

GLOBAL JOURNAL

OF COMPUTER SCIENCE AND TECHNOLOGY: A

Hardware & Computation

Navigation Baseband SoC

Scrambler & Descrambler system

Highlights

The Implementation of DMA

Monitoring and Controlling System

Discovering Thoughts, Inventing Future

VOLUME 15

ISSUE 2

VERSION 1.0



GLOBAL JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY: A
HARDWARE & COMPUTATION

GLOBAL JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY: A
HARDWARE & COMPUTATION

VOLUME 15 ISSUE 2 (VER. 1.0)

OPEN ASSOCIATION OF RESEARCH SOCIETY

© Global Journal of Computer Science and Technology. 2015.

All rights reserved.

This is a special issue published in version 1.0 of "Global Journal of Computer Science and Technology" By Global Journals Inc.

All articles are open access articles distributed under "Global Journal of Computer Science and Technology"

Reading License, which permits restricted use. Entire contents are copyright by of "Global Journal of Computer Science and Technology" unless otherwise noted on specific articles.

No part of this publication may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopy, recording, or any information storage and retrieval system, without written permission.

The opinions and statements made in this book are those of the authors concerned. Ultraculture has not verified and neither confirms nor denies any of the foregoing and no warranty or fitness is implied.

Engage with the contents herein at your own risk.

The use of this journal, and the terms and conditions for our providing information, is governed by our Disclaimer, Terms and Conditions and Privacy Policy given on our website <http://globaljournals.us/terms-and-condition/menu-id-1463/>

By referring / using / reading / any type of association / referencing this journal, this signifies and you acknowledge that you have read them and that you accept and will be bound by the terms thereof.

All information, journals, this journal, activities undertaken, materials, services and our website, terms and conditions, privacy policy, and this journal is subject to change anytime without any prior notice.

Incorporation No.: 0423089
License No.: 42125/022010/1186
Registration No.: 430374
Import-Export Code: 1109007027
Employer Identification Number (EIN):
USA Tax ID: 98-0673427

Global Journals Inc.

(A Delaware USA Incorporation with "Good Standing"; Reg. Number: 0423089)

Sponsors: Open Association of Research Society
Open Scientific Standards

Publisher's Headquarters office

Global Journals Headquarters
301st Edgewater Place Suite, 100 Edgewater Dr.-Pl,
Wakefield MASSACHUSETTS, Pin: 01880,
United States of America

USA Toll Free: +001-888-839-7392

USA Toll Free Fax: +001-888-839-7392

Offset Typesetting

Global Journals Incorporated
2nd, Lansdowne, Lansdowne Rd., Croydon-Surrey,
Pin: CR9 2ER, United Kingdom

Packaging & Continental Dispatching

Global Journals
E-3130 Sudama Nagar, Near Gopur Square,
Indore, M.P., Pin: 452009, India

Find a correspondence nodal officer near you

To find nodal officer of your country, please
email us at local@globaljournals.org

eContacts

Press Inquiries: press@globaljournals.org
Investor Inquiries: investors@globaljournals.org
Technical Support: technology@globaljournals.org
Media & Releases: media@globaljournals.org

Pricing (Including by Air Parcel Charges):

For Authors:

22 USD (B/W) & 50 USD (Color)

Yearly Subscription (Personal & Institutional):
200 USD (B/W) & 250 USD (Color)

INTEGRATED EDITORIAL BOARD
(COMPUTER SCIENCE, ENGINEERING, MEDICAL, MANAGEMENT, NATURAL
SCIENCE, SOCIAL SCIENCE)

John A. Hamilton, "Drew" Jr.,
Ph.D., Professor, Management
Computer Science and Software
Engineering
Director, Information Assurance
Laboratory
Auburn University

Dr. Henry Hexmoor
IEEE senior member since 2004
Ph.D. Computer Science, University at
Buffalo
Department of Computer Science
Southern Illinois University at Carbondale

Dr. Osman Balci, Professor
Department of Computer Science
Virginia Tech, Virginia University
Ph.D. and M.S. Syracuse University,
Syracuse, New York
M.S. and B.S. Bogazici University,
Istanbul, Turkey

Yogita Bajpai
M.Sc. (Computer Science), FICCT
U.S.A. Email:
yogita@computerresearch.org

Dr. T. David A. Forbes
Associate Professor and Range
Nutritionist
Ph.D. Edinburgh University - Animal
Nutrition
M.S. Aberdeen University - Animal
Nutrition
B.A. University of Dublin- Zoology

Dr. Wenying Feng
Professor, Department of Computing &
Information Systems
Department of Mathematics
Trent University, Peterborough,
ON Canada K9J 7B8

Dr. Thomas Wischgoll
Computer Science and Engineering,
Wright State University, Dayton, Ohio
B.S., M.S., Ph.D.
(University of Kaiserslautern)

Dr. Abdurrahman Arslanyilmaz
Computer Science & Information Systems
Department
Youngstown State University
Ph.D., Texas A&M University
University of Missouri, Columbia
Gazi University, Turkey

Dr. Xiaohong He
Professor of International Business
University of Quinnipiac
BS, Jilin Institute of Technology; MA, MS,
PhD,. (University of Texas-Dallas)

Burcin Becerik-Gerber
University of Southern California
Ph.D. in Civil Engineering
DDes from Harvard University
M.S. from University of California, Berkeley
& Istanbul University

Dr. Bart Lambrecht

Director of Research in Accounting and Finance
Professor of Finance
Lancaster University Management School
BA (Antwerp); MPhil, MA, PhD
(Cambridge)

Dr. Carlos García Pont

Associate Professor of Marketing
IESE Business School, University of Navarra
Doctor of Philosophy (Management),
Massachusetts Institute of Technology (MIT)
Master in Business Administration, IESE,
University of Navarra
Degree in Industrial Engineering,
Universitat Politècnica de Catalunya

Dr. Fotini Labropulu

Mathematics - Luther College
University of Regina
Ph.D., M.Sc. in Mathematics
B.A. (Honors) in Mathematics
University of Windsor

Dr. Lynn Lim

Reader in Business and Marketing
Roehampton University, London
BCom, PGDip, MBA (Distinction), PhD,
FHEA

Dr. Mihaly Mezei

ASSOCIATE PROFESSOR
Department of Structural and Chemical
Biology, Mount Sinai School of Medical
Center
Ph.D., Eötvös Loránd University
Postdoctoral Training,
New York University

Dr. Söhnke M. Bartram

Department of Accounting and Finance
Lancaster University Management School
Ph.D. (WHU Koblenz)
MBA/BBA (University of Saarbrücken)

Dr. Miguel Angel Ariño

Professor of Decision Sciences
IESE Business School
Barcelona, Spain (Universidad de Navarra)
CEIBS (China Europe International Business School).
Beijing, Shanghai and Shenzhen
Ph.D. in Mathematics
University of Barcelona
BA in Mathematics (Licenciatura)
University of Barcelona

Philip G. Moscoso

Technology and Operations Management
IESE Business School, University of Navarra
Ph.D in Industrial Engineering and
Management, ETH Zurich
M.Sc. in Chemical Engineering, ETH Zurich

Dr. Sanjay Dixit, M.D.

Director, EP Laboratories, Philadelphia VA
Medical Center
Cardiovascular Medicine - Cardiac
Arrhythmia
Univ of Penn School of Medicine

Dr. Han-Xiang Deng

MD., Ph.D
Associate Professor and Research
Department Division of Neuromuscular
Medicine
Davee Department of Neurology and Clinical
Neuroscience
Northwestern University
Feinberg School of Medicine

Dr. Pina C. Sanelli

Associate Professor of Public Health
Weill Cornell Medical College
Associate Attending Radiologist
NewYork-Presbyterian Hospital
MRI, MRA, CT, and CTA
Neuroradiology and Diagnostic
Radiology
M.D., State University of New York at
Buffalo, School of Medicine and
Biomedical Sciences

Dr. Roberto Sanchez

Associate Professor
Department of Structural and Chemical
Biology
Mount Sinai School of Medicine
Ph.D., The Rockefeller University

Dr. Wen-Yih Sun

Professor of Earth and Atmospheric
SciencesPurdue University Director
National Center for Typhoon and
Flooding Research, Taiwan
University Chair Professor
Department of Atmospheric Sciences,
National Central University, Chung-Li,
TaiwanUniversity Chair Professor
Institute of Environmental Engineering,
National Chiao Tung University, Hsin-
chu, Taiwan.Ph.D., MS The University of
Chicago, Geophysical Sciences
BS National Taiwan University,
Atmospheric Sciences
Associate Professor of Radiology

Dr. Michael R. Rudnick

M.D., FACP
Associate Professor of Medicine
Chief, Renal Electrolyte and
Hypertension Division (PMC)
Penn Medicine, University of
Pennsylvania
Presbyterian Medical Center,
Philadelphia
Nephrology and Internal Medicine
Certified by the American Board of
Internal Medicine

Dr. Bassey Benjamin Esu

B.Sc. Marketing; MBA Marketing; Ph.D
Marketing
Lecturer, Department of Marketing,
University of Calabar
Tourism Consultant, Cross River State
Tourism Development Department
Co-ordinator , Sustainable Tourism
Initiative, Calabar, Nigeria

Dr. Aziz M. Barbar, Ph.D.

IEEE Senior Member
Chairperson, Department of Computer
Science
AUST - American University of Science &
Technology
Alfred Naccash Avenue – Ashrafieh

PRESIDENT EDITOR (HON.)

Dr. George Perry, (Neuroscientist)

Dean and Professor, College of Sciences

Denham Harman Research Award (American Aging Association)

ISI Highly Cited Researcher, Iberoamerican Molecular Biology Organization

AAAS Fellow, Correspondent Member of Spanish Royal Academy of Sciences

University of Texas at San Antonio

Postdoctoral Fellow (Department of Cell Biology)

Baylor College of Medicine

Houston, Texas, United States

CHIEF AUTHOR (HON.)

Dr. R.K. Dixit

M.Sc., Ph.D., FICCT

Chief Author, India

Email: authorind@computerresearch.org

DEAN & EDITOR-IN-CHIEF (HON.)

Vivek Dubey(HON.)

MS (Industrial Engineering),

MS (Mechanical Engineering)

University of Wisconsin, FICCT

Editor-in-Chief, USA

editorusa@computerresearch.org

Sangita Dixit

M.Sc., FICCT

Dean & Chancellor (Asia Pacific)

deanind@computerresearch.org

Suyash Dixit

(B.E., Computer Science Engineering), FICCTT

President, Web Administration and

Development , CEO at IOSRD

COO at GAOR & OSS

Er. Suyog Dixit

(M. Tech), BE (HONS. in CSE), FICCT

SAP Certified Consultant

CEO at IOSRD, GAOR & OSS

Technical Dean, Global Journals Inc. (US)

Website: www.suyogdixit.com

Email: suyog@suyogdixit.com

Pritesh Rajvaidya

(MS) Computer Science Department

California State University

BE (Computer Science), FICCT

Technical Dean, USA

Email: pritesh@computerresearch.org

Luis Galárraga

J!Research Project Leader

Saarbrücken, Germany

CONTENTS OF THE ISSUE

- i. Copyright Notice
- ii. Editorial Board Members
- iii. Chief Author and Dean
- iv. Contents of the Issue

- 1. Design of a Five Stage Pipeline CPU with Interruption System. *1-32*
- 2. Design and Implementation of Data Scrambler & Descrambler System using VHDL. *33-38*
- 3. The Implementation of DMA Controller on Navigation baseband SoC. *39-42*
- 4. GSM based Agriculture Monitoring and Controlling System. *43-45*

- v. Fellows
- vi. Auxiliary Memberships
- vii. Process of Submission of Research Paper
- viii. Preferred Author Guidelines
- ix. Index



Design of a Five Stage Pipeline CPU with Interruption System

By Abdulraqueb Abdullah Saeed Abdo & Professor. Liu Yijun

Guangdong University of Technology, China

Abstract- A central processing unit (CPU), also referred to as a central processor unit, is the hardware within a computer that carries out the instructions of a computer program by performing the basic arithmetical, logical, and input/output operations of the system. The term has been in use in the computer industry at least since the early 1960s. The form, design, and implementation of CPUs have changed over the course of their history, but their fundamental operation remains much the same. A computer can have more than one CPU; this is called multiprocessing. All modern CPUs are microprocessors, meaning contained on a single chip. Some integrated circuits (ICs) can contain multiple CPUs on a single chip; those ICs are called multi-core processors. An IC containing a CPU can also contain peripheral devices, and other components of a computer system; this is called a system on a chip (SoC). Two typical components of a CPU are the arithmetic logic unit (ALU), which performs arithmetic and logical operations, and the control unit (CU), which extracts instructions from memory and decodes and executes them, calling on the ALU when necessary. Not all computational systems rely on a central processing unit. An array processor or vector processor has multiple parallel computing elements, with no one unit considered the "center". In the distributed computing model, problems are solved by a distributed interconnected set of processors.

Keywords: CPU; MIPS; pipeline; Interruption.

GJCST-A Classification : B.2.1 B.5.1



Strictly as per the compliance and regulations of:



Design of a Five Stage Pipeline CPU with Interruption System

Abdulraqueb Abdullah Saeed Abdo ^α & Professor. Liu Yijun ^σ

Abstract- A central processing unit (CPU), also referred to as a central processor unit, is the hardware within a computer that carries out the instructions of a computer program by performing the basic arithmetical, logical, and input/output operations of the system. The term has been in use in the computer industry at least since the early 1960s. The form, design, and implementation of CPUs have changed over the course of their history, but their fundamental operation remains much the same. A computer can have more than one CPU; this is called multiprocessing. All modern CPUs are microprocessors, meaning contained on a single chip. Some integrated circuits (ICs) can contain multiple CPUs on a single chip; those ICs are called multi-core processors. An IC containing a CPU can also contain peripheral devices, and other components of a computer system; this is called a system on a chip (SoC). Two typical components of a CPU are the arithmetic logic unit (ALU), which performs arithmetic and logical operations, and the control unit (CU), which extracts instructions from memory and decodes and executes them, calling on the ALU when necessary. Not all computational systems rely on a central processing unit. An array processor or vector processor has multiple parallel computing elements, with no one unit considered the "center". In the distributed computing model, problems are solved by a distributed interconnected set of processors.

In this paper, firstly I introduce the development of CPU and the background of this paper. On the foundation of that I explicitly introduce the architecture of RISC CPU and MIPS CPU which based on RISC architecture, paving the way for the design of my paper. And then I discuss the design of a five stage pipeline CPU based on MIPS instruction. The CPU in this paper mainly includes pipeline module, control module, interruption module and RAM/ROM module. Using EDA verification software Modelsim to verify the design on functional level and gate level. Finally I download the design to a development-board based on Altera Cyclone4 FPGA. The result of the verification shows that all functions can be achieved.

Keywords: CPU; MIPS; pipeline; Interruption.

