International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395 -0056
Volume: 03 Issue: 08 | Aug-2016 www.irjet.net p-ISSN: 2395-0072

Design for Core link Network Interconnect NIC 400 Architecture

Divya C[1], Mr. Pradeep S V[2]

[1] *PG Scholar, Digital Electronics, VTU Extension Centre, UTL technologies LTD, Bangalore*
[2] *Assistant Professor, Digital Electronics, VTU Extension Centre, UTL technologies LTD, Bangalore*

Abstract - ***This paper work is entirely focused on ARM Core Link NIC-400 Network Interconnect is an extremely versatile piece of IP'S. The generation of the NIC 400 with Advanced Extensible Interface by AMBA Designer software for development of specific implementation, presents the Verilog RTL, test bench and stimulation scripts.The number of ports, number of nets, number of cells, and total area of NIC 400 also the frequency speed has been achieved using Design Compiler. Mentor Questa sim is used as a simulator tool to simulate the read and write transactions.***

***Key Words*: NIC 400, AXI, AMBA Designer etc.**

1. INTRODUCTION

The NIC-400 is the 4th generation AXI interconnect from ARM and is delivered as a base product of AMBA AXI3 and/or AXI4 interconnect. Generation of the NIC-400 micro-architecture configurations Integrators no longer need to understand all the nuances of the NIC-400 micro-architecture Core Link Creator enables all the flexibility of high configurable design without the downsides of complexity and knowledge. The design and configurability of the NIC-400 allows the user to implement the highest performance interconnects for their set of master and slave requirements while minimizing area and power. The NIC-400 network of switches allows scaling up to very large numbers of masters and slaves while maintaining high maximum operating frequencies. As is common in most SoCs, the NIC-400 connects up IP blocks with a range of different AMBA interfaces with multiple data width and variable clock frequencies.

2. ADVANCED EXTENSIBLE INTERFACE

As AXI provides many features such as out of order completion, interleaving; interconnect is responsible to take care of interleaving and out of order. The block level RTL code is automatically configured from a system description file to specify no of master, slave, width of address bus hence interconnect is implemented depending on the application requirements. The AXI was burst based. Every transaction has addressed and control information on the address channel that describes the data.

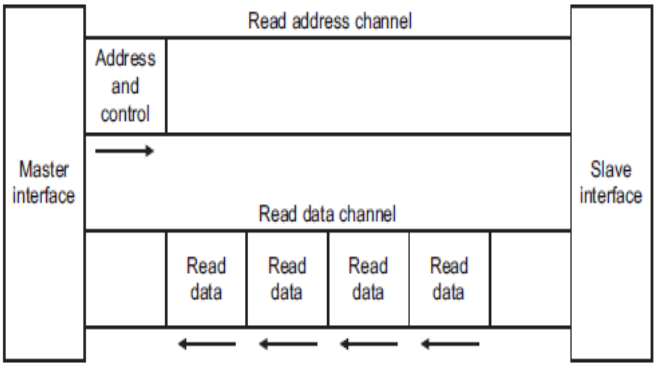

Figure 2.1: AXI channel architecture of reads.

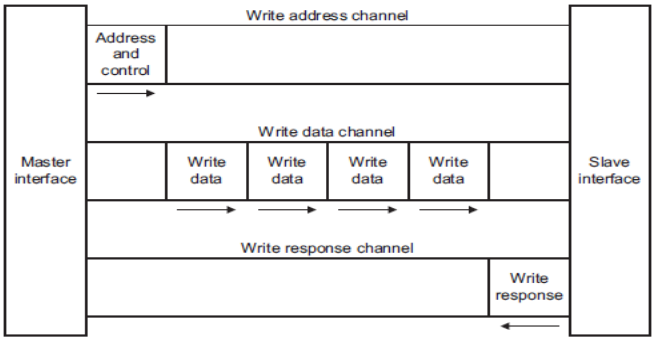

Figure 2.2: AXI channel architecture of writes.

2.1 READ AND WRITE ADDRESS CHANNELS

Read and write transactions each have their own address channel. The appropriate address channel carries all of the required address and control information for a transaction. Each AXI channel transfers information in only one direction, and there is no requirement for a fixed relationships between various channels. This is important because it enables the insertion of a register slice in any channel, at the cost of additional cycle of latency. The AXI protocol supports the following mechanisms: variable-length bursts, from 1 to 16 data transfers per burst, bursts with a transfer size of 8-1024 bit, wrapping, incrementing, and non-incrementing bursts, atomic operations, using exclusive or locked accesses, system-level caching and buffering control.

