

AMBA AXI BUS TO NETWORK-ON-CHIP BRIDGE

NG KENG YOKE

UNIVERSITI TEKNOLOGI MALAYSIA

AMBA AXI BUS TO NETWORK-ON-CHIP BRIDGE

NG KENG YOKE

A project report submitted in partial fulfilment of the
requirements for the award of the degree of
Master of Engineering (Electrical - Computer and Microelectronic System)

Faculty of Electrical Engineering
Universiti Teknologi Malaysia

JUNE 2013

"To my wife who provides all the support"

ACKNOWLEDGEMENT

I would like to express my deepest gratitude especially to my lecturer, Dr. Muhammad Nadzir Marsono, for his wise and continuous guidance, support and encouragement in this project. I am grateful to be able to complete this project with the objectives on time through his supervision. Furthermore, I would like to express my heartfelt thanks to my family, friends, colleagues and fellow coursemates for their constant support throughout the project.

Ng Keng Yoke
Bayan Baru, Penang - Malaysia

ABSTRACT

Bus architectures are a necessity for today's System-On-Chip (SoC) design. Current SoC design is getting more complex with additional features and functions. The bus architecture arbitration need to handle requests from multiple cores where this will ultimately becomes a bottleneck to the bus architecture performances. Most Intellectual Property (IP) designs today use bus protocol such as Advanced Microcontroller Bus Architecture (AMBA) Advanced High-performance Bus (AHB) and are facing such limitations. The ability for an IP core to be reusable in Network-on-Chip (NoC) based SoCs is highly desirable. The solution is to implement the AMBA Advanced eXtensible Interface (AXI) to NoC bridge which emulates the bus protocol and convert it to NoC protocol and vice versa, enabling quick migration of IPs cores designed for a traditional bus architecture to the NoC architecture. In this work, a bus-to-NoC bridge has been designed. The bus-to-NoC bridge converts the AMBA AXI bus protocol to NoC protocol and sends through NoC interface, achieving performance gain comparable to the traditional AMBA bus architectures. The advantages of bus-to-NoC bridge architecture includes 1. Two times performance gain in terms of latency and throughput compared to traditional bus architectures. 2. Supports various AXI command signals such as protection unit supports information signals, Atomic operations signals, error response encoding for AXI and ordering rules signals. 3. The ability to support burst for memory access. This enables the migration of bus architectures to NoC architectures, which will likely be the future design trend.

ABSTRAK

Bus arkitek memainkan peranan penting dalam dunia System-On-Chip (SoC) pada masa kini. Seni bina SoC menjadi semakin kompleks disebabkan oleh bertambahnya fungsi-fungsi baru. Dengan ini arkitek bus tradisi kena mengawal permintaan daripada pelbagai ejen-ejen dan ini telah menyebabkan pretasi bus tradisi tersekat. Kebanyakan seni bina IP hari ini menggunakan protokol bus seperti "Advanced Microcontroller Bus Architecture" (AMBA) Advanced High-performance Bus (AHB) mengalami limitasi ini. Kebolehan IP untuk diguna semula di Network-on-Chip (NoC) berdasarkan SoC adalah sangat diingini. Penyelesaiannya adalah melaksanakan satu seni bina jambatan IP bus iaitu AMBA "Advanced eXtensible Interface" (AXI) kepada NoC. Ia mengemulasikan protokol bus dan menukarkannya kepada protokol NoC. Ini akan membolehkan imigrasi dari IP bus tradisi kepada NoC arkitek dalam masa yang singkat. Projek ini, satu bus kepada NoC telah direka. Bus kepada NoC seni bina ini menukarkan protokol bus kepada NoC paket dan menghataraken kepada NoC, and prestasi NoC ini adalah lebih baik berbanding dengan seni bina bus tradisi. Kelebihan seni bina arkitek bus-to-NoC ini termasuk

1. Pretasi adalah dua kali ganda lebih baik berbanding dengan seni bina bus tradisi dari segi pemprosesan dan "latency".
2. Menyokong pelbagai isyarat seperti isyarat-isyarat perlindungan, isyarat-isyarat Atomic, isyarat-isyarat ralat untuk AXI dan isyarat-isyarat "ordering".

