ARBITRATION SCHEMES OF WISHBONE ON-CHIP BUS SYSTEM

ONG KOK TONG

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

Faculty of Electrical Engineering
Universiti Teknologi Malaysia

JUNE 2014

ACKNOLEDGEMENTS

I would like to take this opportunity to express my appreciation to those people, researchers, and academicians who had provided a lot of advices and guidance to make this project a successful one. Firstly, I would like to express my gratitude and regards to my supervisor, Assos. Prof. Dr. Muhammad Mun'im Ahmad Zabidi for his guidance, support, and advice throughout the entire project.

Besides, I also take this opportunity to express my gratitude to Company Mentor, Mr. Chew Beng Wah for his support, understanding, and advice given to me, which help me to complete this project.

Lastly, I would like to thank my parents, colleagues, friends, and others who have given me constant encouragement that help me to overcome obstacles throughout the entire project.

ABSTRACT

In the SoC development, the compatibility of IP cores is one of the challenges that need to be addressed carefully. Most of the time, IP cores is having different input output specifications with new platform. The Wishbone SoC interconnection Architecture is aim to provide a good solution for SoC integration issues by having common interface specifications. In this project, the Wishbone on-chip computer bus for 32-bit cores is implemented in system verilog along with three different arbitration schemes which are fixed priority, round robin, and priority control. On top of that, the optimum transfer size for Wishbone bus in terms of bus throughput and average wait cycle is presented as well. It is found that the optimum transfer size for Wishbone bus is 64 bytes. Finally, the Wishbone bus is used to examine the bus performance of different arbitration schemes in Modelsim simulation. Round robin arbitration scheme is the best among three arbitration schemes in terms of bus throughput, logic complexity, and maximum wait cycle.

ABSTRAK

Dalam perkembangan SoC, keserasian teras IP adalah salah satu cabaran yang perlu ditangani dengan berhati-hati. Kebanyakan masa, teras IP mempunyai spesifikasi output input yang berbeza dengan platform yang baru. Wishbone SoC sambungan arkitek bertujuan untuk menyediakan satu penyelesaian yang baik untuk isu-isu integrasi SoC dengan mempunyai spesifikasi perantaraan yang sama. Dalam projek ini, Wishbone bas komputer untuk 32-bit teras dilaksanakan dalam Sistem Verilog bersama-sama dengan tiga skim pengawal trafik bas yang berlainan. Skim tersebut ialah keutamaan tetap, robin bulat, dan kawalan keutamaan. Selain itu, saiz optimum pemindahan untuk bas Wishbone juga dicari dari segi kadar pengeluaran bas dan kitaran tunggu purata. Saiz optimum pemindahan untuk bas Wishbone didapati adalah 64 bytes. Akhir sekali, bas Wishbone juga digunakan untuk memeriksa prestasi bas Wishbone untuk skim pengawal traffic bas yang berbeza dalam simulasi Modelsim. Skim pengawal traffic bas robin bulat merupakan yang terbaik antara tiga skim dari segi kadar pengeluaran bas, kerumitan logik, dan kitaran tunggu maksimum.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

Hz	-	Hertz
VLSI	-	Very Large Scale Integration
SoC	-	System on Chip
IP	-	Intellectual Property
HDL	-	Hardware Description Language
BFM	-	Block Functional Module
RTL	-	Register Transfer Level
CPU	-	Central Processing Unit
I/O	-	Input Output
RMW	-	Read Modify Write
TDMA	-	Time Division Multiplexing Access
ASM	-	Algorithmic State Machine
MSB	-	Most Significant Bit
FSM	-	Finite State Machine
RST	-	Reset
CYC	-	Cycle

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

The world's first single-chip microprocessor was introduced by Intel in November 1971. It was named as Intel 4004 and has 2300 transistors which can run at a clock speed of up to 740 KHz [1]. Since then, the compute power of microprocessor increase exponentially year by year. Besides that, the number of transistor on a single chip is also following a trend at which it is doubled every 18 months. This trend of transistor number on a single chip is widely known as Moore's Law which was named after Gordon E. Moore, the co-founder of Intel Corporation. This law still holds true until today. Today, the latest microprocessor available in the market is manufactured using 22nm process technology and the transistor count was more than 1 billion. The latest Intel Core i7 Processor is able to run at clock speed up to 4.00 GHz which is about 5000 times higher than the first Intel 4004.