摘要-

中央处理器广义上指一系列可以执行复杂的计算机程序的逻辑机器。这个空泛的定义很容易地将在“CPU”这个名称被普遍使用，之前的早期计算机也包括在内。无论如何，至少从20世纪60年代早期开始(Weik1961)，这个名称及其缩写已开始电子计算机产业中得到广泛应用。尽管与早期相比，“中央处理器”在物理形态、设计制造和具体任务的执行上有了戏剧性的发展，但是其基本的操作原理一直没有改变。早期的中央处理器通常是大型及特定应用的计算机而定制。但是，这种昂贵的为特定应用定制CPU的方法很大程度上已经让

位于开发便宜、标准化、适用于一个或多个目的的处理器的类。这个标准化趋势始于由单个晶体管组成的大型机和微机年代，随着集成电路的出现而加速。IC使得更为复杂的CPU可以在很小的空间中设计和制造（在微米的量级）。CPU的标准化和小型化都使得这一类数字设备在现代生活中的出现频率远远超过有限应用专用的计算机。现代微处理器出现在包括从汽车到手机到儿童玩具在内的各种物品中。

论文首先介绍了中央处理器发展的历史，以及本文设计的研究背景，并在此基础上着重介绍了精简指令RISCCPU的结构以及基于RISC结构的MIPSCPU的有关背景资料，为论文后续的设计做好铺垫。接着详细介绍了一款基于MIPS指令集的5级流水线CPU的设计。本CPU主要包括流水线模块，控制模块，中断处理模块，以及ROM和RAM模块。使用EDA验证软件Modelsim对设计进行了功能仿真和门级仿真，最后将设计下载到了基于Altera Cyclone4 FPGA 的开发板上。进行了验证验证结果表明本设计能实现所有功能。

关键词: CPU; MIPS; 流水线; 中断

1. INTRODUCTION

a) Research status of CPU design and trend

CPU is one of the main devices of a computer. Its main function is to explain computer's instruction and deal with the data of software. The programmable ability of computer generally means to program CPU. Central process unit, inner memory and input/output device are three core components of modern computer. Before 1970s, CPU is composed of several individual units. Later the CPU manufactured by semiconductor was developed. The complex circuits of a microprocessor can be made as a tiny unit with powerful function.

Central processor broadly means a series of logic machines that can perform complex computer programs. The term has been in use in the computer industry at least since the early 1960s. The form, design, and implementation of CPUs have changed over the course of their history, but their fundamental operation remains much the same.

A computer can have more than one CPU; this is called multiprocessing. All modern CPUs are microprocessors, meaning contained on a single chip. Some integrated circuits (ICs) can contain multiple CPUs on a single chip; those ICs are called multi-core processors. An IC containing a CPU can also contain peripheral devices, and other components of a computer system; this is called a system on a chip (SoC).

Author ^α ^σ: School of Computer Science Guangdong University of Technology Guangzhou, Guangdong, P. R. China, 510006.
e-mail: 970133919@qq.com

Two typical components of a CPU are the arithmetic logic unit (ALU), which performs arithmetic and logical operations, and the control unit (CU), which extracts instructions from memory and decodes and executes them, calling on the ALU when necessary.

Not all computational systems rely on a central processing unit. An array processor or vector processor has multiple parallel computing elements, with no one unit considered the "center". In the distributed computing model, problems are solved by a distributed interconnected set of processors.

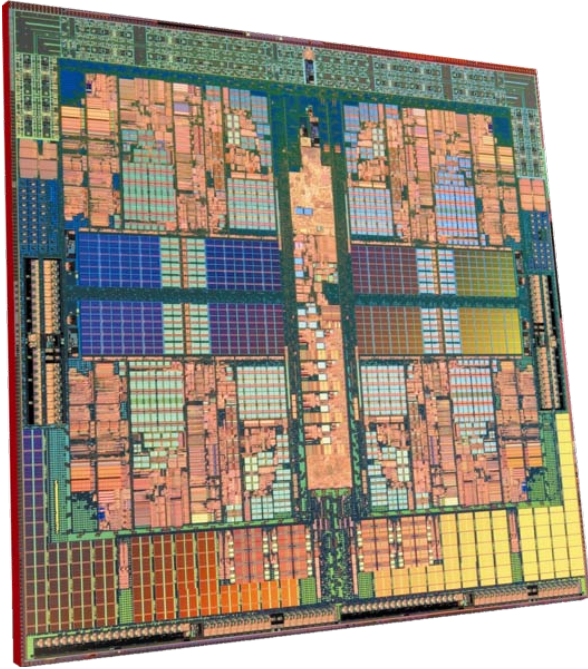


Figure 1-1 : Intel Phenom Quad-Core

Moore's Law makes us can expect the general situation in the future development of the CPU. Undoubtedly, high performance, low power

consumption, high speed and low cost are the future direction of development.

1. Smaller wiring width and more transistors

Nowadays, Intel's and AMD's CPUs have used 0.18 or even 0.13 micrometer technology. For current silicon chips, reducing the wiring width is the key to raising the speed of the CPU.

Experts predict that the design of monolithic integrated chip system will reach such a number of indicators - the minimum feature size reaches 0.1 micrometer, chip integration reaches 200 million transistors. And some breakthroughs are also made from the aspect of the production process. IBM has developed a new chip packaging technology, by which the chip manufacturers can use aluminum instead of the traditional copper wire connections to connect transistors on a chip. Since copper conductors can be made thinner than the aluminum wire, so that the chip can be integrated on a larger number of transistors, which makes the packaging unit of the computing power has been greatly improved. Copper processor chip has become the future direction of development. Researches of copper chip have been performed by many chipmakers such as Intel and AMD.

2. 64-bit CPU chip manufacturers dominate the market

With the release of Intel Itanium, personal PC processor market will also be transited into 64-bit. 64-bit CPU can handle 64-bit data and 64-bit addresses and can provide higher accuracy and larger memory addressing range.

3. Higher bus speeds

Nowadays the bus has increasingly limited the performance of CPU. For which various manufacturers are seeking ways to improve bus speed. It's expected that within three years Bus speed should be able to exceed 1GHz.

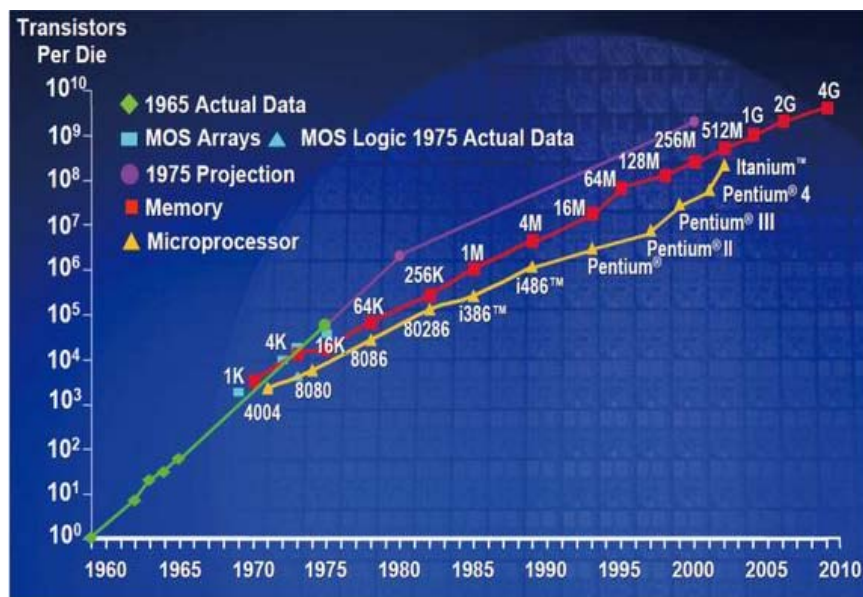


Figure 1-2 : Moore's Law and the development of IC integration

b) RISC and Pipeline processing

Reduced Instruction Set Computing is a design pattern for computer central processor. This design idea has reduced the number of instructions and addressing modes, making implementation easier, higher instructions parallelism, and a more efficient compiler. Current common RISC microprocessors includes DEC Alpha, ARC, ARM, AVR, MIPS, PA-RISC, Power Architecture (PowerPC, PowerXCell) and SPARC.

From the earliest, RISC's name comes from the Berkeley RISC project held by David Patterson in the University of California, Berkeley. But before him, people has been proposed a similar design philosophy. IBM 801 project, held by John Cork, started in 1975 and finished in 1980, probably is the first system designed under the concept of reduced instruction set. This design concept originated from the discovery that although many of the features of traditional processor are designed to make the code easier to write, but these complex features require several cycles to achieve and often are not used by the program. In addition, the difference between the speed of the processor and the main memory has become increasingly bigger. Prompted by these factors, a series of new technologies were introduced, making the processor's instruction executed in pipeline while reducing the number of processor memory access.

In the early period, Characteristics of such an instruction set is the small number of instructions, each instruction word in standard length, short execution time, and implementation details of the central processor for the machine-level program is visible and so. In fact in the later development, RISC and CISC have learned each other during the process of competitions. Now the RISC instruction set also has reached hundreds and operating cycle are no longer fixed. Nonetheless, fundamental principles of RISC design - optimization for pipelined processor – have not changed yet. And following this principle, a concurrent variant of RISC is developed – named VLIW – combining the short and length unity instructions into very long instruction. Each time you run a very long instruction, equal to concurrently run multiple short instructions.

On the other hand, the most common complex instruction set x86 CPU, although the instruction set is CISC, but it will make every effort to accelerate the hardware circuit to control commonly used simple instructions. Complex instruction which is not used often will be given to micro-code sequencer to “decode slowly and run slowly”. Hence it's called "RISCy x86".

RISC processor should be designed to not only make effective execution pipeline processing, but also enable optimizing compiler optimized instruction generated code. Below, we will describe RISC processor design principles and techniques.

1. Efficient pipelining

The relation in pipeline means because there is some association in adjacent or similar instruction, later instruction cannot be executed within originally designated clock cycle. In general, the pipeline relation is divided into the following three types.

1. Data hazards Data hazards occur when instructions that exhibit data dependence modify data in different stages of a pipeline. Ignoring potential data hazards can result in race conditions (sometimes known as race hazards).
2. Structural hazards A structural hazard occurs when a part of the processor's hardware is needed by two or more instructions at the same time. A canonical example is a single memory unit that is accessed both in the fetch stage where an instruction is retrieved from memory, and the memory stage where data is written and/or read from memory.[3] They can often be resolved by separating the component into orthogonal units (such as separate caches) or bubbling the pipeline.
3. Control hazards (branch hazards) Branching hazards (also known as control hazards) occur with branches. On many instruction pipeline micro-architectures, the processor will not know the outcome of the branch when it needs to insert a new instruction into the pipeline (normally the fetch stage).

There are several methods used to deal with hazards, including pipeline stalls/pipeline bubbling, register forwarding, and in the case of out-of-order execution, the scoreboarding method and the Tomasulo algorithm.

2. Short cycle time

To increase the clock frequency by optimizing the process. To optimize circuit design structure, reduce instruction fetching time and read/write latency, thus reducing instruction period, which can greatly improve the efficiency of the machine.

3. Load/Store Structure

Load/Store Structure is used to transfer data between register file and memory. Load is used to fetch data from memory, while store is used to store data into memory. These two instructions are used frequently and is the most significant one in the instruction set. Because the other instructions can only handle register file. When data is in the memory, you have to load the data into register file, and store the data back into it after execution. In the register file, you don't have to access the memory when data have to be used again. This Load/Store structure is the key for single period clock execution.

4. Simple fixed format instruction system

RISC designers focus on those frequently used commands, and try to make them simple and efficient features. For not commonly used functions we often

accomplished through a combination of instruction. Therefore, when implementing the special features on RISC machines, the efficiency may be lower, but you can use pipelining and superscalar techniques to improve and make up. While the CISC instruction set computer is rich, with special instructions to perform specific functions. Therefore, the efficiency of handling special tasks is higher.

5. No micro-code technology

Since RISC use Simple, rational and simplified instruction addressing modes, so it does not need micro-code technology, which means without micro-code ROM, but execute instruction directly in the hardware. This means eliminating the original machine microcode instructions into the intermediate step, and it reduce the number of machine cycles needed to execute an instruction. Also it saves space so that the chip can be saved using the microprocessor chip space expansion function.

6. Huge register file

A register file (register file) is an array consisting of a plurality of registers in the CPU, which usually realized by a fast static random access memory (SRAM). This RAM has a dedicated reading port and writing port, multiple concurrent accessing different registers. CPU's instruction set architecture is always defined a number of registers used for temporary storage of data between memory and CPU computing components. In a more simplified CPU, these architectures registers (architectural registers)-correspondence with the physical register within the CPU. In a more complex CPU, we use register renaming techniques, during the execution architecture which makes physical storage entry in the register which corresponds to the register file (physical entry stores) is dynamically changed. Register file is part of the instruction set architecture. The program can be accessed, which is transparent to the CPU cache (cache) different.

7. Harvard bus architecture

Harvard architecture is a memory structure to store program instructions and data separately. First, the CPU read program instruction in the program instruction memory. And then it gets the data address after decoding. Then it reads data in the according data memory, finally handle next execution (usually instruction). Instruction store and data store is separated, while Storing data and instructions can be simultaneously. Data and instruction can have different data width. For example, Microchip's PIC16's program instruction is 14 bit width, while data is 8 bit width. Harvard bus architecture CPU usually has relatively high execution efficient. Program instructions and data organization and storage instructions apart, implementation can be pre fetch the next instruction.

8. Delayed branch

Insert one or several effective instruction in the branch instruction. When the program is executed, after these into the instruction execution is completed, then executes the instruction, therefore, transfer instruction seems to be delayed, this technique known as delayed transfer of technology.

9. Hard-wired controller

Once control unit was build, unless redesigned and remapping, it's impossible to add new functions. Hard-wired controller is one of the most complex logic components in the CPU. When executes different machine instructions, it decodes the instruction through activates a series of different control signals, making the control unit has few explicit structure and in a mess. Since that, hard-wired controller is replaced by micro-program controller. However, under the same semiconductor process, hard-wired controller is faster than micro-program controller.

10. Assembly technology optimization

11. High-level programming language oriented

c) *Structure and content*

This paper describes the design of a five-stage pipeline CPU with interruption system. Including CPU's research background, instruction set, pipeline data path and the design of interruption and exception system. And we use EDA tools for the simulation of the design. Finally we proof that the design meets the performance requirement.

The chapters are arranged as follows

Chapter one is the brief introduction of the research background. It mainly introduces the background and related research status and CPU's integrated circuit industry.

Chapter two is the brief introduction of the development platform and MIPS architecture. It mainly introduces the software and hardware development platform for the project and FPGA's design flow. It also introduces MIPS architecture.

Chapter three describes the design of pipeline data path. It introduces the pipeline design method, the composition of the pipeline and design and verification of associated component.

Chapter four describes the design of interruption and exception circuits. It introduces the principal of exception circuits and verification of related components.

Chapter five is CPU functional verification.

II. DEVELOPMENT PLATFORM AND MIPS ARCHITECTURE

This chapter mainly introduces the development platform of this paper – the EDA development and verification system based on Altera Cyclone4 FPGA and

Quartus + Modelsim. Then we will introduce the background of our design – MIPS instruction set and architecture.

a) *Technology for CPU hardware design and implementation*

i. *Hardware description language*

In electronics, a hardware description language or HDL is a specialized computer language used to program the structure, design and operation of electronic circuits, and most commonly, digital logic circuits.

A hardware description language enables a precise, formal description of an electronic circuit that allows for the automated analysis, simulation, and simulated testing of an electronic circuit. It also allows for the compilation of an HDL program into a lower level specification of physical electronic components, such as the set of masks used to create an integrated circuit.

A hardware description language looks much like a programming language such as C; it is a textual description consisting of expressions, statements and control structures. One important difference between most programming languages and HDLs is that HDLs explicitly include the notion of time.

