling (TLM) technique and developing intellectual properties (IP) and flexible automated tools for design as well as verification are some of the methods which are targeted towards the same.

In the early days as each chip manufacture had their own bus, standardization of the bus become necessary to enable the reusability of the intellectual properties (IP).Many manufacturer developed standard busses among which some of them become very popular due to their performance, hierarchy and the advanced features.

Standard busses which become very popular are Advanced Micro Controller Bus Architecture (AMBA) from ARM, CORE CONNECT from IBM. Using these standards interconnects not only enables intellectual properties (IP) reusability, provides flexibility, compatibility. These interconnects have multi-layer architecture. Between masters and slaves it can be used as a crossbar switch. Developing and verifying these interconnects or the busses become important. Because of the availability of wide variety of intellectual properties (IP) from ARM and advanced intellectual properties (IP) from third party vendors for Advanced Micro Controller Bus Architecture (AMBA) bus, this is more popular than others. There are various versions of Advanced Micro Controller Bus Architecture (AMBA) from AMBA1.0 to AMBA 5.0.

On the other end of the design flow, a pure logic simulation can take place at the register transfer level (RTL). In a conventional SoC logic simulation, RTL models written in hardware description language (HDL) such as VHDL and Verilog are employed as the system hardware. If a processor model is necessary, a design sign-off model (DSM) will typically be used. The advantage of the logic simulation is evidently its great fidelity to the real implementation, i.e. accurate SoC functional and performance analysis. This is nonetheless a price too expensive to pay in terms of the lengthy simulation time. The time consumption has actually further worsened lately due to the high SoC complexity that requires a longer RTL development phase. Moreover, a pure logic simulation cannot execute any software in a reasonable amount of time. A system can only integrate its associated software for observation and analysis rather late in the

DOI: 10.18535/ijecs/v6i6.53

design flow. Since the breadboard is usually almost ready at this point, any system modification will certainly be too costly at this stage.

In brief, an in-between solution has to be resolved for which three fundamental criteria must always be respected as the doorway to early software development and architecture exploration:

1. Speed. The potential model must simulate millions of cycles within a reasonable time length. The target activities frequently involve a very large scale of simulation cycles. Some of them may entail user interactions that could probably slow down the process. It is unacceptable and unaffordable to wait for even just a day to complete a simulation run.

2. Accuracy. Although speed is an interesting advantage to enhance, the potential model should sustain a certain degree of accuracy to deliver reliable simulation results. Some of the analyses may require full-cycle accuracy to obtain adequate outcomes. As a rule of thumb, the potential model should at least be detailed enough to run the related embedded software.

3. Lightweight Modeling. Any other modeling effort in addition to the compulsory RTL modeling for hardware synthesis must be kept insubstantial to optimize the overall SoC project cost. The potential model should be, for this reason, a quick-to-develop model at a considerably low effort.

In conclusion, the ultimate goal of TLM is leading the SoC industry to a cost- and time-efficient SoC project management in the long run.

2. Literature Review

Syed Saif Abrar et. al. [1] has told that with the acceptance of TLM methodology for system level design, IP customers need ever increasing support for TLM models. They faced time and accuracy challenges while manually converting RTL to TLM. Hence, author here presented an approach to convert automatically RTL-VHDL to system C. Also, abstraction of clock interface has provided about 38% increase in simulation time.

Adam Rose et. al. [2] has proposed OSCI (Open SystemC Initiative) TLM standard and has shown how to use it to solve common modeling problems. Also, further the core TLM proposal is being explained in detail and has given how to combine different levels of abstraction in the same TLM and how to approach towards common modeling problems.

Jian Wang[6] et. al. Discussed a Dynamic priority Arbiter for network-on-chip based on lottery mechanism. This dynamic Arbiter detects the load of input ports in each and every clock cycle and adapts the priority of every input port dynamically, then adjust the input ports to transfer data between master and slaves based on lottery mechanism. And compared the performance of Round-Robin arbitration algorithm and Dynamic Lottery based arbitration algorithm in his paper. And they found that the dynamic lottery based priority Arbiter is useful to improve the communication performance of network-on-chip.

IEEE Standard for Standard System C® Language Reference Manual-1666-2011[7].This is the complete reference manual for System C. It describes complete language for beginners step by step. It includes language definitions, predefined channel class, data types, utilities, etc. Further it has also given TLM2.0 overview, interfaces, generic payloads, base protocol and phases, etc.