Kelebihan-kelebihan ini membolehkan seni bina bus tradisi menukar kepada seni bina NoC, memandangkan ia akan menjadi seni bina untuk masa depan.

TABLE OF CONTENTS

CHAPTER	TITLE	PAGE
	DECLARATION	ii
	DEDICATION	iii
	ACKNOWLEDGEMENT	iv
	ABSTRACT	v
	ABSTRAK	vi
	TABLE OF CONTENTS	vii
	LIST OF TABLES	x
	LIST OF FIGURES	xi
	LIST OF APPENDICES	xiv
1	INTRODUCTION	1
	1.1 Network-on-Chip: A Scalable On-chip Interconnect	1
	1.2 Problem Statement	1
	1.3 Objectives	3
	1.4 Scope of Work	3
	1.5 Methodology	4
	1.6 Report Organization	4
2	LITERATURE REVIEW	6
	2.1 Bus Protocol	6
	2.1.1 AMBA	6
	2.1.2 AMBA protocol specifications	7
	2.1.3 AMBA AXI 3	7
	2.1.4 AXI Features	7
	2.1.5 Read Burst	9
	2.1.6 Write Burst	9
	2.2 Related Works	10
	2.2.1 Bus Emulation in Nostrum NoC	10

	2.2.2	Low-Power with Error-Correcting Implementation	12
	2.2.3	A Low Latency NOC Router Supporting Routing Adaptivity	14
	2.3	Chapter Summary	18
3		IP BUS-TO-NOC BRIDGE	19
	3.1	High Level Architecture of the Bridge Design	19
	3.2	Bus Master Protocol	20
	3.3	Network Interface	21
	3.3.1	X-Y Routing Mechanism	22
	3.3.2	Network Packet Structure	22
	3.3.3	Enhancement to Network Packet Structure	23
	3.3.4	Network Waveform	25
	3.4	Bridge Design Implementation Architecture	25
	3.4.1	AMBA AXI Bus-to-NoC Bridge Master	26
	3.4.2	AMBA AXI Bus-to-NoC Bridge Slave	32
	3.5	Enhancements from Previous NoC Design	39
	3.5.1	NoC Data Bus Expanded	39
	3.5.2	NoC Packet Structure Enhancement	40
	3.6	Chapter Summary	40
4		EMULATION ARCHITECTURE	41
	4.1	Bus-to-NoC Bridge Emulation	41
	4.2	Bus Functional Module (BFM) for AMBA AXI	41
	4.2.1	Bridge Master BFM	41
	4.2.2	Bridge Slave BFM	42
	4.3	Chapter Summary	44
5		RESULTS AND ANALYSIS	46
	5.1	AMBA AXI to NoC Bridge Design Results	46
	5.1.1	Bridge Master	46
	5.1.2	Bridge Slave	48
	5.2	Environmental Setup	49
	5.3	Latency	50
	5.4	Chapter Summary	56

6	CONCLUSION AND FUTURE WORKS	58
6.1	Contribution	58
6.2	Future Works	58
	REFERENCES	60
	Appendices A – B	62 – 72

LIST OF TABLES

TABLE NO.	TITLE	PAGE
1.1	Arteris™. A comparison of Network-on-Chip and Busses.	2
3.1	Write Signals in FIFO queue	28
3.2	Read Signals in FIFO queue	29
5.1	Routing table for case study 1	51
5.2	Access routing table for case study 2	53