3. AMBA DESIGNER

Configuring and stitching together complex IP components manually is a time consuming process, which in turn introduces further complexity when ensuring the compatibility of interface parameters such as write interleaving and acceptance depths, bus widths, ID widths etc. The AMBA Designer (ADR-400) tool provides a single common front end for configuring and integrating System IP and other ARM IP. The main benefits from this approach comprise rapid, error free configuration using intelligent configuration tools, and correct by construction connection of ports using the IP-XACT interface standard. ADR-400 outputs configured Verilog RTL along with industry standard IP-XACT descriptions for ease of integration into third party design and implementation tools. Once a component is generated it can also be added to the ADR-400 component library to maximize re-use. Stitching together of multiple components is supported in a hierarchical manner.

4. BLOCK DIAGRAM AND IMPLEMENTATION

The NIC-400 network of switches allows scaling up to very large numbers of masters and slaves while maintaining high maximum operating frequencies

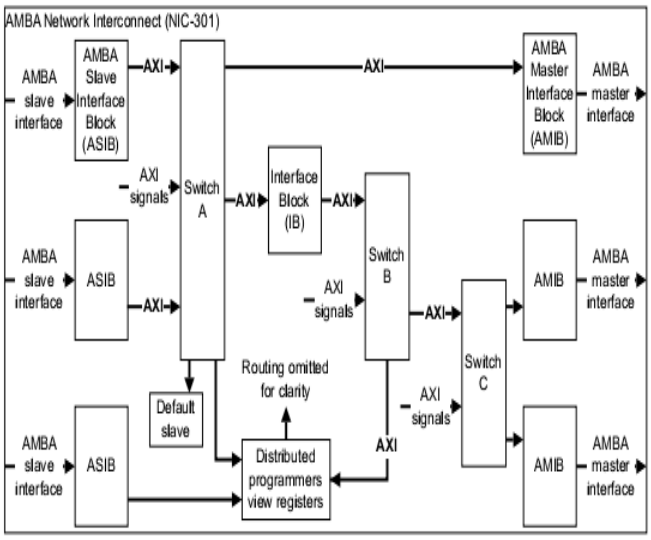

Figure 4.1: NIC 400 top level Interconnection.

The Core Link NIC-400 Network Interconnect is highly configurable and enables you to create a complete high performance, optimized and AMBA-compliant network infrastructure. The possible configurations for the Core Link NIC-400 Network Interconnect can range from a single bridge component, for example an AHB to AXI protocol Conversion Bridge, to a complex interconnect that consists of up to 128 masters and 64 slaves of AMBA protocols. Switches take the functions of the repeater and the bridge and combine them in clever ways to create a multi-port interconnect box that provides wonderful interconnectivity but challenges network protocol analyzer engineers. And let's complicate things by, essentially, merging the bridging and routing functions into a single box from Cisco, Bay Networks, or other vendors.

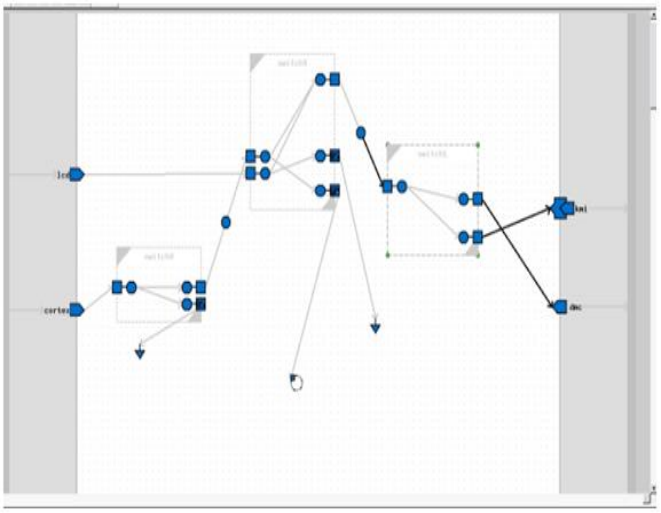

Figure 4.2: NIC 400 Implementation view.

The implementation toolbar enables you to create and modify your implementation. This view also enables you to specify implementation-specific values for the architecture you have chosen in the Architecture View, for example Lock Support. The main view is the Topology window that consists of the abstract layout of the architecture broken down into switches, master interface blocks, slave interface blocks, and interface blocks.

