

**MEMORY MANAGER FOR SIMPLEST
MICROPROCESSOR ON FPGA**

ABBAS IBRAHIM MBULWA

**THESIS SUBMITTED IN PARTIAL FULFILMENT
OF THE REQUIREMENT FOR THE DEGREE OF
BACHELOR OF COMPUTER ENGINEERING**

**PERPUSTAKAAN
UNIVERSITI MALAYSIA SABAH**

**FACULTY OF ENGINEERING
UNIVERSITI MALAYSIA SABAH**

2015




UMS
UNIVERSITI MALAYSIA SABAH

DECLARATION


I hereby declare that this thesis, submitted to Universiti Malaysia Sabah as partial fulfillment of the requirements for the degree of Bachelor of Computer Engineering, has not been submitted to any other university for any degree. I also certify that the work described herein is entirely my own, except for quotations and summaries sources of which have been duly acknowledged.

This thesis may be made available within university library and may be photocopied or loaned to other libraries for the purposes of consultation.

12 June 2015


Abbas Ibrahim Mbulwa

CERTIFIED BY



Ir. Hj. Othman Ahmad

SUPERVISOR



ACKNOWLEDGEMENT

First of all I thank Allah for endowing me with health, patience and knowledge to complete this project. Special appreciation goes to my supervisor, Ir. Othman Ahmad, for his supervision and constant support. His invaluable help of constructive comments and suggestions have contributed to the success and completion of this project.

I would also like to express my sincere gratitude towards Islamic Development Bank (IDB) scholarship program that paved an opportunity for me to pursue a degree program in Computer Engineering from this prestigious university.

I am deeply indebted to all the Lecturers in the Department of Computer Engineering for their comprehensive and unconditional support during my studies in University Malaysia Sabah.



ABSTRACT

Preliminary studies show that commercial microprocessors are not suited for many research projects due to the fact that they are closed source and they cannot be modified. Most non-commercial microprocessor architectures do not have proper memory management that can enable and extend their usability. The proper use of computer memory is vital to ensure efficiency of process executions in microprocessor systems. This project designed a memory management of a simplest microprocessor on FPGA platform. The project implements the design of the simplest 32-bit microprocessor and its memory system using schematic approach. The project uses the concept of management as the art of managing the main memory, buffer and registers. The memory management was designed to allow the microprocessor to use main memory or I/O buffers in a controlled manner with the help of the microprocessor signals generated from a well designed control unit by using four operation code bits. This memory management is also designed to improve memory bandwidth so as to reduce the microprocessor to memory speed miss-match. This design showed some of the characteristic of reliable future microprocessor design such as flexibility, programmability, adaptable, reconfigurable. Altera DE2-115 board was used to validate the design.



ABSTRAK

Keputusan awal menunjukkan bahawa mikropemprosesan komersial tidak sesuai dalam banyak projek disebabkan fakta yang menunjukkan ia merupakan sumber tertutup dan tidak mudah untuk diubahsuai. Kebanyakan senibina mikropemprosesan bukan komersial tidak mempunyai pengurusan memori yang teratur yang membolehkan pengembangan kegunaannya. Kegunaan Rekabentuk komputer yang betul adalah sangat penting untuk memastikan kecekapan proses dilancarkan dalam system mikropemprosesan. Projek ini dilaksanakan dengan menggunakan rekabentuk 32-bit mikropemprosesan yang ringkas dan system memori yang menggunakan pendekatan skematik. Projek ini mengamalkan konsep pengurusan sebagai seni untuk mengurus memori utama, penampakan dan pendaftar. Pengurusan memori telah direka untuk membenarkan mikropemprosesan menggunakan memori utama ataupun penampakan keluar masuk secara terkawal dengan bantuan isyarat mikropemprosesan yang dijana dari unit kawalan yang dibina dengan menggunakan empat kod operasi. Pengurusan memori ini juga dibina untuk menaiktaraf jalur lebar memori serta mengurangkan kelajuan memori mikropemprosesan daripada tidak serasi. Rekabentuk ini menunjukkan beberapa ciri-ciri dipercayai untuk kegunaan rekabentuk mikropemprosesan pada masa hadapan seperti fleksibiliti, kebolehprograman, kesesuaian dan pembentukan semula. Papan Altera DE2-115 telah digunakan untuk mengeahkan rekabentuk ini.