HDLs form an integral part of Electronic design automation systems, especially for complex circuits, such as microprocessors.

ii. *Structure of HDL*

HDLs are standard text-based expressions of the spatial and temporal structure and behavior of electronic systems. Like concurrent programming languages, HDL syntax and semantics include explicit notations for expressing concurrency. However, in contrast to most software programming languages, HDLs also include an explicit notion of time, which is a primary attribute of hardware. Languages whose only characteristic is to express circuit connectivity between a hierarchy of blocks are properly classified as netlist languages used in electric computer-aided design (CAD). HDL can be used to express designs in structural, behavioral or register-transfer-level architectures for the same circuit functionality; in the latter two cases the synthesizer decides the architecture and logic gate layout.

HDLs are used to write executable specifications for hardware. A program designed to implement the underlying semantics of the language statements and simulate the progress of time provides the hardware designer with the ability to model a piece of hardware before it is created physically. It is this excitability that gives HDLs the illusion of being programming languages, when they are more precisely classified as specification languages or modeling languages. Simulators capable of supporting discrete-event (digital) and continuous-time (analog) modeling exist, and HDLs targeted for each are available.

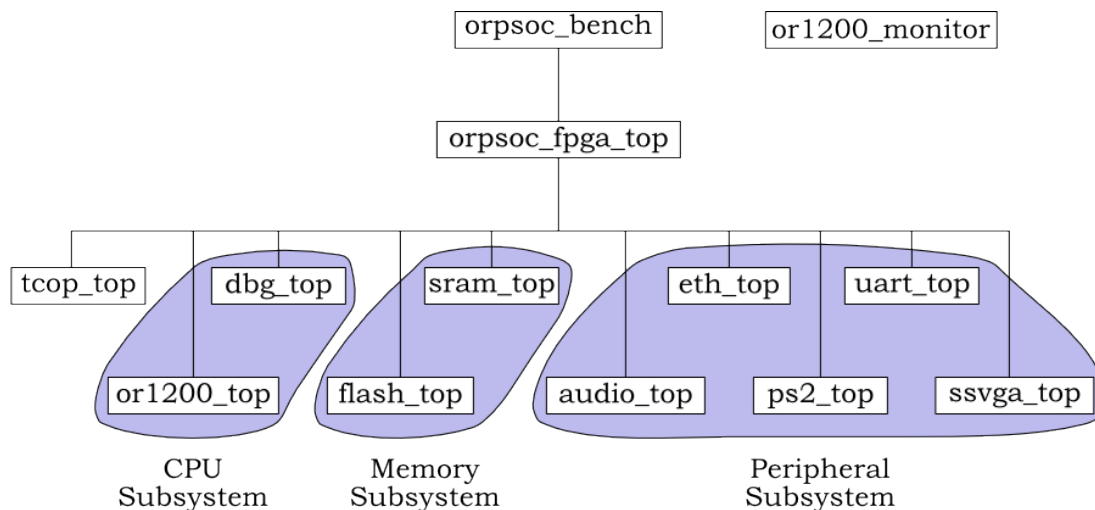


Figure 2-1 : an example of Verilog HDL hierarchy

iii. *Comparison with control-flow languages*

It is certainly possible to represent hardware semantics using traditional programming languages, which operate on control flow semantics as opposed to data flow, such as C++, although to function as such, programs must be augmented with extensive and unwieldy class libraries. Primarily, however, software programming languages do not include any capability for explicitly expressing time, and this is why they cannot function as hardware description languages. Before the

recent introduction of SystemVerilog, C++ integration with a logic simulator was one of the few ways to use OOP in hardware verification. SystemVerilog is the first major HDL to offer object orientation and garbage collection.

Using the proper subset of hardware description language, a program called a synthesizer (or synthesis tool) can infer hardware logic operations from the language statements and produce an equivalent netlist of generic hardware primitives to implement the

specified behavior. Synthesizers generally ignore the expression of any timing constructs in the text. Digital logic synthesizers, for example, generally use clock edges as the way to time the circuit, ignoring any timing constructs. The ability to have a synthesizable subset of the language does not itself make a hardware description language.

iv. *Design using HDL*

As a result of the efficiency gains realized using HDL, a majority of modern digital circuit design revolves around it. Most designs begin as a set of requirements or a high-level architectural diagram. Control and decision structures are often prototyped in flowchart applications, or entered in a state-diagram editor. The process of writing the HDL description is highly dependent on the nature of the circuit and the designer's preference for coding style. The HDL is merely the 'capture language,' often beginning with a high-level algorithmic description such as a C++ mathematical model. Designers often use scripting languages (such as Perl) to automatically generate repetitive circuit structures in the HDL language. Special text editors offer features for automatic indentation, syntax-dependent coloration, and macro-based expansion of entity/architecture/signal declaration.

The HDL code then undergoes a code review, or auditing. In preparation for synthesis, the HDL description is subject to an array of automated checkers. The checkers report deviations from standardized code guidelines, identify potential ambiguous code constructs before they can cause misinterpretation, and check for common logical coding errors, such as dangling ports or shorted outputs. This process aids in resolving errors before the code is synthesized.

In industry parlance, HDL design generally ends at the synthesis stage. Once the synthesis tool has mapped the HDL description into a gate netlist, this netlist is passed off to the back-end stage. Depending on the physical technology (FPGA, ASIC gate array, ASIC standard cell), HDLs may or may not play a significant role in the back-end flow. In general, as the design flow progresses toward a physically realizable form, the design database becomes progressively more laden with technology-specific information, which cannot be stored in a generic HDL description. Finally, an integrated circuit is manufactured or programmed for use.

v. *Simulating and debugging HDL code*

Essential to HDL design is the ability to simulate HDL programs. Simulation allows an HDL description of a design (called a model) to pass design verification, an important milestone that validates the design's intended function (specification) against the code implementation in the HDL description. It also permits architectural exploration. The engineer can experiment with design

choices by writing multiple variations of a base design, then comparing their behavior in simulation. Thus, simulation is critical for successful HDL design.

To simulate an HDL model, an engineer writes a top-level simulation environment (called a testbench). At minimum, a testbench contains an instantiation of the model (called the device under test or DUT), pin/signal declarations for the model's I/O, and a clock waveform. The testbench code is event driven: the engineer writes HDL statements to implement the (testbench-generated) reset-signal, to model interface transactions (such as a host-bus read/write), and to monitor the DUT's output. An HDL simulator — the program that executes the testbench — maintains the simulator clock, which is the master reference for all events in the testbench simulation. Events occur only at the instants dictated by the testbench HDL (such as a reset-toggle coded into the testbench), or in reaction (by the model) to stimulus and triggering events. Modern HDL simulators have full-featured graphical user interfaces, complete with a suite of debug tools. These allow the user to stop and restart the simulation at any time, insert simulator breakpoints (independent of the HDL code), and monitor or modify any element in the HDL model hierarchy. Modern simulators can also link the HDL environment to user-compiled libraries, through a defined PLI/VHPI interface. Linking is system-dependent (Win32/Linux/SPARC), as the HDL simulator and user libraries are compiled and linked outside the HDL environment.

Design verification is often the most time-consuming portion of the design process, due to the disconnect between a device's functional specification, the designer's interpretation of the specification, and the imprecision of the HDL language. The majority of the initial test/debug cycle is conducted in the HDL simulator environment, as the early stage of the design is subject to frequent and major circuit changes. An HDL description can also be prototyped and tested in hardware — programmable logic devices are often used for this purpose. Hardware prototyping is comparatively more expensive than HDL simulation, but offers a real-world view of the design. Prototyping is the best way to check interfacing against other hardware devices and hardware prototypes. Even those running on slow FPGAs offer much shorter simulation times than pure HDL simulation.

b) *EDA system*

i. *QuartusII*

Quartus II is a software tool produced by Altera for analysis and synthesis of HDL designs, which enables the developer to compile their designs, perform timing analysis, examine RTL diagrams, simulate a design's reaction to different stimuli, and configure the target device with the programmer. The latest version is 13sp1 which is a service pack of version 13.

ii. Modelsim

Mentor Graphics was the first to combine single kernel simulator (SKS) technology with a unified debug environment for Verilog, VHDL, and SystemC. The combination of industry-leading, native SKS

performance with the best integrated debug and analysis environment make ModelSim® the simulator of choice for both ASIC and FPGA designs. The best standards and platform support in the industry make it easy to adopt in the majority of process and tool flows.

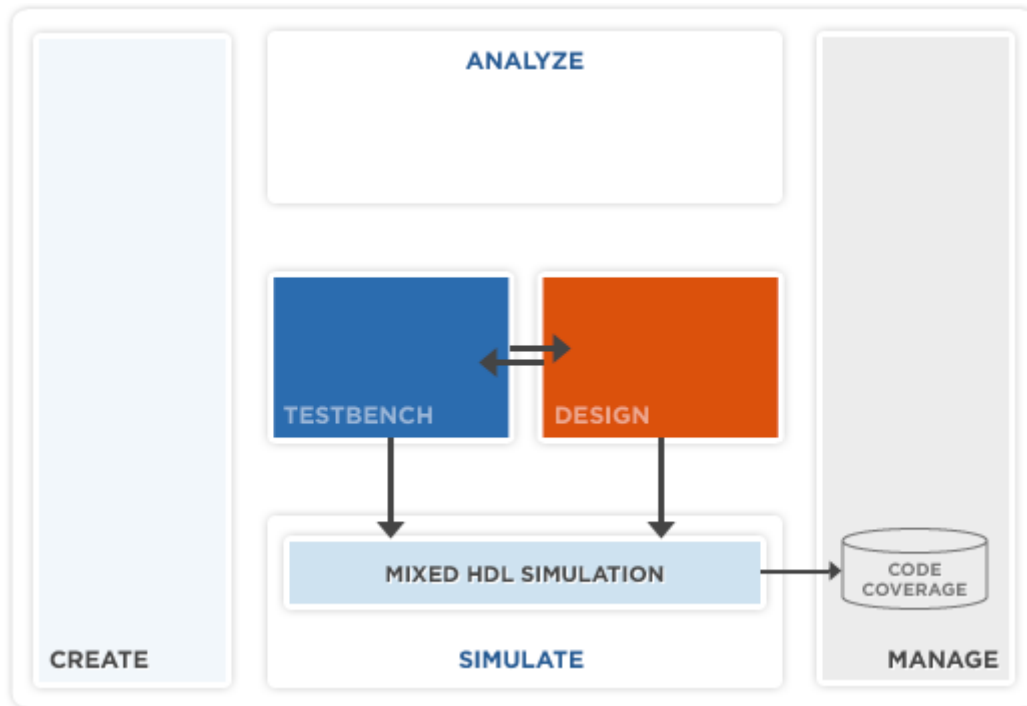


Figure 2-2 : Modelsim simulation structure

c) FPGA design and verification

Field-programmable gate array (FPGA) is a device that has numerous gate (switch) arrays and can be programmed on-board through dedicated Joint Test Action Group (JTAG) or on-board devices or using remote system through Peripheral Component Interconnect Express (PCIe), Ethernet, etc. FPGAs are based on static random-access memory (SRAM). The contents of the memory of an FPGA erase once the power is turned off. Usually, FPGAs can be programmed several thousands of times without the device getting faulty.

Fig. 2-3 shows the architecture of an FPGA. It includes logic blocks, input/output (I/O) cells, phase-locked loops/delay-locked loops (PLLs/DLLs), block RAM and interconnecting matrix. Nowadays, FPGAs are also coming up with several hard intellectual property (IP) blocks of PCIe, Ethernet, Rocket I/O, PHYs for DDR3 interfaces and processor cores (for example, PowerPC in Xilinx Virtex-5 FPGA and ARM cores in both Xilinx and Altera series FPGAs).

To level up with the new technology, both Xilinx and Altera have come up with new series of FPGAs (Virtex 7 from Xilinx and Stratix-V from Altera), which are manufactured with TSMC's 28nm silicon technology. These FPGAs focus on a high speed with low power consumption using various parameters and bringing

down the FPGA core voltage to as low as 0.9V. Along with the new FPGAs, Xilinx and Altera are also focused on improving their synthesis tools to meet the routing constraints and to analyze the timing and power consumption of the FPGA.

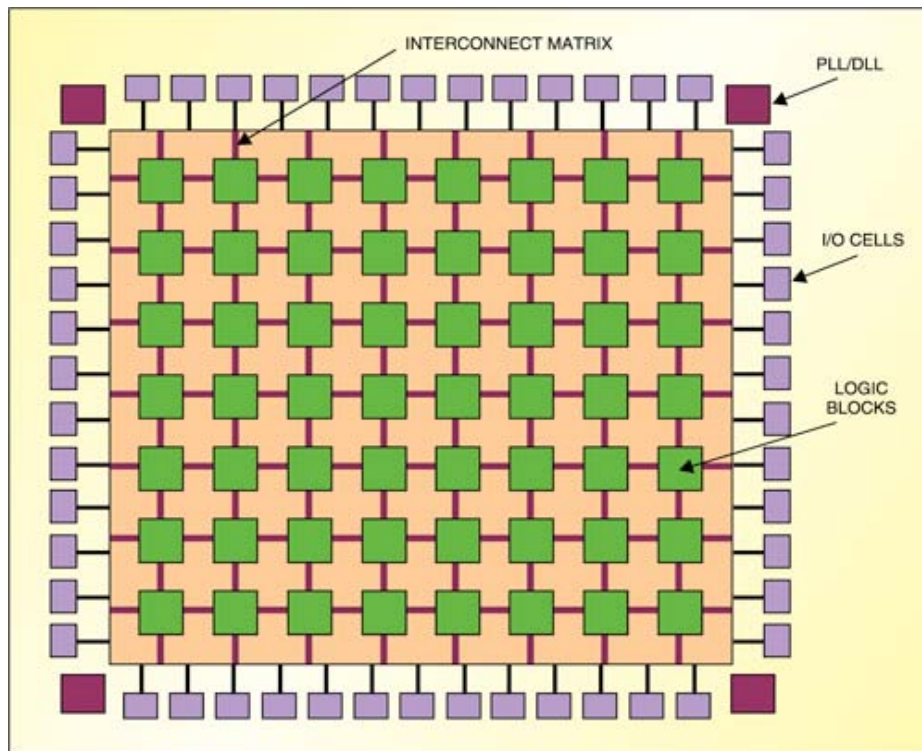


Figure 2-3 : FPGA architecture

As the aim here is to learn the basic technique of FPGA design to work with both the tools and devices, let's get back to the design flow through the steps.

Step 1: Requirement analysis and SRS preparation

Before starting work on the design, all requirements should be documented as system requirement specification (SRS) by designers and approved by various levels in the organization, and most importantly, the client. During this phase, FPGA designers, along with the hardware team, should identify suitable FPGAs for the project. This is very important because designers need to know parameters such as the I/O voltage levels, operating frequency and external peripheral interfaces.

It is also important to determine which IP cores are available with the tools or FPGA family used for the project. Some IP cores are free, while others are licensed and paid for. This cost should be reviewed several times by the team before releasing it to the client and listed separately for approval from the client or management.