Implement the design that you specified with the Architecture View and Address View. This view uses the same system as the Overlays window in the Architecture View. The AMBA Designer GUI infers the bridging requirements of connected masters, connected slaves, and the connections between switches from: the clock domain of the switch, clock domains that you entered in the Architecture View, data width information from the Architecture View, data width information that you define in the Parameter window of the Implementation View.

5. RESULTS

The read and write transactions simulation results of NIC 400 core link network interconnect was simulated using AMBA design software are as shown in the below figures.

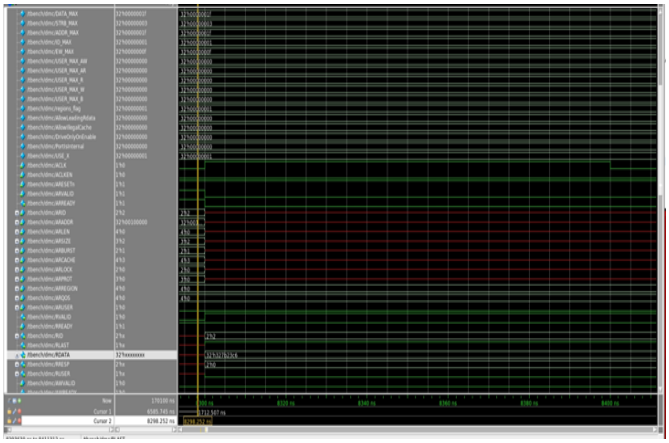

Figure 5.1: Read transaction from DMC to CORTEX.

The above figure shows that master also drives a set of control signals showing the length and type of the burst, but these signals are omitted from the figure for clarity. After the address appears on the address bus, the data transfer occurs on the read data channel. The slave keeps the **VALID** signal LOW until the read data is available. For the final data transfer of the burst, the slave asserts the **RLAST** signal to show that the last data item is being transferred.

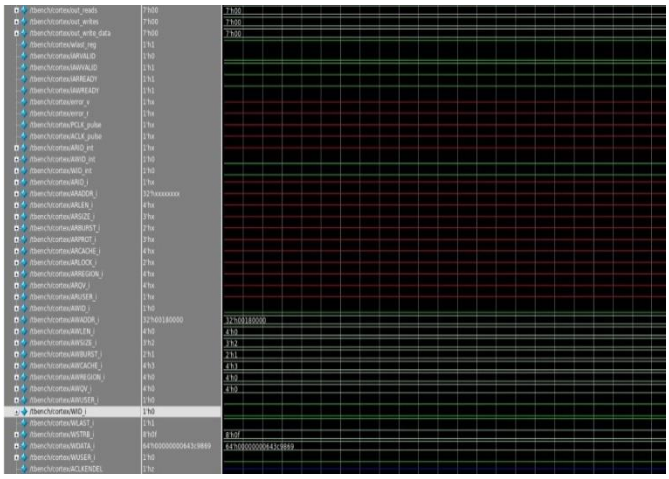

Figure 5.2: Write transaction from CORTEX to DMC.

The above figure shows that when the master sends an address and control information on the write address channel. The master then sends each item of write data over the write data channel. When the master sends the last data item, the **WLAST** signal goes HIGH. When the slave has accepted all the data items, it drives a write response back to the master to indicate that the write transaction is complete.

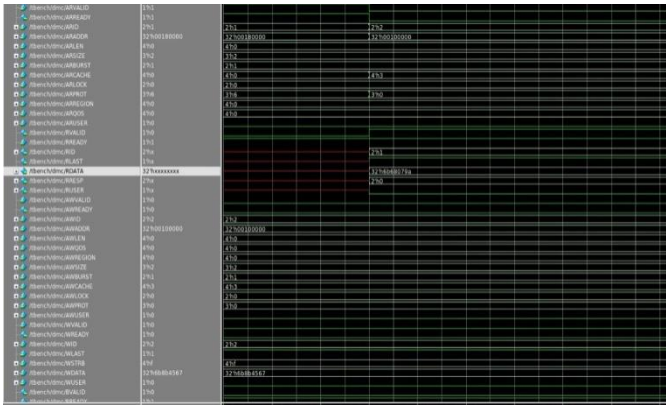

Figure 5.3: Read transaction from DMC to LCD.