CONTENTS

| | Page |
|--|------|
| DECLARATION | iii |
| ACKNOWLEDGEMENT | iv |
| ABSTRACT | v |
| ABSTRAK | vi |
| CONTENTS | vii |
| LIST OF TABLES | xii |
| LIST OF FIGURES | xiii |
| LIST OF SYMBOLS AND ABBREVIATIONS | xvi |
| CHAPTER 1 INTRODUCTION | |
| 1.1 Background | 1 |
| 1.2 Project Description | 2 |
| 1.3 Problem statement | 4 |
| 1.4 Project Objective | 5 |
| 1.5 Project Scope | 6 |
| 1.6 Project Workflow | 7 |
| 1.7 Organisation of Report | 9 |



CHAPTER 2 LITERATURE REVIEW

| | | |
|------------|---|-----------|
| 2.1 | Introduction | 10 |
| 2.2 | Why FPGA | 11 |
| 2.3 | FPGA Technologies and Architecture | 12 |
| 2.3.1 | Configurable Logic Blocks | 14 |
| 2.3.2 | Configurable Interconnects | 16 |
| 2.3.3 | FPGA Configuration | 17 |
| 2.3.4 | FPGA Design Flow | 17 |
| 2.4 | Altera FPGA | 19 |
| 2.4.1 | Cyclone IV device | 19 |
| 2.4.2 | DE2-115 Development Board | 21 |
| 2.4.3 | Quartus II Design Tool | 24 |
| 2.5 | Microprocessor Architecture | 27 |
| 2.5.1 | RISC Architecture | 27 |
| 2.5.2 | CISC Architecture | 27 |
| 2.5.3 | RISC versus CISC | 28 |
| 2.6 | Configurable Microprocessors | 29 |
| 2.6.1 | SWORD32 | 29 |
| 2.6.2 | MicroBlaze | 31 |
| 2.7 | Computer Memory | 33 |
| 2.7.1 | ROM | 33 |
| 2.7.2 | RAM | 36 |
| 2.7.3 | FIFO | 38 |
| 2.7.4 | Cache Memory | 40 |

| | | |
|-------|--|----|
| 2.7.5 | Memory Hierarchy | 40 |
| 2.7.6 | Cache Memory as a Memory Management Approach | 42 |

CHAPTER 3 SYSTEM DESIGN

| | | |
|-------|---|----|
| 3.1 | Introduction | 44 |
| 3.2 | System Description | 48 |
| 3.3 | Project Requirements and System Modules | 47 |
| 3.3.1 | Simplest 32-bit Microprocessor | 47 |
| 3.3.2 | Memory Management | 49 |
| 3.4 | System Design Method and Tools | 51 |
| 3.4.1 | System Design Flow | 51 |
| 3.4.2 | Design Approach | 53 |
| 3.4.3 | Altera Quartus II software | 53 |
| 3.4.4 | Logic Friday | 54 |
| 3.4.5 | Altera DE2-115 FPGA Board | 55 |
| 3.5 | Designing of a Simplest 32-bit Microprocessor | 55 |
| 3.5.1 | Opcode | 56 |
| 3.5.2 | Instruction Set Architecture | 57 |
| 3.5.3 | Opcode Decoding | 58 |
| 3.5.4 | Control Unit and ALU | 59 |
| 3.6 | Memory Management Module Design Method | 60 |

CHAPTER 4 RESULT AND DISCUSSION

| | | |
|------------|------------------------------------|-----------|
| 4.1 | Introduction | 62 |
| 4.2 | Microprocessor Control Unit | 63 |
| 4.2.1 | Procedure | 63 |
| 4.2.2 | Schematic | 64 |
| 4.2.3 | Sample Input Data and Output | 65 |
| 4.2.4 | Observation | 66 |
| 4.2.5 | Discussion | 66 |
| 4.3 | Simplest Microprocessor | 67 |
| 4.3.1 | Procedure | 67 |
| 4.3.2 | Schematic | 69 |
| 4.3.3 | Sample program | 71 |
| 4.3.4 | Observation | 72 |
| 4.3.5 | Discussion | 73 |
| 4.4 | Memory Management | 73 |
| 4.4.1 | Procedure | 74 |
| 4.4.2 | Schematic | 74 |
| 4.5 | Overall System | 75 |
| 4.5.1 | Procedure | 75 |
| 4.5.2 | Schematic | 77 |

| | | |
|-------|------------------------|----|
| 4.5.3 | Sample Program | 79 |
| 4.5.4 | Observation and Result | 79 |
| 4.5.5 | Discussion | 81 |

