International Journal of Research
Available at https://edupediapublications.org/journals

e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 05 Issue-01
January 2018

VLSI Design of AMBA ACE4-Lite Interconnect protocol for SOC

Chiranjeet Kumar[1] , Dr. M. Gurunadha Babu[2] , Prof. Suneel Laxmipuram[3]
chiranjeet9650530563@gmail.com[1], mgurunadhababu@gmail.com[2], suneel.430@mail.com

[1]PG Scholar, VLSI, CMR institute of technology, Kandlakoya, Medchal, Hyderabad, Telangana.
[2]Professor & Head, Department of ECE, CMR institute of technology, Kandlakoya, Medchal, Hyderabad,
[3]Professor CMR institute of technology, Kandlakoya, Medchal, Hyderabad, Telangana.

Abstract: *Advanced microcontroller bus architecture (AMBA) protocol family provides metric-driven verification of protocol compliance, enabling comprehensive testing of interface intellectual property (IP) blocks and system-on-chip (SoC) designs. 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. 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.

I. INTRODUCTION

In recent years, the improvement of the semiconductor process technology and the market requirement increasing. More difference functions IPs are integrated within a chip. Maybe each IPs had completed design and verification. But the integration of all IPs could not work together. The more common problem is violation bus protocol or transaction error. The bus-based architecture has become the major integrated methodology for implementing a SoC. The on-chip communication specification provides a standard interface that facilitates IPs integration and easily communicates with each IPs in a SoC. To speed up SoC integration and promote IP reuse, several bus-based communication architecture standards have emerged over the past several years. Since the early 1990s, several on chip bus-based communication architecture standards have been proposed to handle the communication needs of emerging SoC design. Some of the popular standards include ARM Microcontroller Bus Architecture (AMBA) versions 2.0 and 3.0, IBM Core Connect, STMicroelectronics STBus, Sonics SMARRT Interconnect, Open Cores Wishbone, and Altera Avalon [1]. On the other hand, the designers just integrate their owned IPs with third party IPs into the SoC to significantly reduce design cycles. However, the main issue is that how to efficiently make sure the IP functionality, that works correctly after integrating to the corresponding bus architecture.

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.

II. AMBA AXI ARCHITECTURE

The AMBA AXI protocol supports high-performance, high-frequency system designs. The AXI protocol:

- is suitable for high-bandwidth and low-latency designs
- provides high-frequency operation without using complex bridges
- meets the interface requirements of a wide range of components
- is suitable for memory controllers with high initial access latency
- provides flexibility in the implementation of interconnect architectures
- is backward-compatible with existing AHB and APB interfaces.

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.

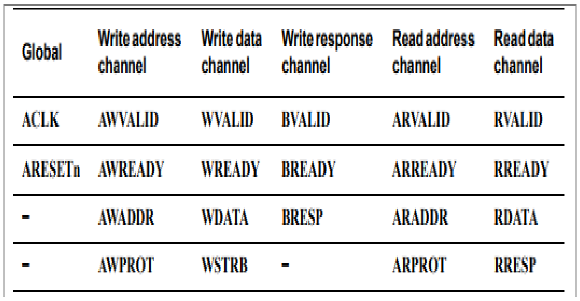
The write operation process starts when the master sends an address and control information on the write address channel as shown in fig 6. The master then sends each item of write data over the write data channel. The master keeps the VALID signal low until the write data is available. The master sends the last data item, the WLAST signal goes HIGH.

III. PROPOSED SYSTEM

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.



Global	Write address channel	Write data channel	Write response channel	Read address channel	Read data channel
ACLK	AWVALID	WVALID	BVALID	ARVALID	RVALID
ARESETn	AWREADY	WREADY	BREADY	ARREADY	RREADY
-	AWADDR	WDATA	BRESP	ARADDR	RDATA
-	AWPROT	WSTRB	-	ARPROT	RRESP

Table B1-1 AXI4-Lite interface signals

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 6]. 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 slave.

EXTENSION.

AMBA ace-lite architecture.

