Transaction based AMBA AXI bus interconnect in Verilog

Shubhi Sharma[1], Vidyadhar Jambhale[2], Abhijeet Shinde[3], S.Ravi[4]

[1]*M.Tech VLSI Design, Vellore Institute of Technology, Vellore*
[2]*M.Tech VLSI Design, Vellore Institute of Technology, Vellore*
[3]*M.Tech VLSI Design, Vellore Institute of Technology, Vellore*
[4]*Professor, Dept. of electronics Engineering, Vellore Institute of Technology, Tamil Nadu, India*

---***---

Abstract - *The AMBA AXI [1] convention is a standard transport convention and the vast majority of the semiconductor companies plan interconnects which underpins AXI transport interface. AXI convention is unpredictable convention on account of its ultra-elite. The ARM Advanced Microcontroller Bus Architecture (AMBA) [3] is an open-standard, on-chip interconnect determination for the association and administration of practical squares in framework on-a-chip (SoC) [3] plans. We are computing the read and write operations along with the latency calculation by growing the RTL Verilog coding and synthesizing it in Synopsys tool in 90nm where area is reduced by 48.04% and power is reduced by 31.74% and in 32nm technology where area gets deduced by 9.25% and power by 2.83%.*

Key Words: AMBA, AXI, SOC, transaction, channel, constraints, RTL

1.INTRODUCTION

The essential part of a SoC is which segments or pieces it houses, as interconnect [3]. AMBA is an answer for the well as how they square each other. These conventions are today the interface with true standard for 32-bit inserted processors as they all are around reported and can be utilized without eminences. The AMBA AXI 4 convention underpins elite, high-recurrence framework outlines. It is reasonable for the high-transmission capacity, also low-inertness plans and gives high-recurrence operation without making utilization for complex extensions. It gives adaptability in the execution of interconnect structures and is in reverse good along with prevailing AHB and APB interfaces. This venture is done for the outline of the AXI master Interface and usage utilizing Verilog RTL coding. This master interface can be utilized to associate diverse secondary within AMBA construct processors in the absence of bridge. The created slave interface can likewise be utilized to associate distinctive secondary like SPI, I2C, UART and so forth, within non AMBA based processors by creating wrapper around AXI slave interface [5]. It is a part of the Advanced Microcontroller Bus Architecture (AMBA) created by ARM (Advanced RISC Machines) [5] organization. The key components of the AXI convention comprises discrete location/control and information stages and backing for unaligned information exchanges, utilizing byte strobes. It uses burst-based exchanges with just the begin address issued. It has separate perused and compose information channels that give minimal effort Direct Memory Access (DMA). It bolsters for issuing various extraordinary locations. It support for out-of-request exchange fulfillment. It allows simple expansion of register stages to give timing conclusion

2. AMBA AXI

A regular framework comprises of various master and slave gadgets associated together through the Interconnect. The AXI convention gives a solitary interface denotation, for the interfaces [5][6]: between a master and the interconnect, between a slave and the interconnect, between a master and a slave [5][6]. The AXI convention is burst based and characterizes the accompanying autonomous exchange channels: Read Address Channel, read Data Channel, Write Address Channel, Write Data Channel and the Write Response Channel [5][6]. A location channel conveys control data that portrays the way of the information to be exchanged. The information is exchanged in the middle of expert and slave utilizing either:

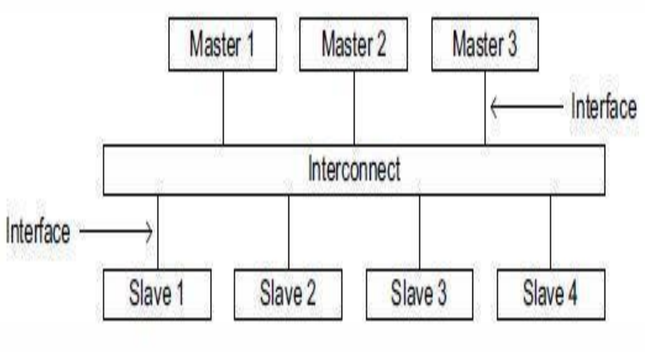

Fig -1: AXI protocol

A compose information channel to exchange data between expert and the slave [6]. In a compose exchange, the slave utilizes the compose reaction channel to flag the finishing of the exchange to the expert. A read information channel to exchange information from the slave to the expert. The AXI convention grants address data to be issued in front of the genuine information exchange. It bolster different remarkable exchanges. It additionally bolsters out-of-request finish of exchange.