CHAPTER 5 CONCLUSION AND RECOMMENDATIONS

| | | |
|-----|--------------------------------|----|
| 5.1 | Future Work and Recommendation | 82 |
| 5.2 | Conclusion | 82 |

| | | |
|-------------------|--|-----------|
| REFERENCES | | 84 |
|-------------------|--|-----------|

| | | |
|-------------------|--|-----------|
| APPENDICES | | 86 |
|-------------------|--|-----------|

| | | |
|------------|--|----|
| Appendix A | Top and Bottom view of the Altera DE2-115 Development Board | 87 |
| Appendix B | The DE2-115 Board, Powered and connected to Programming Cable | 88 |
| Appendix C | Observation of the Microprocessor Control Unit Output on the DE2-115 Board | 89 |
| Appendix D | Output of the Microprocessor as Observed on the DE2-115 Board | 91 |

LIST OF TABLES

| Table No. | | Page |
|-----------|--|------|
| 1.1 | Resources for the Cyclone IV E Device Family | 20 |
| 1.2 | Quartus II Compilation Process | 25 |
| 3.1 | Opcode and Instruction Decoding Table | 59 |
| 4.1 | Sample Input Data and Observed Output of the Control Unit | 65 |
| 4.2 | Sample Instructions | 71 |
| 4.3 | Sample Input Data and Observed Output of the Simplest Microprocessor | 72 |
| 4.4 | Sample Program inserted into Program Memory to test the Overall System | 79 |
| 4.5 | Observation During the Execution of Sample Instruction on the System | 80 |

LIST OF FIGURES

| Figure No. | | Page |
|------------|---|------|
| 1.1 | Project work Flow | 8 |
| 2.1 | Basic FPGA Architecture | 13 |
| 2.2 | LUT diagram | 14 |
| 2.3 | OR gate and truth table | 15 |
| 2.4 | CLB diagram. | 15 |
| 2.5 | Programmable Switches and CLB | 16 |
| 2.6 | Typical traditional standard FPGA design flow | 18 |
| 2.7 | DE2-115 board (Block diagram) | 22 |
| 2.8 | Quartus II Software (User Interface) | 24 |
| 2.9 | Basic Design Flow Using Quartus II Integrated Synthesis | 26 |
| 2.10 | SWORD32™; 8 bit Instruction format | 30 |
| 2.11 | SWORD32™; 16 bit Instruction format | 30 |
| 2.12 | SWORD32™; 19 bit Instruction format | 31 |
| 2.13 | MicroBlaze top-level connections | 32 |

| | | |
|------|---|----|
| 2.14 | Instruction Format of the MicroBlaze architecture | 33 |
| 2.15 | Data Format of the MicroBlaze architecture | 33 |
| 2.16 | A ROM stores four 4-bit words | 34 |
| 2.17 | Single Port ROM model from Altera with only one address line | 35 |
| 2.18 | Dual Port ROM model from Altera with two address lines and two output lines | 35 |
| 2.19 | MOSFET transistors in SRAM cell | 36 |
| 2.20 | Single Port RAM model from Altera | 37 |
| 2.21 | A simple FIFO memory model | 38 |
| 2.22 | Single Clock FIFO model from Altera | 39 |
| 2.23 | Cache Memory | 40 |
| 2.24 | Memory Hierarchy | 41 |
| 2.25 | Typical Cache organization | 42 |
| 3.1 | Block Diagram of The System | 46 |
| 3.2 | Block Diagram of The Simplest microprocessor. | 48 |
| 3.3 | The block diagram of the Memory Management | 50 |
| 3.4 | System Design Flow Diagram | 52 |