LIST OF FIGURES

FIGURE NO.	TITLE	PAGE
1.1	Multiplexor Connection	2
2.1	Read Transaction [1]	8
2.2	Write Transaction [1]	8
2.3	Read Burst [1]	9
2.4	Write Burst for 4 flits of data transfer [1]	10
2.5	Bus emulation with 7x7 NoC	11
2.6	A wrapper is included with AMBA IP core inside a tile in the on-chip network	12
2.7	Architecture of the wrapper	13
2.8	Four bit Hamming code encoder	13
2.9	Bus-invert design	14
2.10	Bus-invert design at Depacketization Module	14
2.11	Hamming code decoder	15
2.12	Functional block diagram of a NoC router	16
2.13	Immediate Grant/Ack Mechanism	16
2.14	Switching Block Logic Diagram	17
2.15	Switching Control Block Logic Diagram: Multiple Request Target to Yneg/South Destination	17
3.1	Bus-to-NoC high level bridge master Interface	20
3.2	Bus-to-NoC high level bridge slave Interface	20
3.3	4x4 Mesh Network Architecture	23
3.4	Packet Structure [2]	24
3.5	Enhanced Packet Structure	24
3.6	Example of flit Structure	24
3.7	Immediate Grant	25
3.8	4 grant per clock	25
3.9	Bridge Master design micro-architecture	26
3.10	Bridge Master design micro-architecture	28
3.11	Bridge Master Write FIFO queue	28
3.12	Bridge Master Read FIFO queue	29

3.13	Bridge Master Clock Crossing Block	30
3.14	Address to NoC destination Look Up Table	31
3.15	Bridge Master Requets Control FSM	31
3.16	MRESPSM (right) and Bridge Master Response block diagram	32
3.17	Bridge Master Response FIFO Queue	33
3.18	Bridge Slave design micro-architecture	34
3.19	Bridge Slave Decode block	35
3.20	Bridge Slave Write FIFO queue	35
3.21	Bridge Slave Read FIFO queue	36
3.22	Bridge Slave Clock Cross block	37
3.23	Bridge Slave Read Control Block(left) with Control State Machine(right)	37
3.24	Bridge Slave Write Control Block(left) with Control State Machine (right)	38
3.25	Bridge Slave Response Block	38
3.26	Bridge Slave Response FIFO queue	39
4.1	Experimental Setup for bus-to-NoC Emulation wrapper design	42
4.2	Bridge Master BFM: AXI Bus Master Write request FSM	43
4.3	Bridge Master BFM: AXI Bus Master Read request FSM	43
4.4	Bridge Slave BFM: AXI Bus Slave Write FSM	44
4.5	Bridge Slave BFM: AXI Bus Slave Read FSM	45
5.1	Bridge master write request	47
5.2	Bridge master read request	47
5.3	Bridge master response waveform	47
5.4	Bridge slave write request	48
5.5	Bridge slave read request	48
5.6	Bridge slave response waveform	49
5.7	Bridge Master and Bridge Slave Setup in 4x4 Mesh NoC	49
5.8	Requests routing for case study 1	50
5.9	The latency for 50 write requests	51
5.10	The latency for 50 read requests	51
5.11	The latency for 50 response	52
5.12	The throughput for 50 write requests	53
5.13	The throughput for 50 response	53
5.14	Requests routing for case study 2	54
5.15	The latency for 50 write requests	54

5.16	The latency for 50 read requests	55
5.17	The latency for 50 response	55
5.18	The throughput for 50 write requests	56
5.19	The throughput for 50 response	57

LIST OF APPENDICES

APPENDIX	TITLE	PAGE
A	SOURCE CODE FOR BRIDGE MASTER	62
B	SOURCE CODE FOR BRIDGE SLAVE	72

CHAPTER 1

INTRODUCTION

Most of current system-level digital designs are developed based on the concept of bus architectures. Bus architectures have successfully been implemented in virtually all complex System-On-Chips (SoCs) [3]. Traditional bus architectures such as Advanced Microcontroller Bus Architecture (AMBA) Advanced High-performance Bus (AHB) bus specification is designed to be used with central multiplexor based interconnections method, shown in Figure 1.1. In this method, more than one bus masters can drive out the address and control signals indicating the transfer that they wish to perform. As current SoC design is getting more complex, the bus architecture needs to handle requests from multiple agents where this will ultimately become a bottleneck to its performances. In such circumstances, Network-On-Chip (NoC) [4] architectures appear to be a more attractive solution for future digital designs.