2.1 AXI channel description

AMBA AXI bolsters information exchanges up to 256 bits and unaligned information exchanges utilizing byte strobes. In AMBA AXI framework 16 experts and 16 slaves are interfaced. Every expert and slave has their own particular 4 bit ID labels [7]. AMBA AXI4 framework comprises of expert, slave and transport. The framework comprises of five channels in particular compose address channel, write data channel read data channel, read address channel, and write response channel [8]. The AXI convention underpins the accompanying instruments: Unaligned information exchanges and up-dated compose reaction prerequisites. Variable-length blasts, from 1 to 16 information exchanges per burst. A burst with an exchange size of 8, 16, 32, 64, 128, 256, 512 or 1024 bits wide is bolstered [6]. Upgraded AWCACHE and ARCACHE signaling details [5].

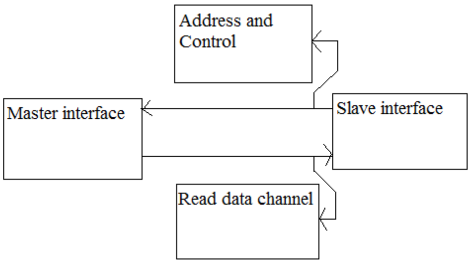

Fig -2: Read transaction

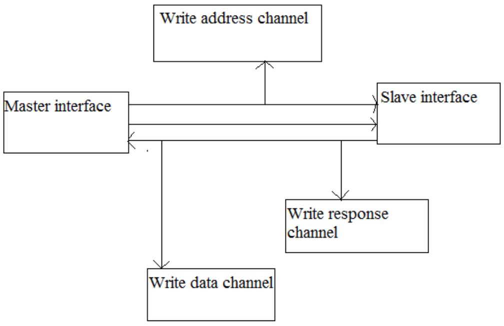

Fig -3: Write transaction

Every exchange is burst based which has address and control data on the location channel that depicts the way of the data is exchanged. The information is exchanged in the middle of master and slave utilizing a compose information channel to the slave or a read information channel to the expert [9]. The compose operation process begins when the expert sends a location and control data on the compose address direct as appeared in fig. 3. The expert then sends everything of write data information over the write information channel. The expert keeps the VALID flag low until the write data information is accessible. The expert sends the last information thing, the WLAST signal goes HIGH.

2.2 Direct memory access

Direct memory access (DMA) is a technique that permits an info/yield (I/O) gadget to send or get information straight forwardly to or from the fundamental memory, bypassing the CPU to accelerate memory operations. The procedure is overseen by a chip known as a DMA controller (DMAC). A PC's framework asset apparatuses are utilized for correspondence in the middle of equipment and programming. The four sorts of framework assets are: I/O addresses Memory addresses Intrude on solicitation numbers (IRQ) Direct memory access (DMA) channels DMA channels are utilized to convey information between the fringe gadget and the framework memory . Each of the four framework assets depend on specific lines on a transport. A few lines on the transport are utilized for IRQs, some for locations (the I/O addresses and the memory location) and some for DMA channels. A DM A channel empowers a gadget to exchange information without presenting the CPU to a work over-burden. Without the DMA channels, the CPU duplicates each bit of information utilizing a fringe transport from the I/O gadget. Utilizing a fringe transport possesses the CPU amid the read/compose prepare and does not permit other work to be performed until the operation is finished.

3. BURST BASED TRANSACTIONS

The AXI protocol defines three burst types described in [7]:

1. Fixed burst [10]
2. Incrementing burst [10]
3. Wrapping burst [10]

3.1 Fixed burst

In a fixed burst, the location continues as before for each move in the burst. This burst sort is for rehashed gets to the same location, for example, when stacking or discharging a fringe FIFO.

3.2 Incrementing burst

In an incrementing burst, the location for every move in the burst is an increment of the past exchange address [7]. The increment esteem relies on upon the measure of the exchange. For case, the location for every move in a burst with a size of four bytes is the past address in addition to four [7].

3.3 Wrapping burst

A wrapping burst is like an increasing burst, in that the location for every exchange in the burst is an augmentation of the past exchange address. Be that as it may, in a wrapping burst the location wraps around to a lower location when a wrap limit is come to. The wrap limit is the measure of every move in the burst duplicated by the aggregate number of moves in the burst. Two confinements apply to wrapping burst: The begin address must be adjusted to the measure of the exchange. The length of the burst must be 2, 4, 8, or 16.