| | | |
|------|--|----|
| 3.5 | Design entry on the Quartus II Software | 53 |
| 3.6 | Logic Friday User Entry, Minimised logic and circuit | 54 |
| 3.7 | Opcode bits at LSB of the 32 bits Program Data | 56 |
| 3.8 | Instruction Format of the Simplest Microprocessor | 57 |
| 3.9 | The Opcode and instruction decoder circuit | 58 |
| 3.10 | 1 Bit address decoding to enable main memory or I/O buffer | 60 |
| 4.1 | The Circuit Diagram of the Microprocessor Control Unit | 64 |
| 4.2 | Program Memory (ROM) for the Simplest Microprocessor | 67 |
| 4.3 | Circuit Diagram of the Simplest Microprocessor | 69 |
| 4.4 | Memory Initialisation for Program Memory | 71 |
| 4.5 | The schematic of the memory management | 74 |
| 4.6 | The schematic of the overall system - The memory manager for a simplest microprocessor | 77 |

| | |
|---------|---|
| ROM | Read Only Memory |
| RTL | Register Transfer Level |
| SoC | System on Chip |
| SPARC | Scalable Processor Architecture, a RISC ISA developed by Sun Microsystems |
| SRAM | Static RAM |
| SWORD32 | Simplest Word-size Scalable Microprocessor, a 32-bit version developed at University Malaysia Sabah |
| VHDL | VHSIC HDL |
| VHSIC | Very-High-Speed Integrated Circuit |
| IC | Integrated Circuit |
| I/O | Input-Output |
| IP | Intellectual Property |
| ISA | Instruction Set Architecture |

CHAPTER 1

INTRODUCTION

1.1 Background

The microprocessor is the brain of a computer. Its architecture contributes to the determination of how fast the computer will be and what capabilities the machine will have. There are several choices when it comes to microprocessor system design such as when using Digital Signal Processors (DSPs), Field-Programmable Gate Arrays (FPGAs), or Application Specific Integrated Circuits (ASICs). FPGAs have become a popular alternative and useful to actually replace custom ASICs and processors for signal processing and control applications. FPGAs have become popular because of their ability to be reprogramed. Using prebuilt logic blocks and programmable routing resources, you can configure these chips to implement custom hardware functionality based on specified requirements.

Ashley et al. (1996) argued that, although computing development continues to squeeze more computational performance from microprocessor systems, the amount of cache remains small relative to the system memory. Since any sort of cache access has a performance inversely proportional to the number of cache misses, we find that



in many cases a computer's speed bottleneck is not in the microprocessor but on the access method and times of the memory.

According to Doug, James, and Alain (1996), the reason why microprocessors fail to scale is because of memory latency and computational overhead. The largest computational bottlenecks occur between the interface of main memory (RAM) and the microprocessor. Doug et al. (1996) suggested that, lightweight multichannel memory with a fast message passing system bandwidth can open the door to novel ways of distributing data and computation.

1.2 Project Description

The memory is designed for use with one system; it simply includes read/write, input data, output data and their enable signals. For the proper utilization of memory by a certain processing unit, there must be some management between memory and the processor in order to facilitate proper utilisation of memory and to avoid errors and data corruptions. This can be very helpful for many different applications to save time, reduce costs, save space, reduce complexity and more. The method to facilitate the effective memory utilization is by implementing a memory management system. The memory management systems are also designed to increase the memory speed so that memory operations can catch-up with the microprocessor speed. There are different memory management approaches used in ASICs and even in some FPGA designs for example cache memory and virtual memory, which are mostly used in recent microprocessors. But the amount of cache memory used remains small relative to the main memory (RAM). Apart from small size of cache, cache memory still suffers from another problem

of cache miss (William 2010; Tanenbaum 200; Ashley et al. 1996). How about looking into another memory management approach, a little different from cache memory? And FPGAs are magnificent for prototyping due to their programmability.