ACE-Lite masters have the additional signals on AXI channels but do not have the additional three ACE snoop channels. ACE-Lite masters can perform transactions only from the Non-shared, Non-cached and Cache Maintenance transaction groups. ACE-Lite enables un-cached masters to snoop ACE coherent masters. This can enable interfaces such as Gigabit Ethernet to directly read and write cached data shared within the CPU. Going forwards, ACE-Lite is the preferred technique for I/O coherency and should be used where possible rather than the Accelerator Coherency Port for best power and performance. Cortex-A15 supports an optional ACP primarily for designs including legacy IP that is not ACE-Lite compliant or designs that are upgrades from other MP Core technology-enabled processors.

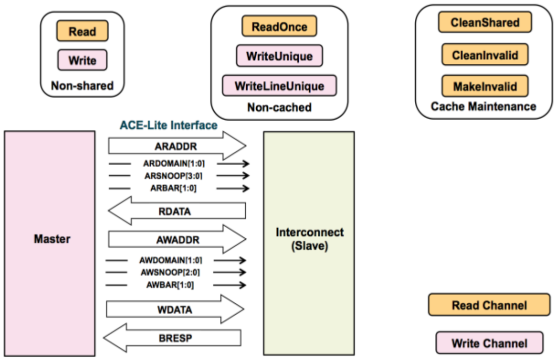
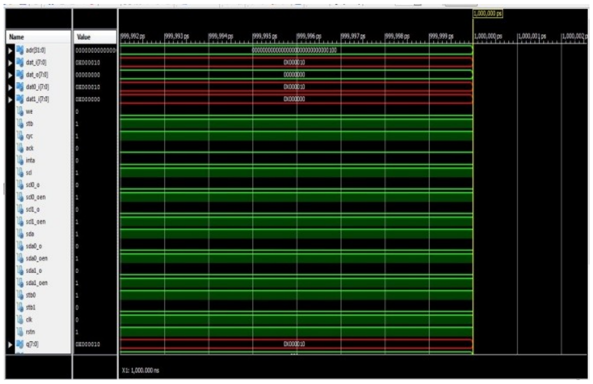


Figure 1: ACE-Lite sub-set

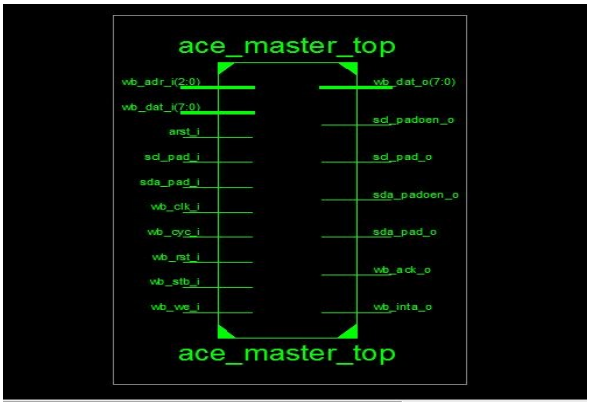
IV. RESULTS

SIMULATION RESULT



```
Name          Value
adr[31:0]     000000000000
dat_i[7:0]    0X000010
dat_o[7:0]    00000000
dat0_i[7:0]   0X000010
dat1_i[7:0]   0X000000
we            0
stb           1
cyc           1
ack           0
inta          0
scl           1
scl0_o        0
scl0_oen      1
scl1_o        0
scl1_oen      1
sda           1
sda0_o        0
sda0_oen      1
sda1_o        0
sda1_oen      1
stb0          1
stb1          0
clk           0
rstn          1
q[7:0]        0X000010
```