1.1 Network-on-Chip: A Scalable On-chip Interconnect

Various research studies [5] have shown that the feasibility and advantages of NoC over traditional bus architectures. Table 1.1 shows the NoC developed by Arteris, has significant advantages over traditional bus architecture in terms of maximum frequency, peak throughput and also system throughput [5].

1.2 Problem Statement

Even though NoC will likely to become the future trend of digital design architecture, almost all the industrial-driven SoC designs today are standard bus architectures such as ARM architectures (AMBA AHB), Open Core Protocol (OCP)

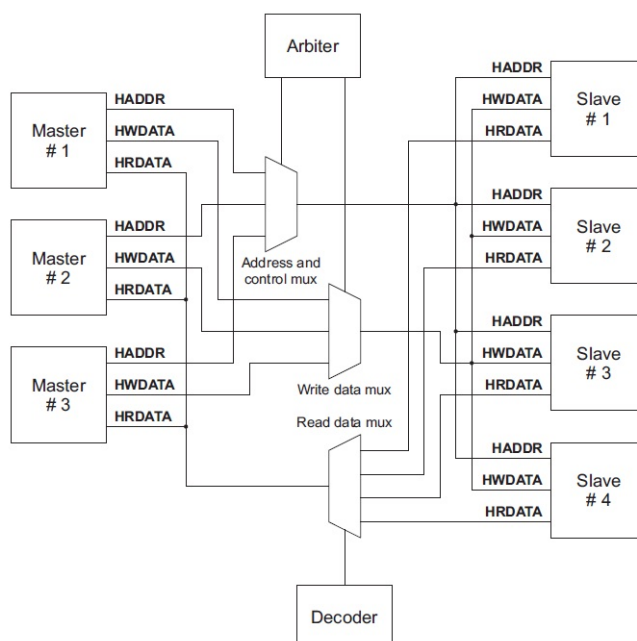


Figure 1.1: Multiplexor Connection

Table 1.1: Arteris™. A comparison of Network-on-Chip and Busses.

Criteria	Bus	NoC
Max Frequency	250 MHz	> 750 MHz
Peak Throughput	9 GB/s <i>(more if wider bus)</i>	100 GB/s
Cluster min latency	6 Cycles @250MHz	6 Cycles @250MHz
Inter-cluster min latency	14-18 Cycles @250MHz	12 Cycles @250MHz
System Throughput	5 GB/s <i>(more if wider bus)</i>	100 GB/s
Average arbitration latency	42 Cycles @250MHz	2 Cycles @250MHz
Gate count	400K	210K
Dynamic Power	Smaller for NoC, see discussion in 3.5.2	
Static Power	Smaller for NoC (proportional to gate count)	

and Altera Avalon, to name a few. As the industry is driving towards shorter time to market for a product, the ability for an Intellectual Property (IP) cores to be reusable in NoC-based SoCs is highly desirable. The solution is to implement a design which emulates the bus protocol and convert it to NoC protocol and vice versa, enabling quick migration of IPs cores from traditional bus architecture to NoC architecture while still benefit from the advantages of NoC architectures.

1.3 Objectives

Based on the existing bus-to-NoC architectures, the objective of this project is to achieve the following improvements:

1. Develop a bus-to-NoC bridge design which emulates the bus protocol for connection to NoC routers. The bus protocol for this project is AMBA 3 Advanced eXtensible Interface (AXI) [1].
2. Analyze the performances of the bus-to-NoC bridge emulation compared to the AMBA AXI bus architecture. The performance for writes, reads and responses through the bridge will be analyzed.

The intention for this analysis is to ultimately prove that bus-to-NoC bridge design with NoC that is able to have a better performance than traditional bus architecture for certain configurations.

1.4 Scope of Work

This project's aim is to develop a new bus-to-NoC bridge design based on the objectives stated above. The micro-architecture level for the design is defined for the bus-to-NoC bridging. Functional blocks are created to partition out the functions that are required to be done by each sub-blocks and is implemented and tested in Verilog [6] environment. Hardware implementation on FPGA is out of the scope for this project. The design is intended for ASIC implementation, in this project it is limited to architectural Proof of Concept (PoC) using VCS software from Synopsys [7].