According to Weber (2001), an arbiter is the term used for an object that facilitates or arbitrates interaction between two distinct blocks. The arbiter follows a set of rules to pass the communication between the microprocessor, other blocks and main memory. The memory manager is designed to handle the complexity of memory operations/usage necessary to ensure proper data transfer. The microprocessor architecture used is the simplest RISC architecture the designed microprocessor in this project is inspired by the simplest word-size scalable 32-bit microprocessor known as SWORDS32. According to Othman (2012), the unique of SWORD32™ microprocessor is its scalability and customizability. While memory manager in the microprocessor architecture can be implemented and utilized in a variety of applications, this project has implemented the design on FPGA. A consideration that a memory manager needs to take is how the whole process of memory read/write operations takes place and how it determines which bus is granted access in order to effectively utilize the memory.

This project set out to design and create the memory manager using Schematic approach. And validate it on ALTERA™ DE2-115 development board. There are a few goals that were essential for the progression and completion of this project. First, to design a simplest microprocessor and control unit for that generates control signals including memory and register read/write signals, so that there is a functioning interface to memory. Second, the memory manager should be designed and interfaced with the simplest microprocessor. Thirdly, a validation process for the memory manager on a microprocessor carried out. These three goals were necessary to complete the design.

1.3 Problem Statement

Microprocessor systems suffer the speed miss-match between the fast microprocessor and the slow main memory (RAM). This is to both configurable and non-configurable microprocessors. Microprocessor is made of flip-flops and gates; this makes the microprocessor to have very fast processing speed in terms of GHz. The main memory or RAM does not have the speed of the processor. For instance DDR3 SDRAM has speed around 1033MHz. This is not enough for the processing speed of the processor hence result a critical problem for fetching and executing instructions because the speed of the CPU and RAM does not match. If the CPU access the RAM and take the data and instructions for processing, the CPU's speed would be limited to the speed of the RAM. This slows down the processing speed and the overall efficiency of the system. Cache memory was introduced to overcome this bottleneck of speed of data transferring but it is also not sufficient as it is relative small compare to RAM and it experiences cache misses which waste the processing time as reported by Ashley et al. (1996).

In order to provide the best achievable flexibility in hardware, scalability and configurability is required. Many microprocessors have been invented and their hardware and software implementations may be widely available. The available invention or designs can be categorized in two kinds, i.e. microprocessors with a commercial license and microprocessors with an open source license. Commercial microprocessors are mostly provided as firm-core implementations. They are technology dependent and are mostly practically non-modifiable. On the other hand, microprocessor with non-commercial license - Currently 95 microprocessors are listed on OpenCores according to OpenCores (2015), they are configurable but they are very limited to scalability and usability.

An important requirement for microprocessors, which are useful for specific functions or studies, in either research or learning purpose, is that they can be changed— in terms of ISA, bus implementation or memory implementation to support multiple processing functionality. Research topics change rather quickly and every topic will probably have different requirements and expectations of a microprocessor in terms of features, clock frequency and processor size. Due to these differences in requirements a processor needs to be configurable and yet has high performance metric. Furthermore, research depends heavily on simulation of components, but simulating firm-core designs is very time-consuming compared to the simulation of a behavioural model. According to Holsmark et al. (2004), customizable processors like MicroBlaze for instance – its simulation and integration within project is a time consuming and exhausting task because of its less customizability.

This project has designed a memory manager on customisable microprocessor which its design is inspired by SWORD32™ and Mano RISC processor, so as to increase their configurability, usability and performance.

1.4 Project Objective

The objective of this project is to develop a memory manager for a customizable yet simplest 32-bit RISC microprocessor which is designed based on SWORD32™ architecture and Mano RISC microprocessor. The result of this development is a microprocessor with its memory manager, which will be suitable for a wide range of development projects and research projects in the field of Computer Engineering.

This development is ought to be suitable and successful if the following design goals are met.

- To design, validate and implement a simplest 32bit microprocessor by first design its control unit, instruction set, and decode opcode bits so as to obtain different instructions and control signals.
- To design memory manager that uses the memory control signal and manage the microprocessor to main memory access as well as buffers and other blocks.
- To interface the simplest microprocessor and the memory manager to obtain the overall design, then to validate the design on Altera DE2-115 Board.