Divekar, Shraddha Tiwari [8] et. al. proposed the Multichannel AMBA AHB with multiple arbitration techniques. The SoC design requires a system bus with high bandwidth and system design complexity has become higher. This multi-channel AMBA AHB bus matrix has been widely used in many SOC designs. In this paper following multiple arbitration techniques are used such as round robin, fixed priority are used. AMBA AHB is basically single layer bus, and the interconnect matrix for AMBA AHB which includes four masters and four slaves and also increases or decreases the number of masters and slaves as par user requirement. In this paper, a new Arbiter is used by combine the round robin Arbiter and dynamic Arbiter we, also design the new Arbiter which removes the dynamic Arbiter disadvantages. By using Xilinx simulators and by using FPGA kit, this provides flexibility and high density. AMBA-AHB protocol using multiple arbitration techniques which provide interconnection scheme between multiples masters. This can be achieved the assistance of both increased overall bus bandwidth and more flexible system structure. The multichannel-AHB bus matrix uses central arbitration. The designs of a central Arbiter for interconnect bus matrix supports the priority policy as multiple arbitrations priority policy.

Laurent Maillet-Contoz et. al.[9] Author in this book has put forward TLM based models to remove RTL design for SoC design flow. It also formalizes TLM abstractions that offer untimed and timed models to tackle with System on Chip design activities ranging from early software development to architecture analysis and also functional verification. The most adopting benefit of TLM is the true hardware/software co-design founded on a unique reference, that results in reduced time-to-market and comprehensive cross-team design methodology.

Jon Connell et. al.[10] has proposed early hardware and software generation of systems enabled using systemC. The system being represented at transaction level shares common abstraction and verification environment.Further processor model is added in system that enables verification of hardware and software interactions. Also, to the level down of verifying the interactions of software, processors, RTOS and hardware subsystems is done.

Preeti Ranjan Panda[11] has described about the features of systemC which is used for modeling the platform at multiple level of abstractions.It supports abstraction at register transfer level, behavioural and at system levels.The language is supported by OSCI, a consortium of wide range of system houses, Intellectual properties(IP), semiconductor companies, etc. The main advantages of using SystemC includes establishment of a common design environment consisting of core C++ libraries, models and tools thereby setting up a foundation for hardware-software co-design and gives the ability to reuse test benches across different levels of modeling abstraction.

3. Existing Method

AMBA AXI architecture

AMBA AXI supports data transfers up to 256 beats and unaligned data transfers using byte strobes. In AMBA AXI4 system 16 masters and 16 slaves are interfaced. Each master and slave has their own 4 bit ID tags. AMBA AXI system consists of master, slave and bus.

The system consists of five channels namely write address channel, write data channel, read data channel, read address channel, and write response channel.

The AXI4 protocol supports the following mechanisms:

- Unaligned data transfers and up-dated write response requirements.
- Variable-length bursts, from 1 to 16 data transfers per burst.
- A burst with a transfer size of 8, 16, 32, 64, 128, 256, 512 or 1024 bits wide is supported.
- Updated AWCACHE and ARCACHE signaling details.

Each transaction is burst-based which has address and control information on the address channel that describes the nature of the data to be transferred. The data is transferred between master and slave using a write data channel to the slave or a read data channel to the master.

The write operation process starts when the master sends an address and control information on the write address channel as shown in fig 1. The master then sends each item of write data over the write data channel. The master keeps the VALID signal low until the write data is available. The master sends the last data item, the WLAST signal goes HIGH.

4. Proposed Methodology

Definition of AXI4-Lite

The key functionality of AXI4-Lite operation is:

- All transactions are of burst length 1
- All data accesses use the full width of the data bus — AXI4-Lite supports a data bus width of 32-bit or 64-bit.
- All accesses are Non-modifiable, Non-bufferable
- Exclusive accesses are not supported.

Due to the increased customer demands design complexity of system on chip (SOC) increases day by day. Hence there is always a productivity gap. To address this issue various advanced methods are adopted during the design, development and verification phase of any project. It might be developing an Intellectual property (IP), using an automated tool, hardware software co design or using various modeling methodologies at the earlier phase of the project for system level architecture exploration.

In this paper we are discussing two modeling techniques to develop and verify the Advanced Extensible interface (AXI4) bus interconnect , they are Register Transfer level (RTL) method and Transaction Level modeling (TLM) method. From this analysis we come to the conclusion that using TLM method increases the simulation speed, and reduces the effort due to the availability of open source packages which can support the Transaction Level modeling (TLM) method.