The SRS should contain the following (the list pertains to the FPGA only):

1. Aim of the project
2. Functionalities to be handled by the design, followed by a short description
3. A concept-level block diagram depicting the major internal peripherals/IPs of the FPGA
4. FPGA vendor, family, speed grade, package, core voltage, supported I/O levels, commercial/industrial type

5. List of blocks that will be used as IPs. Mention clearly what's available for free with the vendor-provided IPs, hard IPs available within the FPGA and paid licensed IPs to be used
6. Type of processor interfaces used (soft processor or external processor interfaces)
7. Type of memory interfaces used
8. A section about the timing diagram of the major peripheral interfaces such as the processor interface and flash interface.
9. Type of FPGA configurations to be used
10. Reset and clock interface planned
11. A brief summary of the estimated resources required for implementation of the logic and I/O pins to be used
12. HDL (VHDL, Verilog, 'C' or mixed) used for RTL coding, tools and version to be used for synthesis, implementation and simulation

To calculate the approximate resources required, go through the IP datasheets for the resources used for each IP, and also calculate the resources used by custom RTL. There is no rule of thumb for calculating resources at this level. These can be calculated approximately based on experience, reviews or analysis. The most important thing is to get the resource requirement reviewed by the hardware team, software team and a third party several times before submitting it to the client.

Step 2: Detailed design document preparation

Once the SRS is approved by the client, the next phase is to make the detailed design document.

This document should consist of:

1. Brief introduction to the project
2. FPGA part details with proper specification
3. Detailed block diagram depicting the internal modules of the FPGA design
4. Top-level module block diagram showing input and output ports with their active levels and voltage levels which are connected to the external peripherals, connectors and debug points
5. Hierarchical tree of the modules
6. Each module should have:
 - i. Detailed explanation of the functionality
 - ii. Register information
 - iii. List of input and output ports with source and destination module name, and active level of the signal
 - iv. A block diagram/digital circuit diagram of finite-state machines indicating how the RTL will be implemented
 - v. Clock frequency to be used, if a synchronous module is used
 - vi. Reset logic implementation
 - vii. File name which will be implemented
 - viii. Approximate FPGA resource utilization
 - ix. Testbench for testing each module independently
7. Input system clock frequency and reset level
8. Explanation of how the internal clock frequencies are derived—using phase-locked loop (PLL) or delay-locked loop (DLL) with the input clock. Also, explain how the global clock buffers are used. Mention clock signals with their frequency and voltage levels that are driven out of the FPGA for external peripherals.
9. A simulation environment setup for the design (called 'device under test') with a top-level testbench. A block diagram indicating how the clock source, reset and pattern generators, and bus functional modes are connected to the top-level module under testing will be helpful here. Mention how log files are used to register the activity of the required signal
10. Make a page with the heading 'FPGA Synthesis and Resource Utilization.' Keep it blank with a note that once the final implementation is done, this page will be updated
11. Under the heading 'Timing Analysis,' mention the major timing parameters of the control signals to be maintained, with a timing budget and waveform drawn manually or using timing analyzer tool. Mention the major timing constraints that will be used in the UCF or QSF files of the design

As mentioned in Step 1, the FPGA team members, hardware and software team members and architects should review the document at several stages before releasing it to the client.

Step 3: Design entry and functional simulation

Each module owner should develop a testbench for his module, capture simulated waveforms or assertion-based log report, and get it reviewed by the team lead. Before going for synthesis, every module should be verified thoroughly for functionality using simulation. Regular code review will help to reduce errors and simulation time. Once the simulation of individual modules is done, the next step is to integrate the module and do full-system-level functional simulation with assertion-based log report.

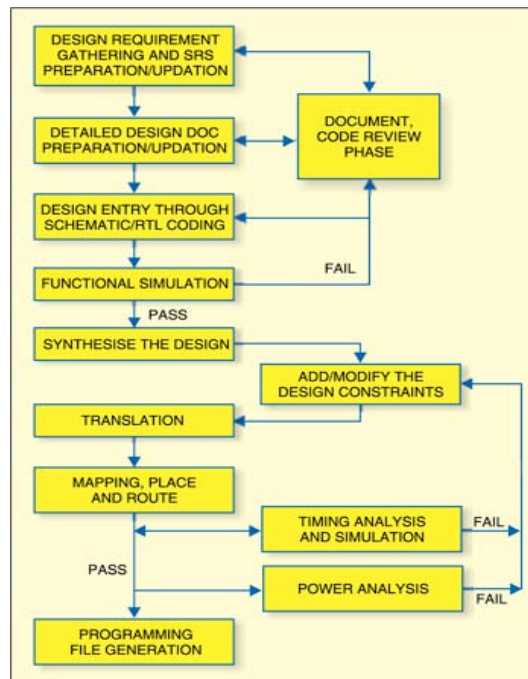


Figure 2-4 : FPGA design flowchart

Step 4: Synthesis

If the functional simulation satisfies the requirement, the next phase is synthesis.

In this phase, the integrated project is synthesized using a vendor-specific synthesis tool based on the optimization settings. Whenever RTL is modified, it is always good to complete Step 3 with unit-level and full-system functional simulation. Always follow vendor-specific coding guidelines and library modules for better optimization of the design.

During this phase, synthesis tools verify the design for syntax errors and do block-level floor planning.

Step 5: Adding design constraints

Once synthesis is complete, constraints can be added to the design. These constraints are usually included in a separate file where the designer lists out the signal with its corresponding FPGA pin number, I/O voltage levels, current-driving strength for output signals, input clock frequency, hard block or module location, timing paths to be ignored, false paths, other IP-specific constraints recommended by the vendor, etc. This information is passed on to the placement phase.

Step 6: Placement and routing phase

Before routing, the synthesis tool maps the buffers, memory and clock buffers to the specific vendor libraries. That is, in this phase, logical blocks are translated into physical file format. Then, in the place-and-route process, the tool places and routes the design considering the user constraints and optimization techniques. Timing simulation can be done at this stage to verify the functionality, so that the design meets all the functional and timing requirements.

Step 7: Programming file generation

After obtaining a satisfactory timing and functional behavior of the design, it is time to generate the bit file that is downloaded to the FPGA to test the functionality on the board with actual peripherals.

For each stage, the tool will provide the corresponding report files, using which the designer can analyse time delays, power, resource usage, unrouted signals and I/O pins list.

In short

To summarize the above points, the FPGA design flow is shown as a simple flow-charting Figure 2-3. There may be minor variations in the design flow during the requirement stage and the design and document preparation phase, from one organization or project to another, but the overall FPGA design flow remains the same.

d) MIPS architecture

MIPS (originally an acronym for Microprocessor without Interlocked Pipeline Stages) is a reduced instruction set computer (RISC) instruction set

architecture (ISA) developed by MIPS Technologies (formerly MIPS Computer Systems, Inc.). The early MIPS architectures were 32-bit, with 64-bit versions added later. Multiple revisions of the MIPS instruction set exist, including MIPS I, MIPS II, MIPS III, MIPS IV, MIPS V, MIPS32, and MIPS64. The current revisions are MIPS32 (for 32-bit implementations) and MIPS64 (for 64-bit implementations). MIPS32 and MIPS64 define a control register set as well as the instruction set.

i. MIPS classical five-stage pipeline

MIPS architecture is designed based on pipeline architecture. Every MIPS instruction is divided into five stages once it is fetched from cache and every stage takes stable time. Usually each stage takes one clock cycle, while RD/WB operation takes half clock cycle. The execution process of MIPS processor is divided into five stages as follow.

1. Instruction Fetch (IF) Stage**a. Instruction Fetch**

Instruction's address in PC is applied to instruction memory that causes the addressed instruction to become available at the output lines of instruction memory.

b. Updating PC

The address in PC is incremented by 4 but what is written in PC is determined by the control signal PCSrc. Depending upon the status of control signal PCSrc, PC is either written by the branch target address (BTA) or the sequential address (PC + 4).

2. Instruction Decode (ID) Stage

- a. Instruction is decoded by the control unit that takes 6-bit opcode and generates control signals.
- b. The control signals are buffered in the pipeline registers until they are used in the concerned stage by the corresponding instruction.
- c. Registers are also read in this stage. Note that the first source register's identifier in every instruction is at bit positions [25:21] and second source register's identifier (if any) is at bit positions [20:16].
- d. The destination register's identifier is either at bit positions [15:11] (for R-type) or at [20:16] (for lw and addi). The correct destination register's identifier is selected via multiplexer controlled by the control signal RegDst. However, this multiplexer is placed in the EX stage because the instruction decoding is not finished until the second stage is complete. But this identifier is buffered until the WB stage because an instruction write sa register in the WB stage.

3. Execution (EX) Stage

- a. This stage is marked by the use of ALU that performs the desired operation on registers(R-type), calculates address (memory reference instructions), or compares registers (branch).

- b. An ALU control accepts 6-bit function field and 2-bit control signal ALU Op to generate the required control signal for the ALU.
- c. BTA is also calculated in the EX stage by a separate adder
4. Memory (M) Stage
 - a. Data memory is read (lw) or written (sw) using the address calculated by the ALU in EX stage.
 - b. ZERO output of ALU and BRANCH signal generated by the control unit are ANDed to determine the fate of branch (taken or not taken).
5. Write Back (WB) Stage
 - a. Result produced by ALU in EX stage (R-type) or data read from data memory in M stage (lw) is written in destination register. The data to be written in destination register is selected via multiplexer controlled by the control signal MemToReg.

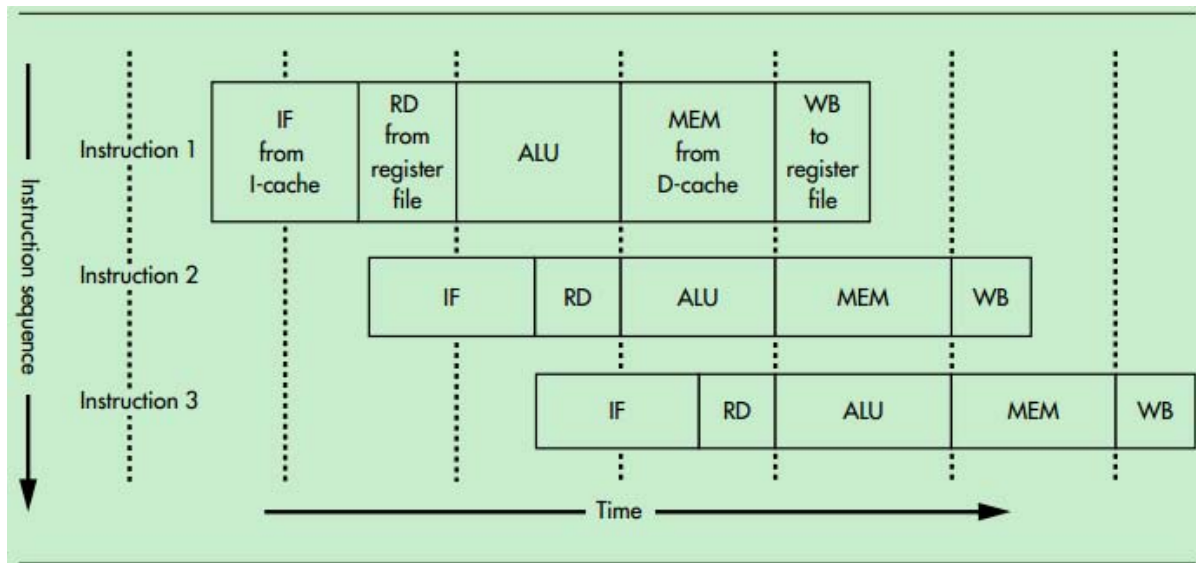


Figure 2-5 : MIPS five-stage pipeline

ii. MIPS register

MIPS have 32 common register (\$0-\$31). The Table 2-1 below describes the aliases and function of these 32 registers.

Table 2-1 : MIPS register

REGISTER	NAME	USAGE
\$0	\$zero	constant value 0
\$1	\$at	Reserved for assembler
\$2-\$3	\$v0-\$v1	values for results and expression evaluation
\$4-\$7	\$a0-\$a3	arguments
\$8-\$15	\$t0-\$t7	Temporary or random
\$16-\$23	\$s0-\$s7	saved
\$24-\$25	\$t8-\$t9	Temporary or random
\$28	\$gp	Global Pointer
\$29	\$sp	Stack Pointer
\$30	\$fp	Frame Pointer
\$31	\$ra	return address

iii. MIPS instruction set

Instructions are divided into three types: R, I and J. Every instruction starts with a 6-bit opcode. In addition to the opcode, R-type instructions specify three registers, a shift amount field, and a function field; I-type instructions specify two registers and a 16-bit immediate value; J-type instructions follow the opcode with a 26-bit jump target.

Since the MIPS instruction set involves many, not all will be used in our design, so only the selection and design-related instructions are described in this article.

Now the MIPS instruction used in this article are listed below, there are two main type of instruction - integer instruction and interrupt and exception handling instructions.

Table 2-2 : MIPS instruction set

integer instruction	meaning
add	add rd, rs, rt #rd $\leftarrow$ rs op rt
sub	sub rd, rs, rt #rd $\leftarrow$ rs op rt
and	and rd, rs, rt #rd $\leftarrow$ rs op rt
Or	or rd, rs, rt #rd $\leftarrow$ rs op rt
Xor	xor rd, rs, rt #rd $\leftarrow$ rs op rt
Sll	sll rd, rt, sa #rd $\leftarrow$ rt shift sa
Srl	srl rd, rt, sa #rd $\leftarrow$ rt shift sa
Sra	sra rd, rt, sa #rd $\leftarrow$ rt shift sa
Jr	jr rs #PC $\leftarrow$ rs
addi	addi rt, rs, imm #rt $\leftarrow$ rs + imm
andi	andi rt, rs, imm #rt $\leftarrow$ rs op imm
Ori	ori rt, rs, imm #rt $\leftarrow$ rs op imm
xori	xori rt, rs, imm #rt $\leftarrow$ rs op imm
Lw	lw rt, offset(rs) #rt $\leftarrow$ memory[rs + offset]
Sw	sw rt, offset(rs) #memory[rs + offset] $\leftarrow$ rt
beq	beq rs, rt, label #if (rs==rt) PC $\leftarrow$ label
bne	bne rs, rt, label #if (rs!=rt) PC $\leftarrow$ label
Lui	j target #PC $\leftarrow$ target
J	jal target #r31 $\leftarrow$ PC+8 ; PC $\leftarrow$ target
Jal	jr rs #PC $\leftarrow$ rs

Interrupt and exception instruction	meaning
syscall	System call
eret	Exception execution return
mfc0	Fetch control word
mtc0	Store control word

III. DESIGN OF THE PIPELINE CIRCUIT

a) The basic concepts of pipelining

CPU pipeline is a kind of technology that decomposes instruction into multiple steps, making each step of the operation overlapped, so as to realize a few instructions in parallel processing and to speed up the programming process. Each step has its dependent circuit to handle. When a step is finished, it goes into the next step, and the further step handles the next instruction. When the pipeline technique is adopted, there is no acceleration of single instruction execution, operation steps for each instruction doesn't reduce yet. While different steps of instructions executed at the same time, therefore looked from the overall it speeds up the instruction process, shortens the program execution time. In order to meet the higher clock frequency that common pipeline design can't adapt to, pipeline depth in the high end CPU gradually increases. When the pipeline depth at the 5~6 level and above, usually called super pipelining structure (Super Pipeline). Obviously, the more pipeline stages, each stage time shorter, clock cycle can be designed more short, instruction faster, instruction average execution time is short. Pipelining is by increasing the computer hardware to achieve. It requires each functional section can work independently of each other, which should increase the hardware, correspondingly increase the complexity of control. Without the operating components independent of each other, is likely to occur

in various conflicts. For example, to be able to prefetch instructions, we need to increase the hardware instruction, and store the fetched instructions in the instruction queue buffer, so the microprocessor can fetch and execute instructions to operate at the same time.

b) Design of each stage of the pipeline

i. The design of the instruction fetch stage IF

1. Functional description

IF stage is the first stage of the pipeline, it has four main functions.