1.5 Project Scope

The design natively executes simple 32-bit instructions and handles memory read/write operations with the help of a designed memory manager which is the link between the simplest 32-bit microprocessor, the memory resource modules and the I/O. The microprocessor shall interface with a memory manager to handle its communication with memory resource which consists of main memory, registers and I/O buffers. The implementation of this design is done on machine code level by using microcodes which have operands, opcode, registers and immediate values.

The memory manager will consist of address decoder, arbiter, and bus system which will route the data to the correct module, either main memory or buffers to the microprocessor. A memory arbitration part is to monitor access to the main memory or I/O buffers so as to control traffic.

This project set out to design the memory manager by using schematic approach. And then validated it using ALTERA™ DE2-115 development board. There are few goals that were essential for the progression and completion of this project. First, sending read/write signals to memory, which would be done with no memory manager, just a simple direct memory access so that there is a functioning interface to memory. This is achieved by designing a simplest microprocessor with a control unit that generates read/write signals along with other important signals required to meet objectives of the project. Second, the memory manager is designed and interfaced with the designed microprocessor to form one integrated design. The purpose of the memory manager is to manage and improve the usage of the computer memory such as creating effective memory fetch cycles (memory read and write), effective movement of data between main memory, registers and buffers and selection of memory such as main memory, registers or I/O Buffers to work on a particular execution process. This is to eliminate the speed miss-match between microprocessor and memory. Thirdly, a validation process for the memory manager would be carried out. These three goals are necessary to create a useful, functional memory manager.

1.6 Project Workflow

The design of memory manager for simplest microprocessor on FPGA started with literature search on memory technologies and different memory management approaches that exist today. The literature search went further to study different configurable microprocessor and the whole concept of designing on FPGA platforms. The SWORD32™ microprocessor was studied and tested on Altera DE2-115 development

board. Then the simplest microprocessor was designed. The design of the simplest microprocessor is inspired by the SWORD32™ and the Mano RISC Processor. Then the schematic for the designed system that contains simplest microprocessor module and memory manager module, is analyzed and compiled on Quartus II software. Then the overall design is tested and validated on the DE2-115 board.

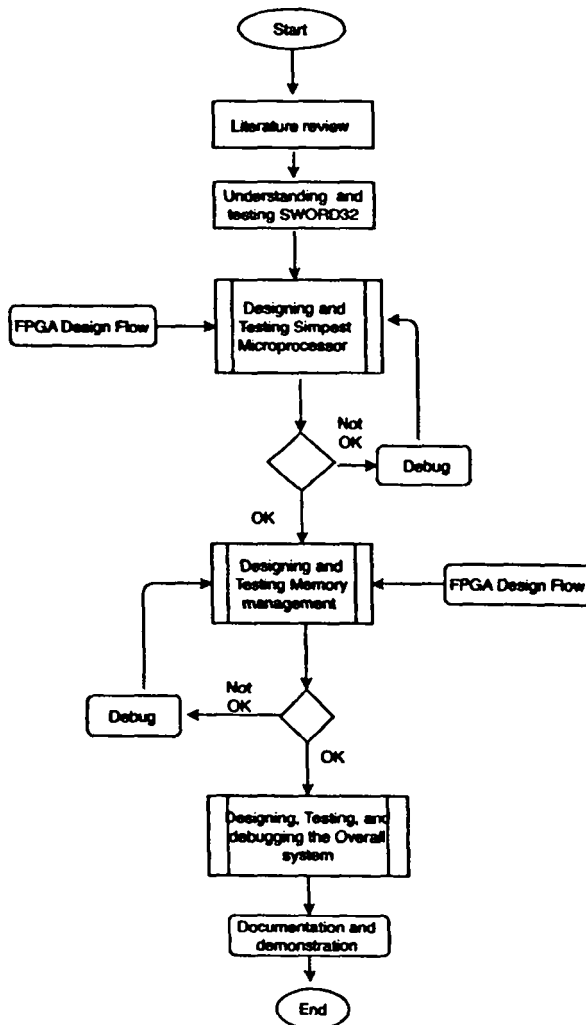


Figure 1.1 Project Workflow

1.7 Organization of Report

The organization of this report is as follows. Chapter 1 gives the elaboration of the project, which includes project description, problem statement, project objective, project scope, workflow and organization. Within the project description and problem statement, the motivation is stated. In Chapter 2 existing simplest and scalable microprocessor architecture (SWORD32™) is evaluated as well as different memory technologies. FPGA platform tools to be used for development were also elaborated in detail. Chapter 3 is about the system design, adaptation of the selected microprocessor architectures and explanation of the design module by module starting by simplest microprocessor control unit design, ALU, and memory management design. A great deal of effort is put on this chapter on making and improving the design, especially on the memory management and microprocessor part so as to make the system works.