4. WRITE TRANSACTION STATE MACHINE

The compose state machine comprises of 4 expresses that incorporate the reset, address hold up, information hold up and reaction hold up . The reset state is the underlying state. The state machine forsakes every other capacity and returns to same state if the reset sign is stated [5]. In the location hold up state, the slave anticipates the location and control data from the master for the compose exchange [5]. The information hold up state takes after the location handshake and proceeds with the information exchange for compose. In the reaction sit tight express the slave anticipates for the last information and afterward handshake by means of the reaction channel and affirms the information exchange with a reaction signal.

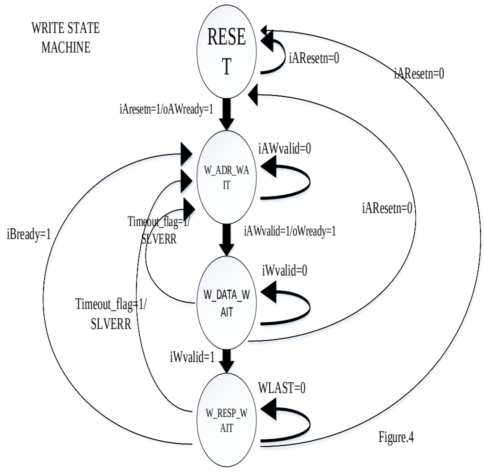

Fig -4: write state FSM

The read state machine comprises of 3 expresses that incorporate the reset, address hold up and the information hold up [5]. The reset state is the introductory state. The state machine surrenders every other capacity and returns to this state if the reset sign is declared. In the address hold up state, the slave anticipates the location and control data from the Master for the read exchange. The information hold up state takes after the location handshake and proceeds with the information exchange for the read. The read exchange does not include a different channel for reaction consequently it gives a reaction sign to each information exchange rather than the last information exchange over the read information channel itself.

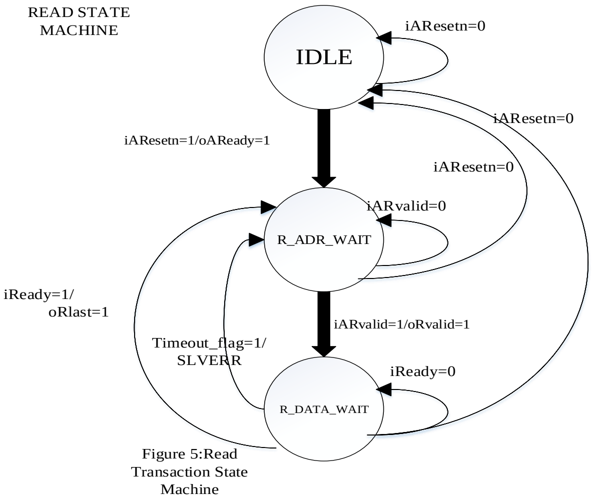

Fig -5: Read state FSM

5. RESULTS

S.NO		90nm	32nm
1	WITH CONSTRAINT		
	AREA (umm^2)	1306.341808	380.714173
	POWER (uW)	54.4585	9.6171
	TIMING (s)	93.22	98.46
2	WITHOUT CONSTRAINT		
	AREA (umm^2)	2514.388096	419.537704
	POWER (uW)	79.7765	9.8993
	TIMING (s)	0.01	0.10

We have processed the write operation alongside the idleness figuring in 90nm and 32 nm technology, where area is lessened by 1208.08umm^2 (48.04%) and power is diminished by 25.32uW (31.74%) in 90nm and in 32nm technology, area gets decreased by 38.82umm^2 (9.25%) and power by 0.28uW (2.83%). The constraints we used for the accomplishment to minimize the latency is gated clock.

6. CONCLUSION

It is a part of the Advanced Microcontroller Bus Architecture (AMBA) created by ARM (Advanced RISC Machines) organization [5][10]. It is an On-Chip correspondence convention. The AMBA AXI convention underpins superior, high-recurrence framework plans. The AXI convention is suitable for high-data transmission and low-inertness plans. It gives high-recurrence operation without utilizing complex extension. It meets the interface prerequisites of an extensive variety of parts. AXI convention is appropriate for memory controllers those are having lofty starting access inactivity [5][9]. It gives adaptability in the usage of interconnect designs.