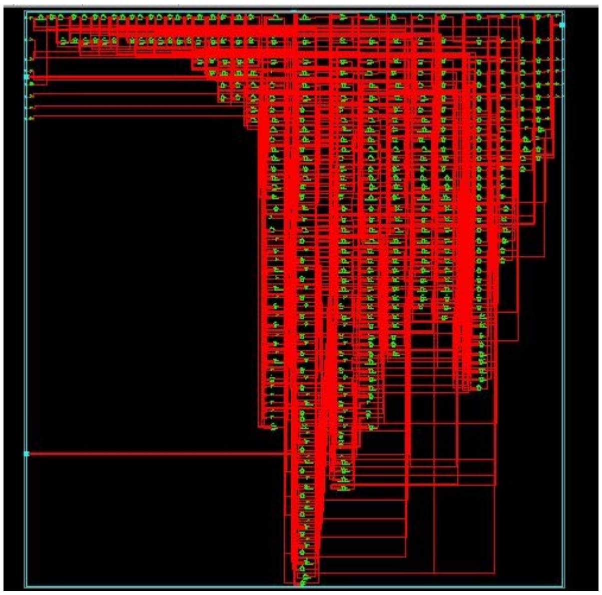
RTL schematic:



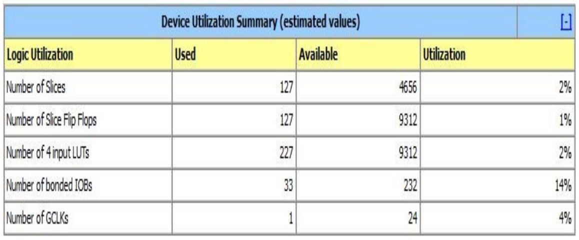
TECHNOLOGY SCHEMATIC:

International Journal of Research
Available at https://edupediapublications.org/journals

e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 05 Issue-01
January 2018

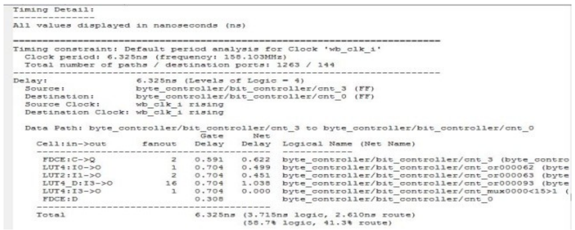


DESIGN SUMMARY:



Device Utilization Summary (estimated values)			[-]
Logic Utilization	Used	Available	Utilization
Number of Slices	127	4656	2%
Number of Slice Flip Flops	127	9312	1%
Number of 4 input LUTs	227	9312	2%
Number of bonded IOBs	33	232	14%
Number of GCLKs	1	24	4%

TIMING REPORT:



```
Timing Detail:
--------------
All values displayed in nanoseconds (ns)

=========================================================================
Timing constraint: Default period analysis for Clock 'wb_clk_i'
  Clock period: 6.325ns (frequency: 158.103MHz)
  Total number of paths / destination ports: 1263 / 144
-------------------------------------------------------------------------
Delay:               6.325ns (Levels of Logic = 4)
  Source:            byte_controller/bit_controller/cnt_3 (FF)
  Destination:       byte_controller/bit_controller/cnt_0 (FF)
  Source Clock:      wb_clk_i rising
  Destination Clock: wb_clk_i rising

  Data Path: byte_controller/bit_controller/cnt_3 to byte_controller/bit_controller/cnt_0
                                Gate     Net
    Cell:in->out      fanout   Delay   Delay  Logical Name (Net Name)
    ----------------------------------------  ------------
     FDCE:C->Q             2   0.591   0.622  byte_controller/bit_controller/cnt_3 (byte_contro
     LUT4:I0->O            1   0.704   0.499  byte_controller/bit_controller/cnt_or000062 (byte
     LUT2:I1->O            2   0.704   0.451  byte_controller/bit_controller/cnt_or000063 (byte
     LUT4_D:I3->O         16   0.704   1.038  byte_controller/bit_controller/cnt_or000093 (byte
     LUT4:I3->O            1   0.704   0.000  byte_controller/bit_controller/cnt_mux0000<15>1 (
     FDCE:D                    0.308          byte_controller/bit_controller/cnt_0
    ----------------------------------------
    Total                      6.325ns (3.715ns logic, 2.610ns route)
                                       (58.7% logic, 41.3% route)
```

V. CONCLUSION

Transaction level modeling (TLM) is put forward as a promising solution above Register Transfer Level (RTL) in the SoC design flow. This paper 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.

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., ―NonLinear 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.