- 1) Automatically adds 4 to PC address according to the clock.
- 2) Take PC address to the instruction memory, and fetch the next instruction from the instruction memory, and pass it to the pipeline register in the next level.
- 3) Make decision for the instruction process flow. First, when the CPU processes according to the sequence of the instruction address, we choose the address of the next instruction as the address of the previous instruction plus 4. Second, when the CPU performs conditional branch instruction, we use mux to choose branch address. Third, when the CPU performs register branch instruction, we register branch address according to the mux. Fourth, when the CPU performs jump instruction, we use mux to choose branch address.

- 4) When the control hazard occurs, the CPU fetches temporary instruction and send empty instruction to the decode stage.

2. Module division

As Figure 3-1 shows, IF stage is made of two modules – program pointer register PC and program memory module.

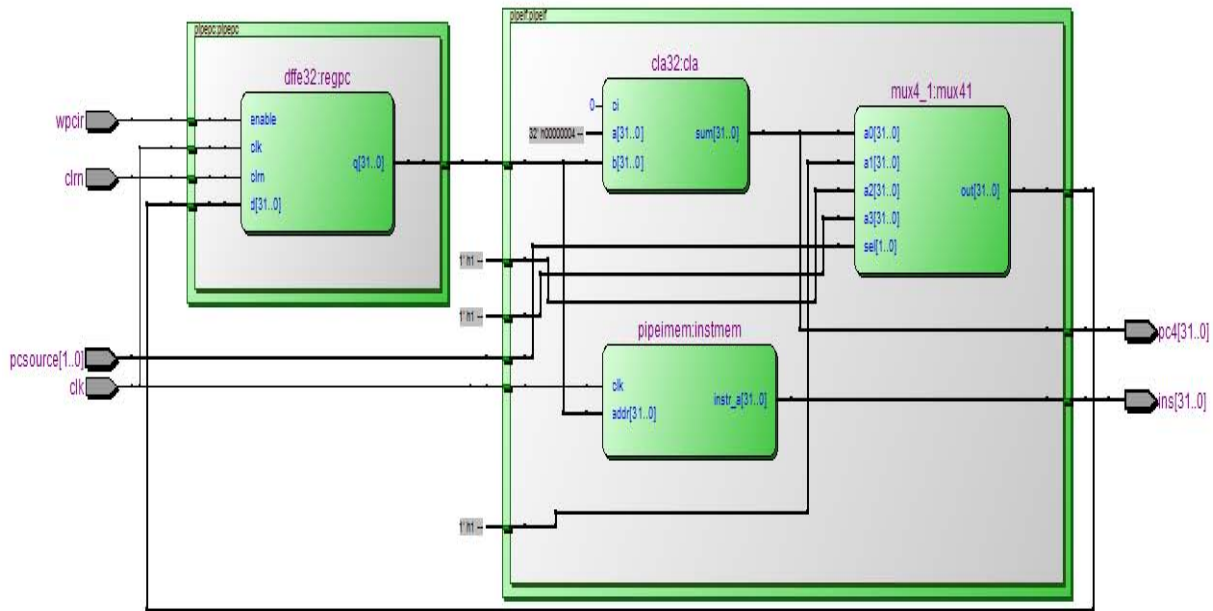


Figure 3-1 : IF stage module division

3. Logic implementation

The design uses 32 bit register with enable bit to implement program pointer register PC. The automatically adding of the address is done by a constant adder with incremental value 4. We use Altera LPM_Mem IP to implement instruction memory quickly.

ii. The design of the instruction decode stage ID

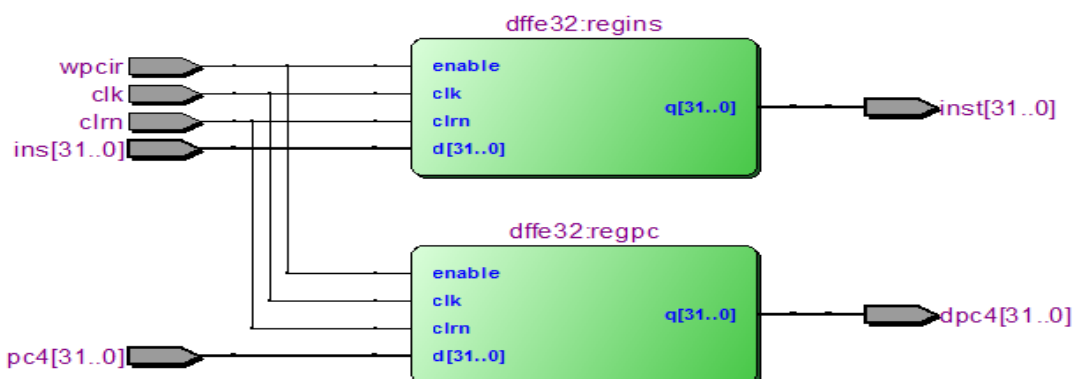
1. Functional description

ID stage is the second stage of the pipeline, it has three main functions.

- (1) Decode the instruction and control each module of the CPU according to the decoding result.
- (2) Implement register file
- (3) Control the pipeline process through control unit.

2. Module division

As Figure 3-2 shows, ID stage is made of three modules –pipeline register, common register file and control unit.



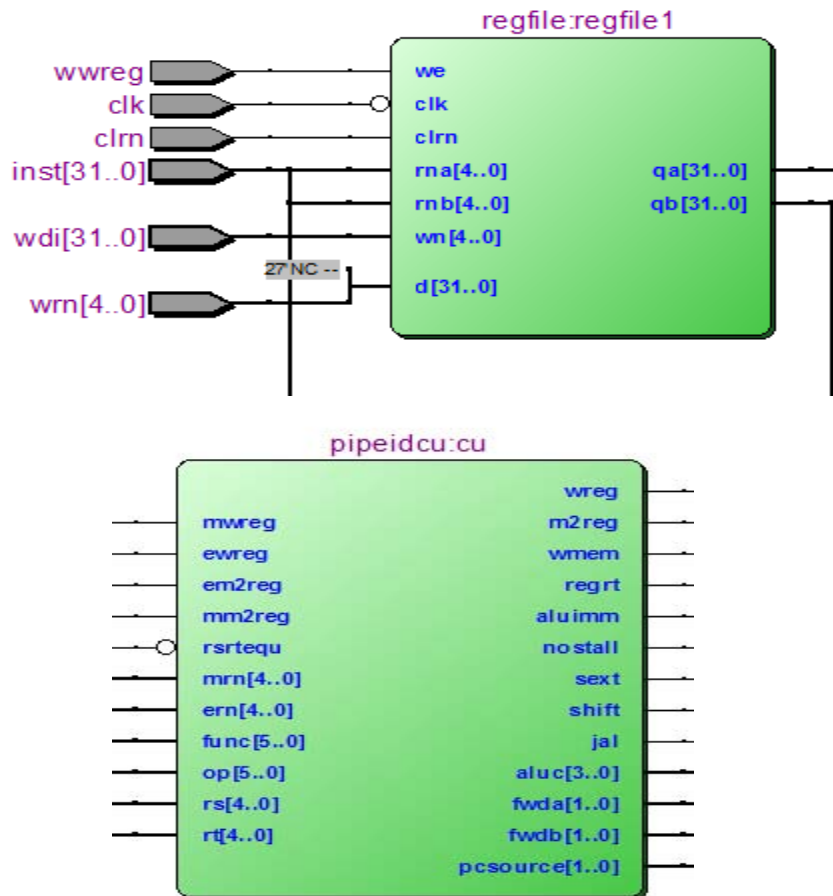


Figure 3-2 : ID stage module division

3. Logic implementation

Realization method of pipeline registers is identical with the PC in the front section, so we will not repeat them. Regfile uses multiplexer for multiple address choice. At the same time, according to MIPS architecture, we set the value of register 0 as constant 0. The control unit all uses a hard-wired logic circuits to achieve.

iii. The design of the execution stage EXE

1. Functional description

EXE stage is the third stage of the pipeline. Its main functions are to calculate the input data and other logic process according to the control signal aluc. Control signal aluc is defined as follows

Table 3-1 : aluc control signal

aluc[3:0]	
X000	ADD
X100	SUB
X001	AND
X101	OR
X010	XOR
X110	LUI
0011	SLL
0111	SRL
1111	SRA

2. Module division

As Figure 3-3 shows, EXE stage is made of ALU and multiplexer.

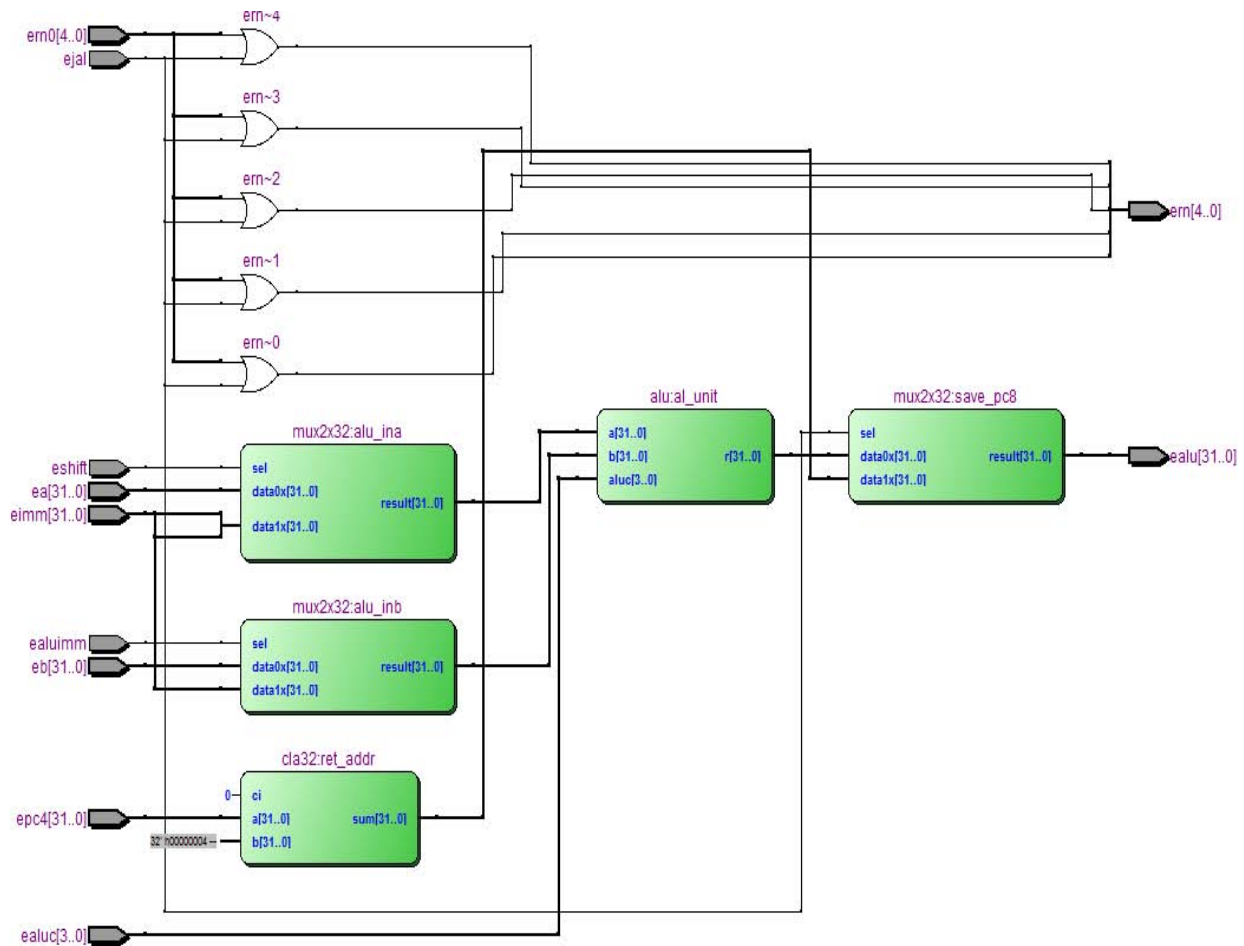


Figure 3-3 : EXE stage module division

3. Logic implementation

Because the ALU to complete a series of logic operations such as addition and subtraction shift, so we need to use special optimization algorithms and architectures, in order to realize the fast operation, and shorten the critical path delay line.

iv. The design of the memory stage MEM

1. Functional description

MEM stage is the fourth stage of the pipeline. Its main function is to read and write data memory.

2. Module division

As Figure 3-4 shows, MEM stage is made of a RAM module.

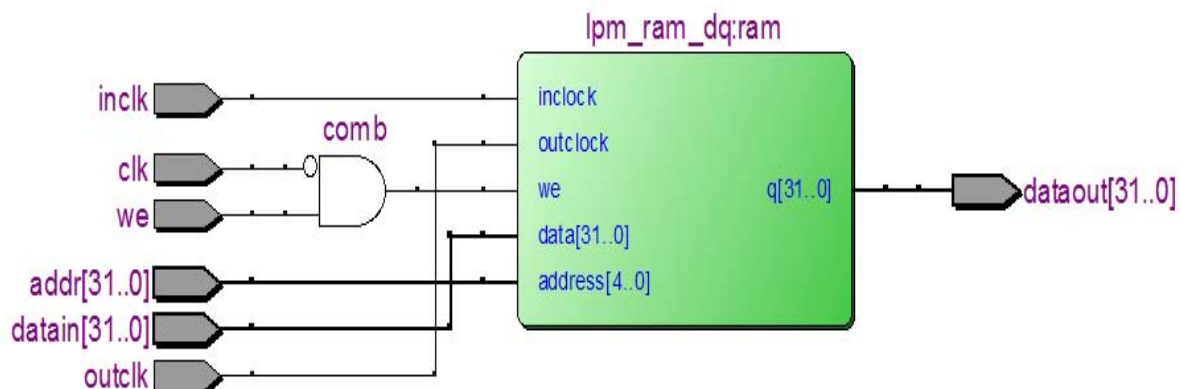


Figure 3-4 : MEM stage module division

3. Logic implementation

We use Altera's LPM_Mem Ram IP core to build the RAM module.

2. Module division

v. The design of the write back stage WB

1. Functional description

Write back stage is the fifth stage of the pipeline. Its main function is to put the result of previous stage back to the register file.

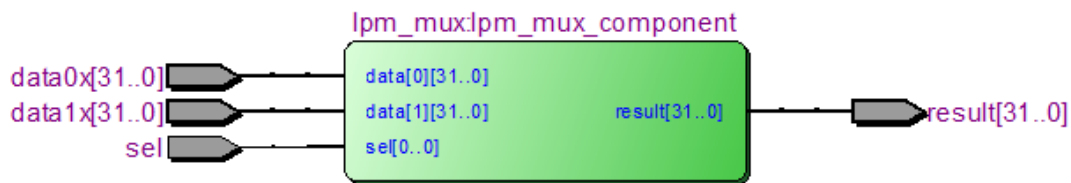


Figure 3-5 : WB stage module division

3. Logic implementation

Write back stage is only made of a multiplexer.

c) Solve the pipeline hazard

Foreword

Hazard means

In CPU design, Hazards are problems with the instruction pipeline in central processing unit (CPU) microarchitectures when the next instruction cannot execute in the following clock cycle, and can potentially lead to incorrect computation results. There are typically three types of hazards.