ACKNOWLEDGEMENT

We acknowledge the help of Prof. S Ravi who guided us helped a lot to get idea and methodologies to do this project. We are also thanking our program chair Prof. Sivasnskaran. We are indebted to our lab assistant Prof. Karthikeyan and all other faculties of VLSI System department.

REFERENCES

[1] Golla Mahesh, Sakthi vel.SM, “Functional Verification of the Axi20cp Bridge using System Verilog and effective bus utilization calculation for AMBA AXI 3.0 Protocol”, IEEE Sponsored 2nd International Conference on Innovations in Information, Embedded and Communication systems (ICfIECS), 2015

[2] Chien-Hung Chen, Jiun-Cheng Ju, and Ing-Jer Huang, “A Synthesizable AXI Protocol Checker for SoC Integration”, International SoC Design Conference (ISOCC), 2011.

[3] Xiongfei Liao, Jun Zhou and Xin Liu, “Exploring AMBA AXI On-Chip Interconnection for TSV-based 3D SoCs”, IEEE International 3D Systems Integration Conference (3DIC), 2012.

[4] Bruce Mathewson, “The Evolution of SOC Interconnect and How NOC Fits Within It”, Design Automation Conference (DAC), 47th ACM/IEEE, 2010.

[5] Mahesh Pai, H Sudha, Lakshmikantha, “Design and Implementation of AMBA based AXI 4 Slave Interface”, International Journal of Innovative Research in Science, Engineering and Technology, 2015.

[6] Dr. K.C. Mahajan, Mukesh K Yadav, “Review of System – On- Chip Advanced extensible Interface (AXI) Bus Protocol”, International Journal of Digital Application & Contemporary Research, 2015.

[7] Smitha A, Ashwini, “uvm based test bench to verify amba axi4 slave protocol”, International Research Journal of Engineering and Technology (IRJET), 2016.

[8] N. Sudhakar, M. Kiran Kumar, “Interfacing between High Performance Drivers and Low Power Devices using ABP Bridge”, International Journal of Science and Research (IJSR), 2013.

[9] Georgiy Shederovich, “SOC open bus standards overview”, 2007.

[10] Harsha Garua1 , Keshav Sharma2 , Chusen Duari, “verification of amba axi bus protocol implementing incr and wrap burst using system verilog”, ijret: International Journal of Research in Engineering and Technology, 2016.

[11] Manoj Gupta, Ashok Kumar Nagawat, “Design and Implementation of High Performance Advanced Extensible Interface (AXI) Based DDR3 Memory Controller”, International Conference on Communication and Signal Processing, 2016.

[12] Rini Sebastian, Silpa Rose Mary, Gayathri M, Anoop Thomas, “Assertion Based Verification of SGMII IP core incorporating AXI Transaction Verification Model”, International Conference on Control, Communication & Computing India (ICCC), 2015.

[13] G.Mahesh, Sakthivel. SM, “Verification of Memory Transactions in AXI Protocol using System Verilog Approach”, International Conference on Communications and Signal Processing (ICCSP), 2015.

[14] Chadi Al khatib, Claire Aupetit, Cyril Chevalier, Chouki Aktouf, Gilles Sicard, Laurent Fesquet, “A Generic Clock Controller for Low Power Systems: Experimentation on an AXI Bus”, IFIP/IEEE International Conference on Very Large Scale Integration (VLSI-SoC), 2015.

[15] Anitha.H.T, Dr.Nataraj.K.R, “implementation of multi-slave interface for axi bus”, Fourth International Conference on Advances in Recent Technologies in Communication and Computing (ARTCom), 2015.

[16] Marco Ramirez, Masoud Daneshtalab, Juha Plosila, Pasi Liljeberg, “NoC-AXI Interface for FPGA-based MPSoC Platforms”, 22nd International Conference on Field Programmable Logic and Applications (FPL), 2012.

[17] Tao Lv1, Tong Xu, Yang Zhao, Huawei Li, Xiaowei Li, “Bug Analysis and Corresponding Error Models in Real Designs”, IEEE International High Level Design Validation and Test Workshop, 2007.

[18] Jun Zheng, Kang Sun, Xuezeng Pan, and Lingdi Ping, “Design of a Dynamic Memory Access Scheduler”, 7th International Conference on ASIC, 2008.

[19] Sudeep Pasrichat, Nikil Dutt', Mohamed Ben-Romdhanet, “Extending the Transaction Level Modeling Approach for Fast Communication Architecture Exploration”, Design Automation Conference, Proceedings, 2005.