The increasing number of transistor per chip is mainly due to transistor scaling throughout each technology node. This enabled the possibilities of performing more features in a chip because more and more transistor can be packed inside a chip. Therefore the complexity of logic calculation a chip can perform is also increased. Besides transistor scaling, microarchitecture also play an important role in microprocessor performance. With the advances in microarchitecture, the compute power of microprocessor can also be increased dramatically. We have seen aggressive clock scaling and introduction of speculative execution, parallel instruction issue, out-of-order processing, and larger caches over the course of microprocessor development [1]. All of these were introduced with the goal of increasing microprocessor performance.

On top of transistor scaling and microarchitecture enhancement, the VLSI (Very Large Scale Integration) industry is trending towards SoC (System-on-Chip) design. SoC is a technology that packages the whole system inside a single chip. It has many advantages over traditional VLSI design such as fast, low power, small size, and low cost.

One of the challenges in the SoC world is the pressure of time to market for chip manufacturers. Because of the fast changing requirements in the consumer world, chip manufacturers no longer have the luxury of long design cycle time. In the contrary, the design cycle time need to be minimal so that the product can reach the market fast and on time. In order to reduce design cycle time, one of the solution is to maximize the reuse of previously designed and proven working IP and integrate different IPs together to form a new product with desired features. Hence, the successfulness of a SoC product depends heavily on effectiveness of integration of different IP cores.

In order to ease integration work, the compatibility of IP (intellectual property) cores is a vital issue that needed to be taken care of. Most of the time, IP cores that are planned for reuse usually are designed earlier and the input output specifications are inconsistent with the new platform [2]. This will cause IP integrator to have the need to consider interface design and testing which will cause complication in integration process [3]. As a result, integration of different IP becomes harder with non-standard interconnection scheme.

The Wishbone SoC Interconnection Architecture for Portable IP Cores is introduced to use with semiconductor IP Cores with an aim to provide flexible design methodology. It is designed to be a reliable and good solution for SoC integration issues. Besides that, it also fosters design reuse for IP cores that follow common interface specifications. Having a common interface specification is important to ensure that the integration of different IP cores is at minimal effort. Moreover, usually IP provider is from different vendor so this is even more important to have common interface specifications because the integrator might not know very well with the IP cores itself.

Wishbone arbitration scheme is user defined to arbitrate multiple bus requests at the same time. When two or more bus masters are requesting to use the shared system bus at the same time, there is a need of an efficient arbitration scheme to handle this else there will be bus contention issue since two or more bus master is sending its respective data at the same time to destination address. This will cause the data receiving at the bus slave to have errors. Without a good arbitration scheme, it might cause starvation of bus master whereby certain bus master did not get bus grant even after waited for long period of cycle time because the shared bus is in used by other bus master. Therefore, the decision of which bus master getting granted should be made fair so that every bus master has the equal opportunity to use the shared bus. Hence, the efficiency of arbiter will have impact on system performance because the cycle time to complete a transaction is not the same for different arbitration scheme.

1.1 Problem Statement

The problem statements for this project are as follows:

- IP cores that is designed with non-standard interfaces and transaction protocol cause extra effort in integration works and slow down design cycle
- The arbitration scheme in Wishbone bus is user defined. There is a lack of data to compare the performance of different arbitration schemes in Wishbone bus.

1.2 Research Objectives

With the problem statement presented on previous section, this project is aim to have objectives that are able to solve the problem. Therefore, the research objectives for this project are as follows:

- To implement the Wishbone on-chip computer bus with arbitration for 32-bit cores in system verilog
- To examine the performance of different arbitration schemes of the Wishbone bus

1.3 Project Scope

This project will be focusing on the following scopes:

- A shared bus system is developed with address and data width of 32 bits using Wishbone bus system in SystemVerilog
- The Wishbone shared bus system is used to examine performance of the following 3 arbitration schemes that support up to five bus masters:
    - Fixed Priority
    - Round Robin
    - Priority with waiting time control
- Bus master and bus slave are emulated with BFM (Block Functional Module) in SystemVerilog that can send bus traffic
- Altera Modelsim Starter Edition 10.1d is used as simulation tool

1.4 Contribution of Work

The objectives of this project are to implement a 32-bit Wishbone on chip computer bus and to examine the performance of different arbitration schemes of the Wishbone bus. There are three different arbitration schemes implemented. The three arbitration schemes are:

1. Fixed Priority
2. Round Robin
3. Priority with waiting time control

Therefore, this project presents a functional Wishbone Shared bus system RTL that is come with built in arbiter. The Wishbone Shared bus system is synthesis ready and is able to integrate with other IP cores that is following standard Wishbone interface. Also, there will be three different arbitration schemes presented in this project. Each of the arbitration schemes is supporting up to five bus masters. After that, this project compares the performance between these three arbitration schemes quantitatively on Wishbone Shared bus system. It finally come up with a conclusion of what is the optimum transfer size of Wishbone shared bus system and which arbitration scheme is a more suitable one compared with the other two.