In Chapter 4, a simplest microprocessor is tested for compliance with the ISA design and opcode, as well as control unit signals. Each module designed as explained on chapter 3 were tested and validated by using DE2-115 board. This chapter also analyses and discuss about the obtained results. Chapter 5 will give a reflection of what has been achieved during this project it also gives recommendation regarding the design.

REFERENCE

- Altera. 2004. Cyclone IV Device Handbook volume 1, 2, and 3 Practices. Retrieved 10 March 2015 from <http://www.altera.com/literature/>
- Altera 2009. FPGA Design Methodology and Guideline. Retrieved 10 March 2015 from <http://www.altera.com/literature/>
- Altera. 2013. Internal Memory (RAM and ROM) User Guide. Retrieved 10 March 2015 from <http://www.altera.com/literature/>
- Altera. 2014. Recommended Design Practices. Retrieved 10 March 2015 from <http://www.altera.com/literature/>
- Ashley Saulsbury, Fong Pong, Andreas Nowatzky. 1996. Missing the Memory Wall: The Case for Processor/Memory Integration (ISCA 1996). Sun Microsystems, ACM 0-89791-786-3.
- Bob Zeidman. 2002. Designing with FPGAs and CPLDs. pp 33-139. CMP Books.
- Bruce Jacob, Spencer Ng, David Wang. 2007. Memory systems: cache, DRAM, disk. Morgan Kaufmann.
- Doug Burger, James R. Goodman, Alain Kagi. 1996. Memory Bandwidth Limitations of Future Microprocessor. Computer Science Department, University of Wisconsin Madison
- Goossens, K., J. Dielissen, A. Radulescu. 2005. Ethereal network on chip Concepts, architectures, and implementations. IEEE Design and Test of Computers 22(5): 414-421.
- Holsmark, R., A. Johanson, and S. Kumar. 2004. Connecting cores to packet switched on-chip networks: A case study with MicroBlaze processor cores. 7th IEEE Workshop DDECS, 04 April.
- Ian Kuon, Russell Tessier and Jonathan Rose. 2008. FPGA Architecture - Survey and Challenges. Foundations and Trends in Electronic Design Automation .Vol. 2, No. 2 (2007): 135-253.
- Othman. 2012. Simplest Word-size Scalable Microprocessor. Invention convention, : 2012, Universiti Malaysia Sabah. IPC 2014.01
- Mano, M.M, Kime, C.R. 2008: Computer Design Fundamentals. Upper Saddle River,



NJ: Prentice Hall.

OpenCores. Configurable processors. 2015. Retrieved 17 May 2015 ftp: //www.
opencores.org.

Philip H.W. Leong. 2008. Recent Trends in FPGA Architectures and Applications. 4th
IEEE International Symposium on Electronic Design, Test & Applications
(DELTA 2008), page(s): 137–141.

Rajit Manohar, 2001. The Future of FPGAs. Cornell University, Computer System Lab.
Retrieved 20 May 2015, from <http://vlsi.cornell.edu>

Tanenbaum A.S. 2000. Structured Computer Organization. Upper Saddle River, NJ:
Prentice-Hall.

Weber, M. 2001. Arbiters: Design Ideas and Coding Styles. Boston: Synopsys Users
Groups.

William Stallings. 2010. Computer organization and architecture, designing for
performance, 8th Ed, pp. 110-180,348-518.Upper Saddle River, NJ: Prentice
Hall.

Zhang, W., G. D, Y. X, M. Gao, L. Geng, B. Zang, Z. Jiang, N. Hou, and Y. Tang.2006.
Design of a hierarchy-bus based MPSoC on FPGA. 8th International
Conference on Solid-State and Integrated Circuit Technology (ICSICT 2006),
1966–1968.