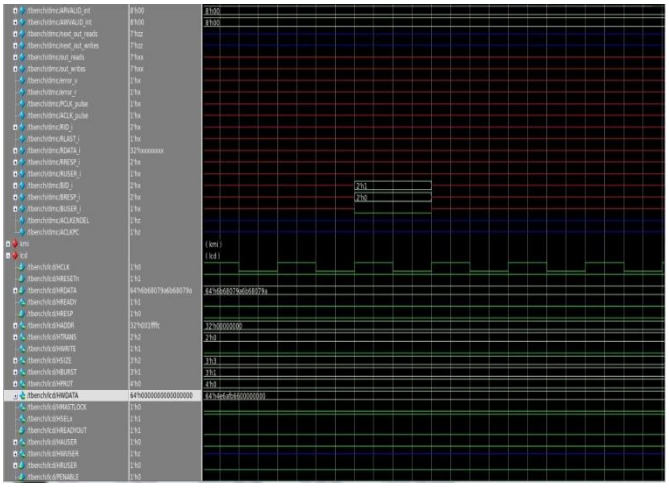

Figure 5.4: Write transaction from LCD to DMC.

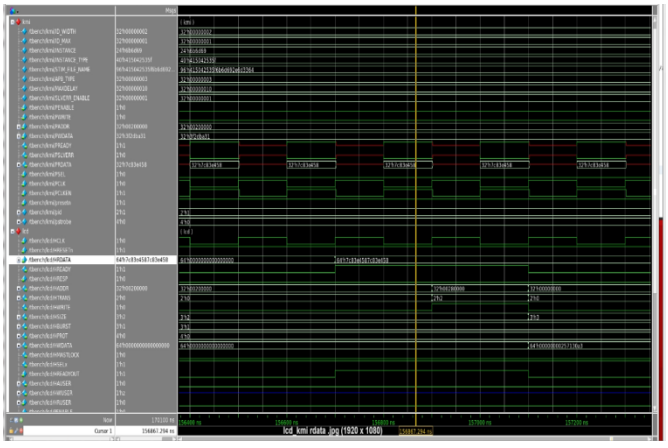

Figure 5.5: Read transaction from KMI TO LCD.

6. CONCLUSION

Read and write transaction simulations of two masters and two slaves were simulated. Core link network interconnect has a frequency speed of 533MHZ, cells required for the total area, switching power, dynamic power has calculated by the design compiler as a synthesis tool.

REFERENCES

[1]. Ms. AnushaRanga, Mr. L. HariVenkatesh, Mr. Venkanna, "Design and Implementation of AMBA-AXI Protocol using VHDL for SOC Integration," in International Journal of Engineering Research and Applications, Vol. 2, Issue4, July-August 2012, pp.1102-1106.

[2]. Ref Shaila S Math, Manjula R B "Survey of system on Chip buses based on industry standards," Conference on Evolutionary Trends in Information Technology (CETIT), Belgaum, Karnataka, India, pp. 52, May 2011.

[3]. William Orme, ARM BillNeifert, Carbon Design Systems ATC-300,"Getting the Most Out of the ARM® Core Link™ NIC-400" October 2013.

[4]. International Journal of Engineering Research and General Science, AXI Interconnect Between Four Master and Four Slave Interfaces Volume 2, Issue 4, June-July, 2014.

[5]. AMBA AXI Protocol Version: 2.0 Specification, ARM Ltd, pp. 1-1.

[6]. ARM, AMBA AXI protocol specifications, Available at, http://www.arm.com, 2003.

[7]. Umit Y. Ogras, Jingcao Hu, RaduMarculescu Department of Electrical Computer Engineering Carnegie Mellon University Pittsburgh, PA 15213-3890, USA.

[8]. Silicore Corporation, Wishbone system-on-chip (soc) interconnection Architecture for portable IP cores.

[9]. M SivaPrasad Reddy, B. Babu Rajesh, TvsGowtham Prasad, "A Synthesizable Design of AMBA-AXI Protocol for SOC Integration," in International Journal of Engineering Inventions Volume 1, Issue 3 (September2012) PP: 19-26.

[10]. ARM, AMBA AXI Protocol Specification (Rev 2.0), Available at http://www.arm.com, March 2010.

[11]. ARM, AMBA AXI Protocol Specification (Rev 2.0), Available at http://www.arm.com, March 2010.

[12]. ARM, AMBA Specifications (Rev2.0). [Online]. Available at http://www.arm.com,1999.