- data hazards
- structural hazards
- control hazards (branching hazards)

There are several methods used to deal with hazards, including pipeline stalls/pipeline bubbling, register forwarding, and in the case of out-of-order execution, the scoreboarding method and the Tomasulo algorithm.

i. Data hazard

Data hazard

Data hazards occur when instructions that exhibit data dependence modify data in different stages of a pipeline. Ignoring potential data hazards can result in race conditions (sometimes known as race hazards). There are three situations in which a data hazard can occur:

1. Read after write (RAW), a true dependency
2. Write after read (WAR), an anti-dependency
3. Write after write (WAW), an output dependency

For simple pipeline, only RAW may result in data hazard. Other two circumstances can only occur in superscalar CPU. So we will only discuss RAW data hazard in this paper.

Solution

First we'll analyze what lead to data hazard.

Situation 1 data hazard occurs when the previous instruction doesn't finish, while its next instruction will use its results. For register level, see the following instruction sequence.

```
add r3, r1, r2
sub r4, r9, r3
or r5, r3, r9
xor r6, r3, r9
and r7, r3, r9
```

The first instruction put the result of the adding process into register r3. In this case, the following instructions sub, or, xor cannot take the right result in the ID stage.

There are two ways to solve this problem.

1. Stall the pipeline. Although this way can fundamentally solve data hazard, it will make pipeline stall and reduce instruction number in unit time. Therefore it is the worst.
2. Use internal forwarding. Let's look into the first and the second instruction. Because when ALU is doing subtraction, addition has been completed. Therefore we allow ALU sent the result of the addition directly to the next instruction in the ID stage to use. This method will not stall the pipeline. So it has advantages over method one.

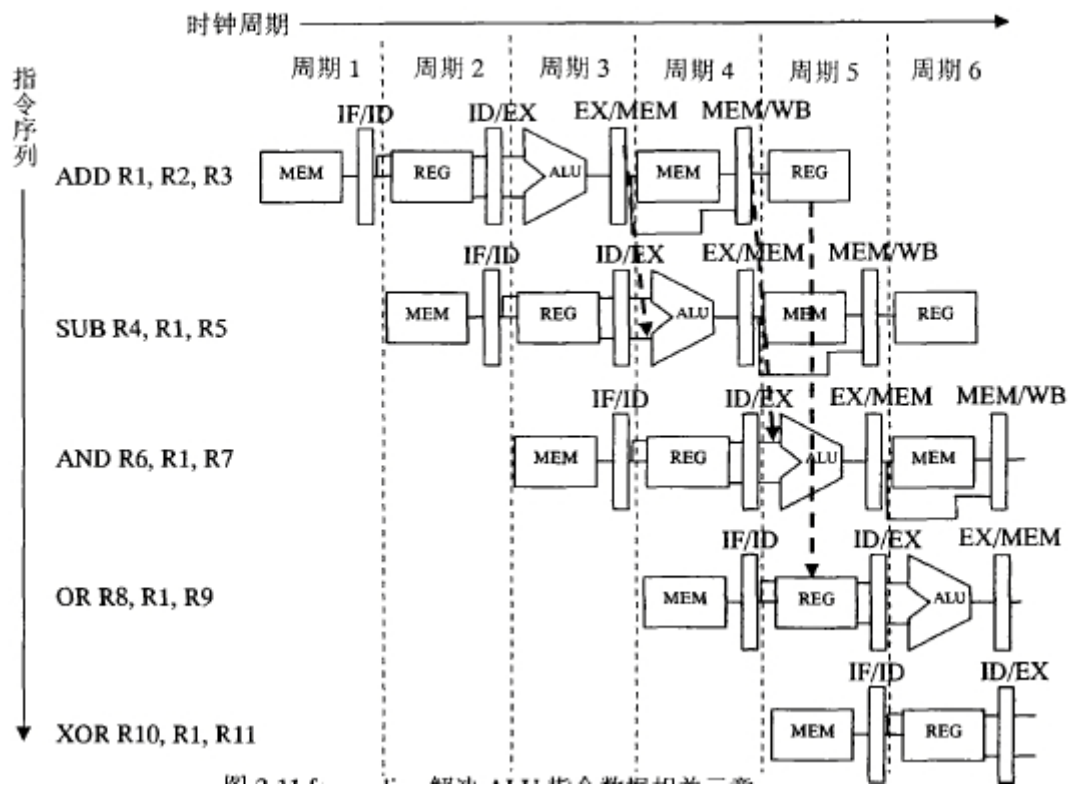


Figure 3-6 : MEM stage module division

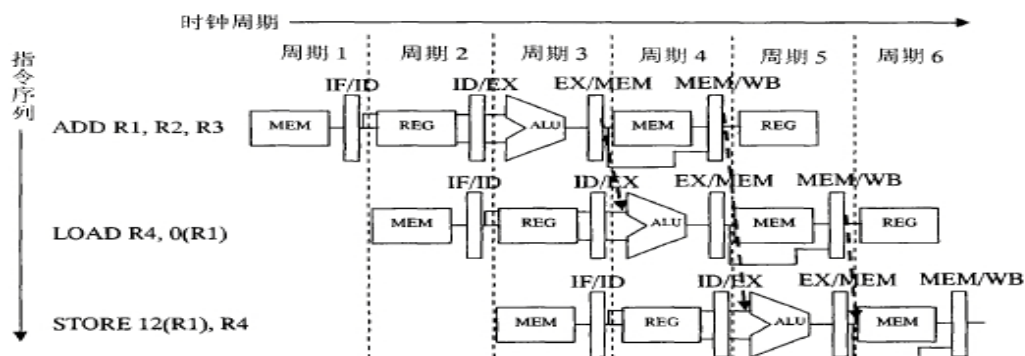


Figure 3-7 : MEM stage module division

When specific to the logic circuit design, consider by increasing the connection from the EXE stage and MEM stage to the ID stage to achieve Internal Forwarding.

Make different stages of the internal forwarding pipeline input into one MUX, and the control unit control the MUX channel selection by forwarding conditions.

The main circuit is as follow

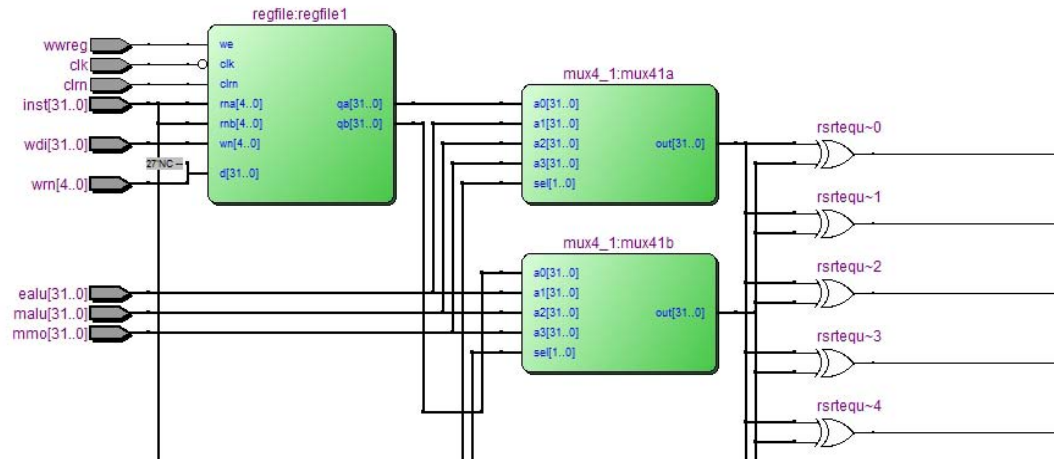


Figure 3-8 : internal forwarding

As Figure 3-6 shows, signal `ealu[31:0]` is the internal forwarding signal of ALU's output, `malu[31:0]` is the internal forwarding signal of MEM stage. `mmo[31:0]` is the internal forwarding signal of data cache's output. We use MUX to connect each data path.

Situation 2 Can we solve all the data hazard by internal forwarding? No. The result of ALU can push forward from EXE stage and MEM stage to ID stage, while the data that instruction `lw` read from data memory can only push from MEM to ID. That means if the next instruction is associated with the `LW` instruction, we have to stall the pipeline for one cycle—results in pipeline bubble.

See the following instruction sequence.

```
lw r3, 0(r1)
sub r4, r9, r3
or r5, r3, r9
xor r6, r3, r9
and r7, r3, r9
```

As Figure 3-9 shows, when the CPU processing the second instruction, it has to stall the pipeline for one cycle to ensure that ID stage gets the right input data, which means the CPU has to repeat the execution for one time.

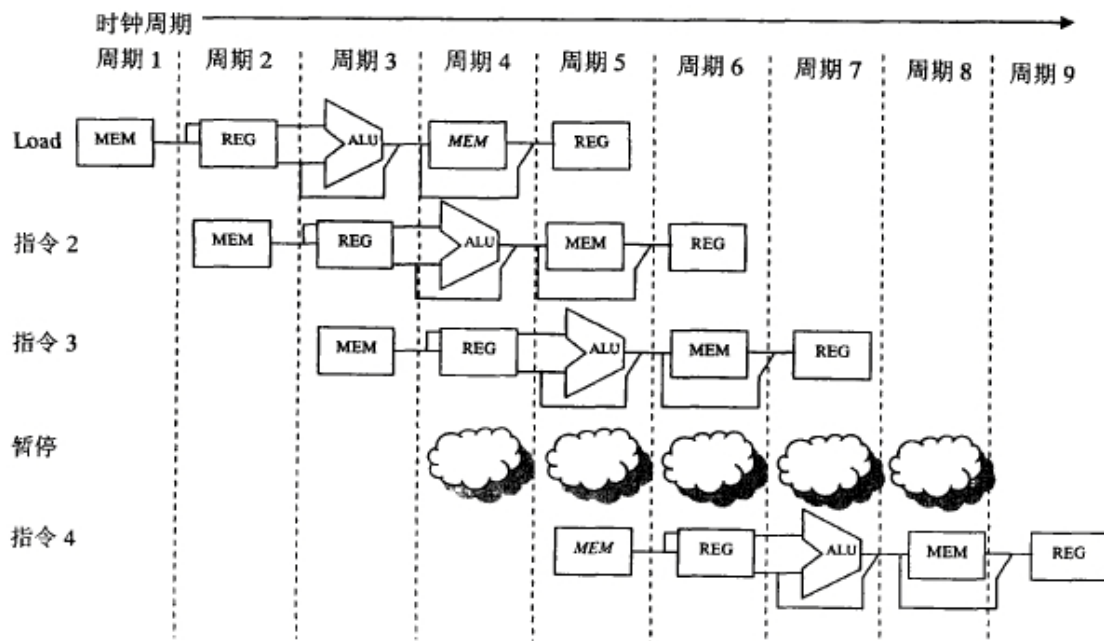


Figure 3-9 : pipeline bubble

When specific to the logic circuit design, consider by adding enable control signal to control PC and ID stage's pipeline register, thus to stall the pipeline.

Corresponding to this, we can use logic statements to judge if there is data hazard. Since if we do not take other measures, stalling the pipeline will result in the re-

execution of the instruction in the IR stage, so the CPU has to discard the execution for one time. We can achieve this by banning modifying CPU state – to block the write register signal wreg and write memory signal wmem.

HDL code is as follows.

```
Stall = ewreg & em2reg & (ern != 0) & (i_rs &
(ern == rs) | i_rt & (ern == rt) );
```

Among them, stall is the control signal to stall the pipeline when data hazard occurs. Ewreg is the signal for writing register file in the EXE stage. Em2reg signal controls the data memory to write data into register file. If the condition is true the stall signal turns high, the line suspension.

ii. Control hazard

Control hazard

Branching hazards (also known as control hazards) occur with branch. On many instruction pipeline microarchitectures, the processor will not know the outcome of the branch when it needs to insert a new instruction into the pipeline (normally the fetch stage).

Solution

Traditional ways

1. Delay the pipeline for two cycles. Since the address and condition for branch target are identified in EXE stage, so the next two instructions after beq have already been put into the pipeline. As Figure 3-10 shows.

周期	1	2	3	4	5	6	7	8	9
分支	IF	ID	EXE	MEM	WB				
延迟槽		IF	ID	EXE	MEM	WB			
后继指令 1			stall	stall	IF	ID	EXE	MEM	WB
后继指令 2						IF	ID	EXE	MEM

Figure 3-10 : Delay the pipeline for two cycles

2. Delay the pipeline for one cycle. If we can identify the address and condition in ID stage, then only one

subsequent instruction will put into the pipeline. As Figure 3-11 shows.

周期	1	2	3	4	5	6	7	8	9
分支	IF	ID	EXE	MEM	WB				
延迟槽		IF	ID	EXE	MEM	WB			
后继指令 1			stall	IF	ID	EXE	MEM	WB	
后继指令 2					IF	ID	EXE	MEM	WB

Figure 3-11 : pipeline bubble

Solutions for control hazard in MIPS architecture

MIPS architecture introduces Branch delay slot concept and it solves the problem of control hazard. Branch delay is an instruction after a branch instruction. No matter branch occurs or not it's always executed. Besides, the instruction in the delay slot is committed before branch instruction.

In the pipeline, branch instruction has to wait until the second stage to identify the address of next instruction. The instruction fetch stage of pipeline will not work until branch instruction is executed, therefore the pipeline has to waste (block) a time slice.

To use this time slice, we define a time slice after branch instruction as branch delay slot. In the branch delay slot the instruction is always executed, and branching occurs whether or not it doesn't matter. In this

way we efficiently take advantage of a time slice, eliminating a "bubble line".

Solution methods in this paper

According to the MIPS architecture, this paper chooses a design method which using a delay slot to identify branch target address and condition in ID stage. Whether to branch or not, the one (instruction i) after branch instruction (instruction i+1) is always executed. As if it's instruction i-1. As Figure 3-12 shows.

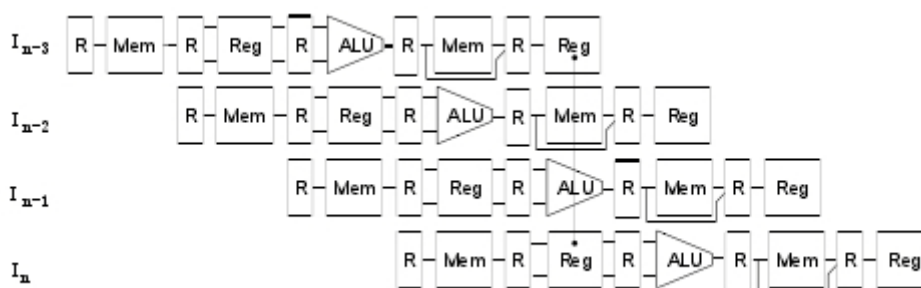


Figure 3-12 : pipeline bubble

For the pipeline CPU in this paper, there're 5 branch instructions – jr, beq, bne, j and jal. Since j, jal and jr are unconditional jump instruction, the CPU can identify branch target in ID level. For instruction beq and bne, we consider using XOR gate and NOR gate to realize these two comparisons in ID level.

iii. Structure hazard

Structure hazard

Structure hazard occurs when multiple instructions visit a hardware component of the processor at the same time. A typical example is an

instruction fetches operands from a storage unit while the other one writes into it.