REFERENCES

1. Danowitz, A., et al., *CPU DB: recording microprocessor history.* Commun. ACM, 2012. **55**(4): p. 55-63.
2. OpenCores, *WISHBONE System-on-Chip (SoC)Interconnection Architecturefor Portable IP Cores*. 2010.
3. Gajski, D.D., et al. *Essential issues for IP reuse.* in *Design Automation Conference, 2000. Proceedings of the ASP-DAC 2000. Asia and South Pacific*. 2000.
4. Null, L. and J. Lobur, *The Essential of Computer Organization and Architecture*. 2003: Jones and Bartlett.
5. Patterson, D.A. and J.L. Hennessy, *Computer Organization and Design*. 3rd edition ed. 2005: Morgan Kaufmann.
6. Sharma, M. and D. Kumar, *WISHBONE BUS ARCHITECTURE – A SURVEY AND COMPARISON.* International Journal of VLSI design & Communication Systems (VLSICS), 2012. **3**(2).
7. ARM. *AMBA Specification (rev 2.0)*. 1999; Available from: http://www.arm.com.
8. Altera. *Avalon bus specification: Reference Manual*. 2003; Available from: http://www.altera.com.
9. IBM. *CoreConnect bus Architecture*. 1999; Available from: www.ibm.com/chips/products/coreconnect/
10. Sharma, M. and D. Kumar. *Design and synthesis of Wishbone bus Dataflow interface architecture for SoC integration*. in *India Conference (INDICON), 2012 Annual IEEE*. 2012.
11. Conti, M., et al., *Performance analysis of different arbitration algorithms of the AMBA AHB bus*, in *Proceedings of the 41st annual Design Automation Conference*. 2004, ACM: San Diego, CA, USA. p. 618-621.
12. Ahmad Zabidi, M.M.i. and S. Rajagopal, *Open Source microprocessor and on-chip-bus for system-on-chip*, in *Advances In Embedded Systems*. 2008, Penerbit UTM: Johor. p. 1-20.
13. Lu, R. and C.-K. Koh, *A high performance bus communication architecture through bus splitting*, in *Proceedings of the 2004 Asia and South Pacific Design Automation Conference*. 2004, IEEE Press: Yokohama, Japan. p. 751-755.

14. Ruibing, L., C. Aiqun, and K. Cheng-Kok, *SAMBA-Bus: A High Performance Bus Architecture for System-on-Chips.* Very Large Scale Integration (VLSI) Systems, IEEE Transactions on, 2007. **15**(1): p. 69-79.
15. Chang Hee, P., et al. *The efficient bus arbitration scheme in SoC environment.* in *System-on-Chip for Real-Time Applications, 2003. Proceedings. The 3rd IEEE International Workshop on.* 2003.
16. Kandasamy, S., *32 Bit Low-speed Bus System For Mips Uart And Ps2 Interfacing*, in *Faculty of Information and Communication Technology.* 2010, University of Tunku Abdul Rahman.
17. Poletti, F., et al., *Performance Analysis of Arbitration Policies for SoC Communication Architectures.* Design Automation for Embedded Systems, 2003. **8**(2-3): p. 189-210.
18. Loghi, M., et al. *Analyzing on-chip communication in a MPSoC environment.* in *Proceedings of the conference on Design, automation and test in Europe-Volume 2.* 2004. IEEE Computer Society.
19. Shrivastav, A., G. Tomar, and A.K. Singh. *Performance Comparison of AMBA Bus-Based System-On-Chip Communication Protocol.* in *Communication Systems and Network Technologies (CSNT), 2011 International Conference on.* 2011. IEEE.
20. Christiansen, T. and N. Torkington, *Perl cookbook, second edition.* 2003: O'Reilly Media, Inc. 976.