1.5 Methodology

A bridge's task is to take in the bus master's outputs and convert the requests to NoC protocol packets before passing them to a NoC router. Subtasks that need to be done include converting bus protocol to NoC protocol and, packing the request and decoding the destination on the NoC interface. The packet will be sent to the destination router through the NoC interface. The bridge receives the NoC packet from the destination router and converts it from network packet protocol back to bus protocol before sending it to the bus slave. The bridge performs bus emulation work where the NoC interface remains transparent to the bus master and slave.

In this project, Verilog testbenches are used to set up the eco-system which consists of bus, NoC router interface and bridge designs. The bus-functional-modules, which behave like bus masters, are developed and used to initiate and send requests to the bus-to-NoC bridge. The BFM bus slaves are also developed, for providing responses to bus master's requests and receiving and handling requests that are sent from the bridge.

Synopsys VCS tool is used to set up the design environment and the whole testbench. The NoC routers from previous works [2] are instantiated as part of the design environment.

1.6 Report Organization

This report is organized as follows.

Chapter 2 discusses about the existing works that have been done on bus-to-NoC emulation designs.

Chapter 3 discusses about the proposed wrapper architecture and implementation details of the design, and explains the implementation choice taken.

Chapter 4 discusses the experimental setup and general framework of the overall design, and tools that are used to implement the design.

Chapter 5 shows the implementation results and analysis of the design. Performance analysis is also done on the bus-to-NOC design and is compared to normal bus architecture.

Chapter 6 derives the conclusion for this IP Wrapper for bus-to-NoC Emulation project based on the results obtained.

REFERENCES

1. ARM Limited. AMBA AXI Protocol. 2004.
2. Gwee, Y. C. *A Low Latency NoC Router Supporting Routing Adaptivity*. Master's Thesis. Master in Microelectronics and Computer System, Faculty of Electrical Engineering, Universiti Teknologi Malaysia. 2011.
3. Wikipedia. System on a chip. http://en.wikipedia.org/wiki/System_on_chip.
4. Wikipedia. Network On Chip. http://en.wikipedia.org/wiki/Network_on_chip.
5. Arteris. A comparison of Network-on-Chip and Busses. *Communications, Circuits and Systems Proceedings, 2006 International Conference on*. Arteris. 2005, vol. 1. 1–10.
6. Association, I. S. *et al.* IEEE Standard for SystemVerilog-Unified Hardware Design, Specification, and Verification Language, 2005.
7. Synopsys. VCS. <http://www.synopsys.com/home.aspx>.
8. Specification, O. C. P. and I, V. Release 2.0, 2003.
9. ARM Limited. AMBA Specification. Rev 2.0. 1999.
10. Opencores, S. *Wishbone system-on-chip (soc) interconnection architecture for portable ip cores*. Technical report. Opencores. 2002.
11. Wikipedia. Advanced Microcontroller Bus Architecture. http://en.wikipedia.org/wiki/Advanced_Microcontroller_Bus_Architecture.
12. Andrzejewski, M. *AMBA bus emulation in the Nostrum NoC using best effort communication*. Master's Thesis. Master thesis, School for Information and Communication Technology, Royal Institute of Technology, Stockholm, Sweden. 2005.
13. Wu, C., Chi, H. and Huang, Y. A Wrapper for Low-Power Error-Correcting Data Delivery in On-Chip Networks. *Communications, Circuits and Systems Proceedings, 2006 International Conference on*. IEEE. 2006, vol. 4. 2662–2666.
14. Fletcher, R. J. Integrated circuit having outputs configured for reduced state

changes, 1987. US Patent 4,667,337.

15. Stan, M. R. and Burleson, W. P. Bus-invert coding for low-power I/O. *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, 1995. 3(1): 49–58.
16. Bolotin, E., Cidon, I., Ginosar, R. and Kolodny, A. QNoC: QoS architecture and design process for network on chip. *Journal of Systems Architecture*, 2004. 50(2): 105–128.