Let's take MIPS pipeline for example. For the early processors, programs and data memory are not separated, as Figure 3-13 shows, there're memory access at the same time in IF and MEM stage. This results in that one of the accesses has to wait for a cycle. For modern processors, the program is stored in L1P Cache and the data is stored in L1D Cache. They are accessed separately so structure hazard is not a problem.



Figure 3-13 : structure hazards

Solution methods in this paper

In this paper, since we use separated I-cache and D-cache, structure hazard is avoided.

IV. DESIGN OF THE INTERRUPT AND EXCEPTIONAL CIRCUIT

In this chapter we introduce design of the Interrupt and Exceptional Circuit. First we introduce the concept of interrupt, exception and precise interrupt. And then we introduce the hardware interrupt processing structure with MIPS architecture and the related interrupt exception handling instruction set. Finally, we discuss the pipelined CPU terminal and exception handling circuit realization. Meanwhile, we will provide the RTL diagram and related codes.

a) The MIPS exception and interrupt handling principle

i. Exception, interrupt and precise exception

In systems programming, an interrupt is a signal to the processor emitted by hardware or software indicating an event that needs immediate attention. An

interrupt alerts the processor to a high-priority condition requiring the interruption of the current code the processor is executing (the current thread). The processor responds by suspending its current activities, saving its state, and executing a small program called an interrupt handler (or interrupt service routine, ISR) to deal with the event. This interruption is temporary, and after the interrupt handler finishes, the processor resumes execution of the previous thread.

A hardware interrupt is an electronic alerting signal sent to the processor from an external device, either a part of the computer itself such as a disk controller or an external peripheral. For example, pressing a key on the keyboard or moving the mouse triggers hardware interrupts that cause the processor to read the keystroke or mouse position. Unlike the software type (below), hardware interrupts are asynchronous and can occur in the middle of instruction execution, requiring additional care in programming. The act of initiating a hardware interrupt is referred to as an interrupt request (IRQ).

A software interrupt is caused either by an exceptional condition in the processor itself, or a special instruction in the instruction set which causes an interrupt when it is executed. The former is often called a trap or exception and is used for errors or events occurring during program execution that are exceptional enough that they cannot be handled within the program itself. For example, if the processor's arithmetic logic unit is commanded to divide a number by zero, this impossible demand will cause a divide-by-zero exception, perhaps causing the computer to abandon the calculation or display an error message. Software interrupt instructions function similarly to subroutine calls and are used for a variety of purposes, such as to request services from low level system software such as device drivers. For example, computers often use software interrupt instructions to communicate with the disk controller to request data be read or written to the disk.

ii. Exception and interrupt handling in MIPS

Interrupts

The processor supports eight interrupt requests, broken down into four categories:

- Software interrupts - Two software interrupt requests are made via software writes to bits IP0 and IP1 of the Cause register.
- Hardware interrupts - Up to six hardware interrupt requests numbered 0 through 5 are made via implementation-dependent external requests to the processor.
- Timer interrupt - A timer interrupt is raised when the Count and Compare registers reach the same value.
- Performance counter interrupt - A performance counter interrupt is raised when the most significant bit of the counter is a one, and the interrupt is enabled by the IE bit in the performance counter control register.

Timer interrupts, performance counter interrupts, and hardware interrupt 5 are combined in an implementation dependent way to create the ultimate hardware interrupt 5.

Exceptions

Normal execution of instructions may be interrupted when an exception occurs. Such events can be generated as a by-product of instruction execution (e.g., an integer overflow caused by an add instruction or a TLB miss caused by a load instruction), or by an event not directly related to instruction execution (e.g., an external interrupt). When an exception occurs, the processor stops processing instructions, saves sufficient state to resume the interrupted instruction stream, enters Kernel Mode, and starts a software exception handler. The saved state and the address of the software exception handler are a function of both the type of exception, and the current state of the processor.

b) Pipeline CPU precise exception and interrupt processing circuit

The complexity of pipelined CPU exception and interrupt handling is mainly reflected in two respects. (1) Pipeline CPU has multiple instructions simultaneously in operation. There is not a time point that all the instructions are executed. (2) MIPS pipeline allows the branch delay. If exception and interrupt occurs in the delay slot of ID stage, then the return address will not be judged. Therefore, to achieve precise exception and interrupt handling, we must carefully study the characteristics of CPU and design of hardware.

i. Types of exception and interrupt and associated registers

The registers for pipeline CPU exceptions and interrupts are shown as below. The sixth to second bit of the cause register is the codes that generate exception and interrupt. IM[3:0] in the status register is a 4 bit mask. Each corresponding to an exception or interrupt mask bit, 1 allows the exception or interrupt and 0 bans it. S[3:0] is IM[3:0] which is left shifted by 4 bits. EPC is used for saving the return address. If the instruction that causes exception is in the delay slot of branch or jump instruction, then the BD bit is set to 1. Under normal circumstances we set BD to 0.

Table 4-1 : MIPS exception and interrupt registers

Cause	31	4	3	2	1	0	
	BD	Unused				ExcCode	0
Status	31	8	7	4	3	0	
	Unused				S[3:0]	IM[3:0]	
EPC	31	0				EPC	

The following table lists the exceptions and interrupts may appear which level in the pipeline.

Table 4-2 : MIPS exception and interrupt

ExcCode	Alias	Type	Mask	Description	Stage
0	Int	Int	IM[0]	External Interrupt	Any stage
1	Sys	Except	IM[1]	Syscall	ID stage
2	Unimpl	Except	IM[2]	Non-exist instruction	ID stage
3	Ov	Except	IM[3]	Overflow	EXE stage

When interrupt occurs EPC saves the return address, and when exception occurs EPC saves the address that causes exception. However, if the abnormal instruction is in delay slots, EPC save the delayed branch instruction address. Therefore, we must have some means to determine whether an instruction in the delay slot. When an exception or interrupt occurs, we need to scrap the subsequent instruction and even the current instruction.

ii. *Interrupt response process of the pipeline CPU*

In order to realize the precise interrupt, we divide the interrupt request into the following 3 kinds of circumstances.

- (1) Interrupt occurs while ID's executing the transfer instruction

The design idea is to put the address of branch instruction into EPC register, and the base address of

interrupt execution into PC. When returns from interrupt execution, the CPU re-execute branch instruction.

As Figure 4-1 shows, if the address of branch instruction is PC in the IF level, then in the ID level is PCD. When ID finish, the CPU write PCD into EPC, and put the entrance address of exception and interrupt into PC. Meanwhile, the cancel signal generated at the ID level is also written to the pipeline register, its output at EXE stage is e_cancel, to abolish the next instruction. When it returns from interrupt, directly write the value of EPC into PC, and re-execute the branch instruction.

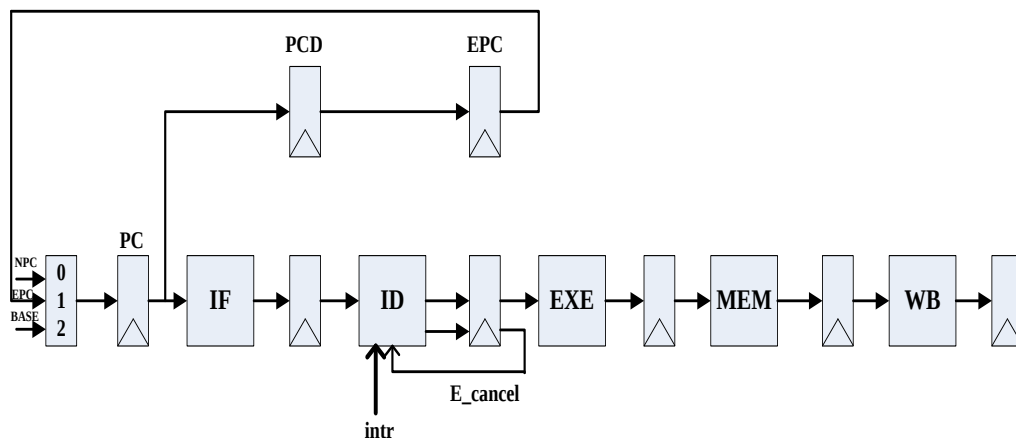


Figure 4-1 : Interrupt occurs while ID's executing the transfer instruction

- (2) Interrupt occurs while ID's the delay slot

The design idea is to let the instruction in the delay slot be finished. Because the branch address is decoded in ID stage, so we put the address value of PC into EPC. Meanwhile the BD bit in the Cause register should be set to 1.

As Figure 4-2 shows, At the ID level delay slot instruction at the end, the branch target address instruction has been taken into the pipeline, we disable it using e_cancel.

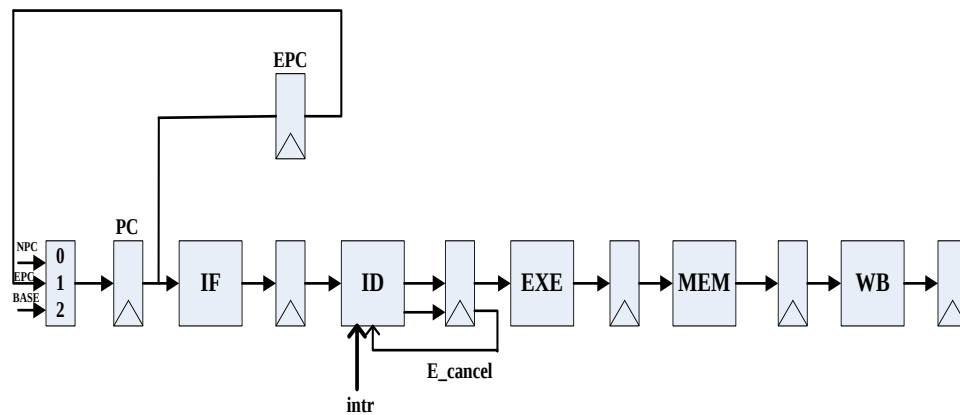


Figure 4-2 : Interrupt occurs while ID's the delay slot

(3) Interrupt occurs in general situation

The design idea in this circumstance is to response to the interrupt in ID stage, to abolish the next instruction, and write the address of the next instruction into EPC. In this case we don't need to set the BD to 1. The pipeline design in this case is identical to the second situation, so It is unnecessary to go into details here.

iii. Precise interrupt of the pipeline CPU

The pipeline CPU puts the address that causes exception into EPC. If this instruction is in the delay slot, then put its previous branch or jump address into EPC and set the BD as 1. The following describes in detail the methods to handle pipeline exception.

(1) Syscall

No matter using assembly language or high level language to program, we can let the syscall instruction not to appear in the delay slot. Therefore, we consider only usually system call instructions implementation. Figure 4-3 shows the pipeline progress that CPU executing syscall instruction. Jump to exception and interrupt handling program and abolish its next instructions. EPC saves PCD -the address of syscall instruction. As the figure shows the input of the EPC connects to a mux, when it modifies EPC it chooses DATA.

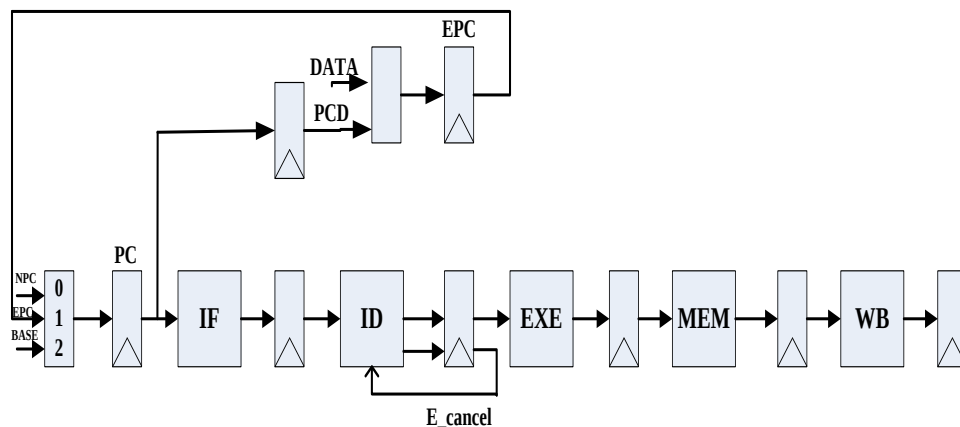


Figure 4-3 : Interrupt occurs in general situation

(2) Unimplemented instruction

The following picture shows the pipeline that CPU's executing unimplemented instructions in the delay slot. In this case EPC will save the address of its previous instruction PCE, the BD bit of Cause register should be set to 1.

Figure 4-4 shows the pipeline progress that CPU executing unimplemented instruction. It is similar with the implementation of the syscall instruction.

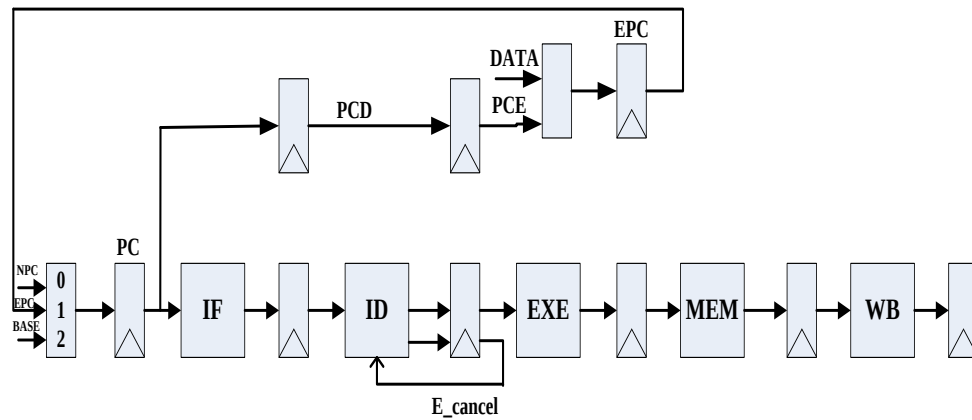


Figure 4-4 : Interrupt occurs in general situation

(3) Calculation result overflow

Calculation result overflow appears in EXE stage, and the result of overflow cannot be saved into

register file. So the CPU has to block the wreg signal in the EXE stage. Besides, instructions in ID stage should be abolish too.

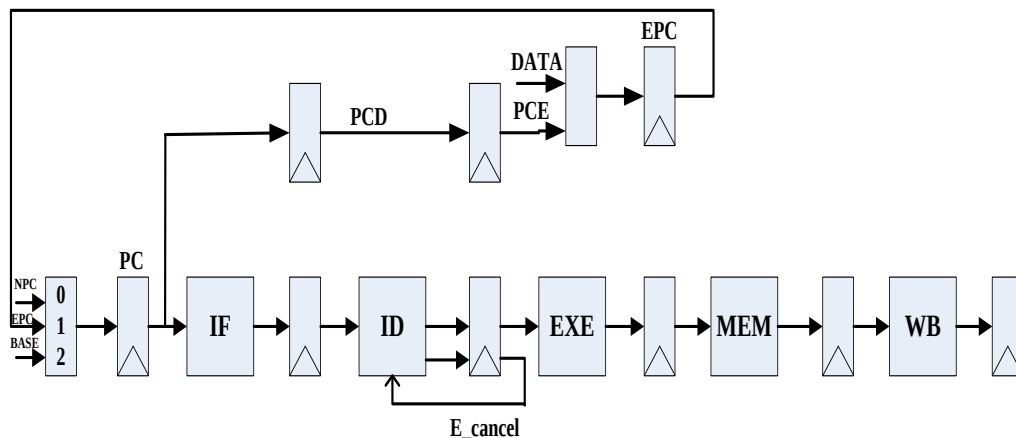


Figure 4-5 : Interrupt occurs in general situation

Figure 4-4 is the pipeline that when CPU is in the delay slot overflow. At this time EPC has to save the PCM address of its previous instruction. The BD bit of cause register should be set to 1.