There are various methods of modeling the system or an interconnect to verify the functionality at the architecture level like RTL, cycle accurate (CA), temporal model and TLM model, Result oriented model (ROM). Each has its own disadvantage like, RTL method requires more simulation time, Cycle accurate model cost is more, Result oriented model is platform based. But TLM stands better compared to other modeling methodologies.

In Transaction level model [TLM], Transaction is an “object that encompasses the handshakes and the signals which are required to establish the communication between the components or the modules”. Here the communication is performed by using the functional calls [fig 1]. All the components of the system are represented in terms of TLM modules. The TLM channels are used to connect different modules. Modules are bound to channels through the TLM ports. The module that requests the transaction is called the TLM master or TLM initiator. And the module which does the requested operation is called the TLM target or TLM

DOI: 10.18535/ijecs/v6i6.53

slave. Using TLM raises the abstraction level above the RTL. Hence it is placed above the RTL level in the SOC design flow.

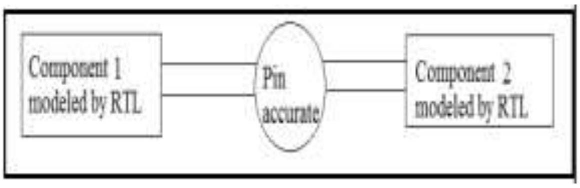


Fig.1 (a): RTL model simulates every event

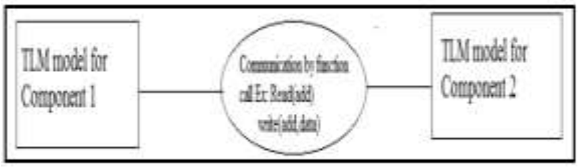
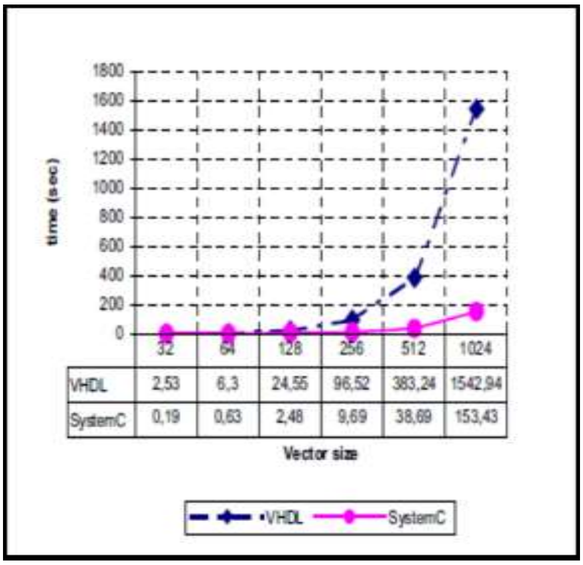

TLM model for Component 1
Communication by function call Ex: Read(add) write(add, data)
TLM model for Component 2

Fig.1 (b): TLM model, Simulation speed higher then RTL

A simulation time for system C TLM model of R8 processor is less compared to RTL model developed in VHDL. Also it concludes that hardware size obtained by using RTL of VHDL and the RTL of system C is the same.

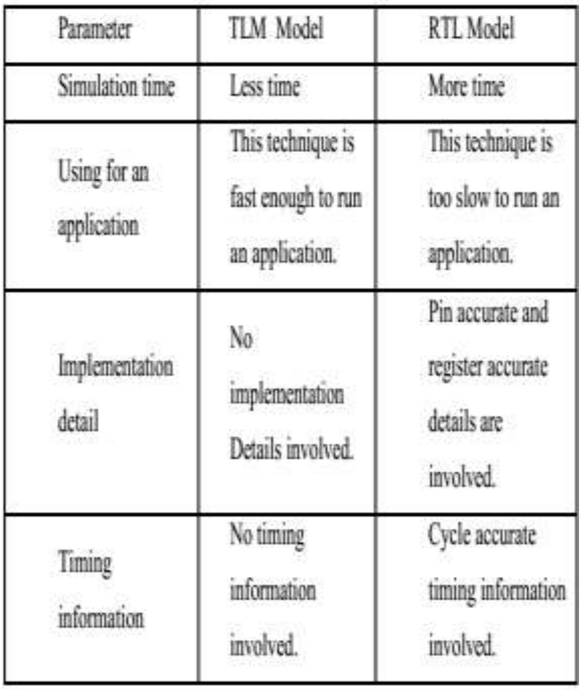


	32	64	128	256	512	1024
VHDL	2,53	6,3	24,55	96,52	383,24	1542,94
SystemC	0,19	0,63	2,48	9,69	38,69	153,43

Fig.2: RTL versus TLM simulation time Comparison

Table 2 gives the comparison of RTL and TLM model method.

Table 2: RTL and TLM model method



Parameter	TLM Model	RTL Model
Simulation time	Less time	More time
Using for an application	This technique is fast enough to run an application.	This technique is too slow to run an application.
Implementation detail	No implementation Details involved.	Pin accurate and register accurate details are involved.
Timing information	No timing information involved.	Cycle accurate timing information involved.

Advantages of this methodology are:

- It works with VHDL2002 but is mainly based on VHDL 2008.
- Random generation and functional coverage present here are having advanced features.
- Randomized values also support various distributions like Gaussian and Poisson distribution.
- The Transaction level modeling (TLM) can also be implemented using these packages.
- Reporting features for functional coverage is also been implemented.
- Randomizing the values is done by checking whether all the possibilities are covered or not. This feature is called intelligent randomization.

Following packages are taken from this forum to support the TLM method in this project

- Random package.
- Coverage package.

Random package helps to generate the required data randomly. Coverage package is used to check whether all the data generated by the random package covers all the scenarios or not, So as to ensure the correct functionality of the system. These packages have standard function calls .These calls are used to achieve the transaction level modeling.

5. Conclusion

Transaction level modeling (TLM) is put forward as a promising solution above Register Transfer Level (RTL) in the SoC design flow. This chapter formalizes TLM abstractions to offer untimed and timed models to tackle SoC design activities ranging from early software development to architecture analysis and functional verification. The most rewarding benefit of TLM is the veritable hardware/software co-design founded on a unique reference, culminating in reduced time-to-market and comprehensive cross-team design methodology. From the above analysis we can conclude that using TLM technique to develop and verify the bus interconnect i.e. AXI4 is better than the RTL technique. Also the availability of the VHDL open source packages reduces the time and the effort

References

[1] Syed Saif Abrar, Maksim Jennihin, Jaan Raik, “Abstraction of clock interface for conversion of RTL VHDL to SystemC,” in proc. IEEE International Advance Computing Conference, pp.50-55, Feb 2012.

[2] Adam Rose, Stuart Swan, John Pierce, Jean-Michel Fernandez, “Transaction Level Modeling in SystemC,”Cadence Design Systems, Inc,pp.1-17.

[3] Fan-Gang Zeng, Stephen Rebscher, William Harrison, Xiaoan Sun and Haihong Feng, “Cochlear Implants: System Design, Integration, and Evaluation”, in proc. IEEE Reviews in Biomedical Engineering, vol. 1, pp. 115 – 142, 2008.

[4] P. Mahalakshmi, M. R. Reddy, “Signal Analysis by Using FIR Filter Banks in Cochlear Implant Prostheses”, in proc. International Conference on Systems in Medicine and Biology, IEEE, pp. 253 - 258, December 16-18, 2010.

[5] Lunner T., Hellegren J., Arlinger S., Elberling C., “Non-Linear Signal Processing in Digital Hearing Aids” in proc. Scand Audiol IEEE , pp. 40 – 49, 1998.

[6] Jian Wang, “Arbiter for network-on-chip based on lottery mechanism, pp.3029-3038, Feb 2013.

[7] IEEE Standard for Standard System C® Language Reference Manual-1666-2011.

[8] S. Divekar and A. Tiwari, “Multichannel AMBA AHB with multiple arbitration technique,” Int. Conf. Commun. Signal Process. ICCSP 2014 - Proc., pp. 1854–1858, 2014. [10]“wang2009.pdf.”

[9] Laurent Maillet-Contoz and Frank Ghenassia, “Transaction Level Modeling-An Abstraction Beyond RTL,”Springer Publication, 2005.

[10]Jon Connell, Bruce Johnson, “Early Hardware/Software Integration Using SystemC 2.0”, Class 552, ESC San Francisco 2002

[11]Preeti Ranjan Panda, “SystemC- A modeling platform supporting multiple design abstractions”, Synopsys Inc. 700 E. Middlefield Rd. Mountain View, CA 94043, USA, pp. 75-80, Oct. 2001.