Figure 4-5 is the pipeline that overflows occurs in normal situation. EPC saves the PCE address of overflow instruction.

simulation, finally the design is downloaded to the FPGA device.

This paper will adopt the design process, optimizing the design details, in order to achieve high quality and efficient design objective.

V. CPU VERIFICATION

Figure 5-1 shows the typical FPGA design and verification overflow. After the design personnel will be HDL code input and comprehensive utilization comprehensive tool, will carry on the first simulation of design: functional simulation. Functional simulation is not with circuit delay parameters, only validation logic function is correct. If the function simulation, then the layout of design, after second times simulation: timing simulation is also the gate level simulation. With Gate level simulation with circuit delay parameters, the result is more accurate, more close to the actual device performance. After passing through the gate level

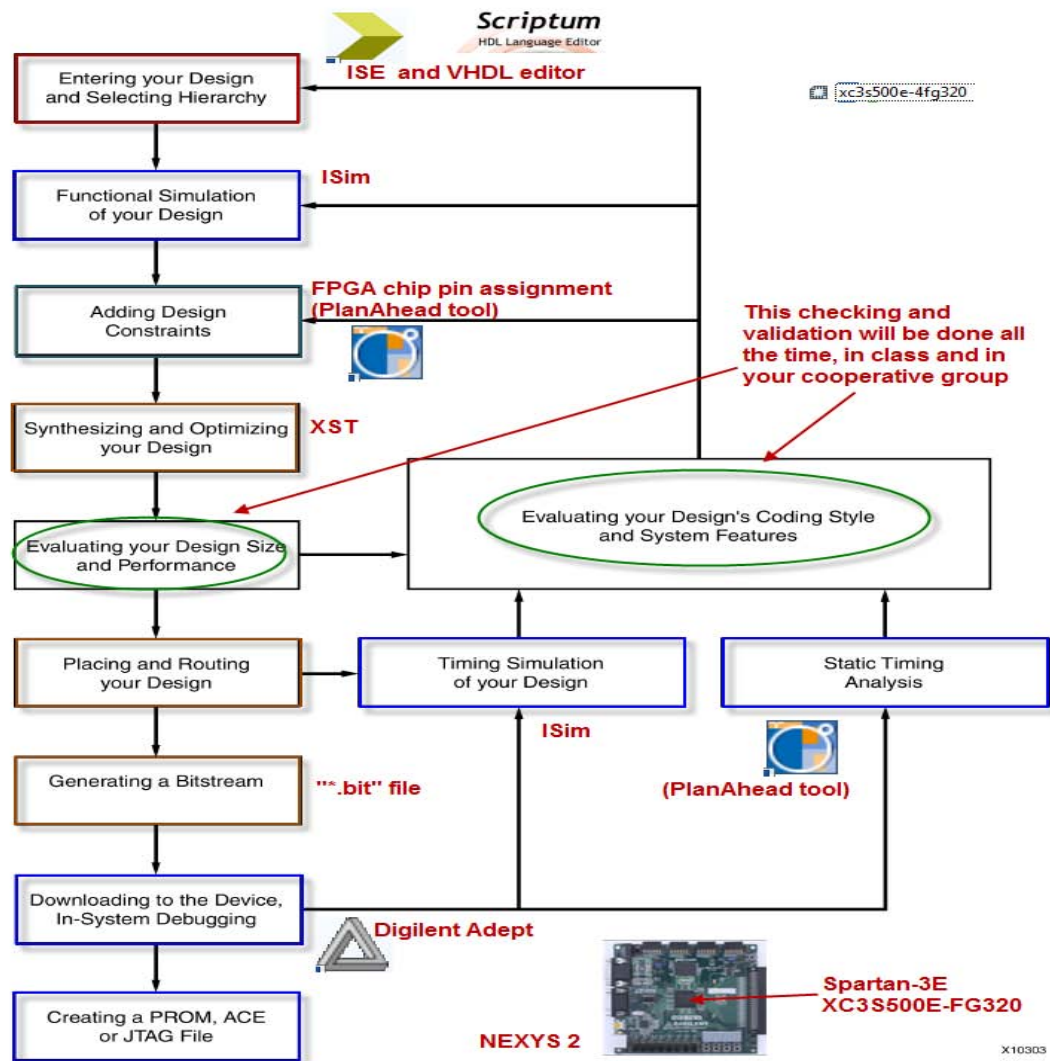


Figure 5-1 : Typical FPGA design & verification overflow

a) Pipeline verification

We implement the idea of bottom-up, hierarchical verification. Firstly, we run functional simulation for single module. If the circuit can realize logic function then we gate-level simulation for every single module. When all the sub-module pass verification, we run simulation for each stage of the pipeline and finally the whole pipeline circuit.

• IF stage

IF stage has two functions: (1) calculate the instruction address (2) fetch instruction and pipeline process. As Figure 5-2 shows, when reset signal *clrn* is low, register PC is set to initial value 0. Signal PC4 is the address value plus 4, at this time the enable signal of register PC is low, and stage IF doesn't work yet. When reset signal and enable signal turns high, register PC starts to work. We can see that PC of continues adding 4 with the change of *clk* signal, which proves that stage IF can finish address calculating. Meanwhile, instruction signal *ins* changes with the change of PC4 signal, and the output corresponding with ROM address of the

instruction value, proving that IF stage can complete instruction fetching and pipeline process.

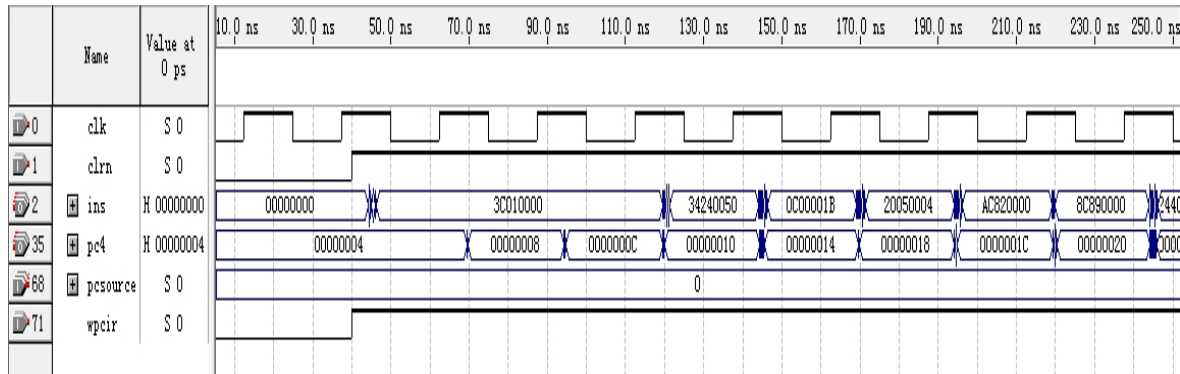


Figure 5-2 : IF stage verification

- ID stage

ID stage has three functions: (1) decode the instruction fetched from instruction cache (2) put the corresponding control word, immediate word and address value to control unit and register file (3) control unit output corresponding signals according to the input logic. In this chapter we will verify the regfile and control unit.

As Figure 5-3 shows, it's each signal of register file and their simulation waves. Clrn is reset signal. d is input data value. qa is the output value of register file output a. qb is the output value of register file output b. ma is qa's output address, which means corresponding

register number. We is write-enable signal, wn is write register number.

When clrn is low, register file is reset, and all the value of registers are set to 0. When clrn turns to high, register file starts to work. When write signal we turns high, wn controls the write register address. As Figure 5-3 shows, the CPU writes value into 0 to 4 register. According to MIPS architecture, 0 register cannot be modified, thus the output of 0 register is always 0. For register 1 to 3, the output is the input value when we is high. So this verifies that register file can work normally and realize the logic function.

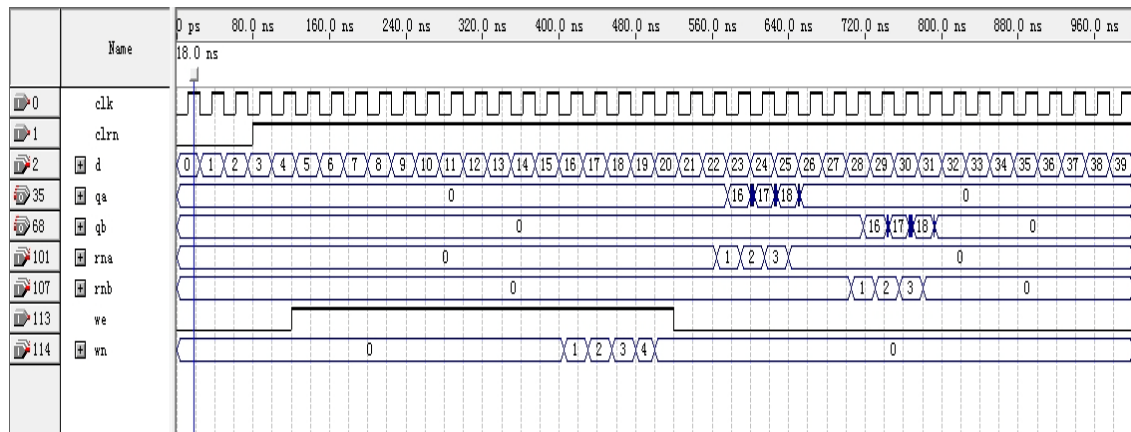


Figure 5-3 : regfile verification

Each signal of control unit and corresponding simulation waves are shown in Figure 5-4. Since control unit is pure logic circuit, we can see apparent glitch in the output wave of the circuit. The main signals are as follows: aluc controls the output of signal alu. Op is instruction word, representing instruction code. Pcsorce controls the mux for address source of pc. We can see from the simulation waves that the control unit can output correct control logic. Further verification will be shown in the later chapter.

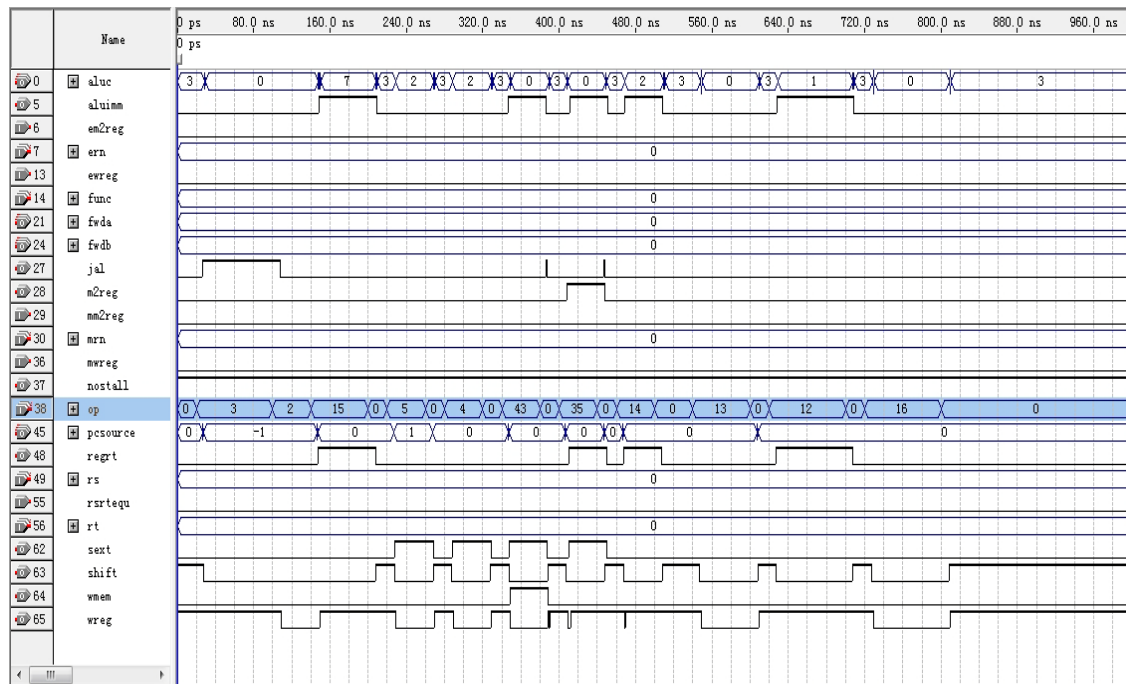


Figure 5-4 : Control Unit verification

- EXE stage

The function of EXE stage is to calculate the value from ID and put the result to the next stage. According to the control signal passed from ID stage, EXE will finish corresponding calculation in the pipeline clock cycle. Since arithmetic unit may cause long delay, so EXE stage is the critical path of the pipeline. We will verify ALU in this chapter.

As Figure 5-5 shows, it's each signal of ALU and their simulation waves. The definition of the control

signals of ALU sees the 3.2.3 section. When ALUC=0 the two input numbers are added, the result is correct. When ALUC=4 the two input numbers are subtracted, the result is correct. The other functions of ALU are verified as above.

We can see from the waveform that, because the signal judging overflow and zero is used in combinational logic circuit, so it will easily produce burr.

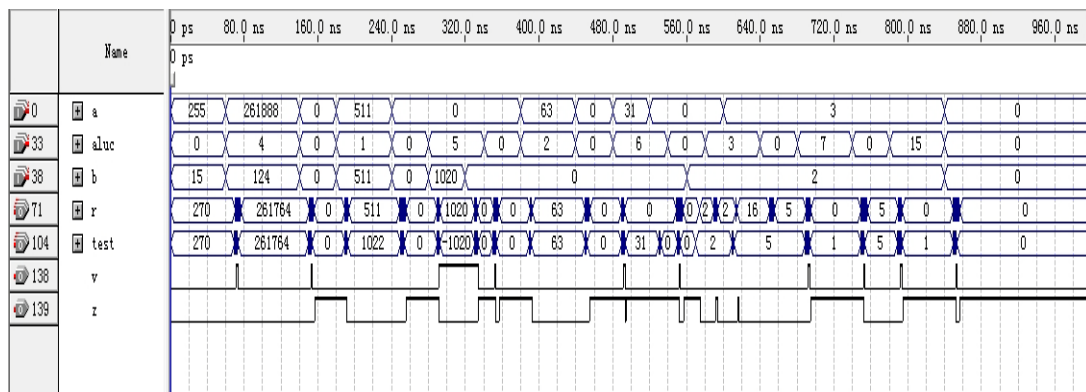


Figure 5-5 : ALU verification

- MEM stage

The function of MEM stage is to load and store data. The main module in MEM stage is data memory. In this paper we use the LPM Ram provided by Altera to achieve this.

As Figure 5-6 shows, it's the simulation wave for data memory. Addr controls the RAM address of data input and output, and we is write enable signal. We can

see from the waveform that, when signal we is low, dataout reads data from RAM according to the value of addr. When signal we turns high, dataout writes data into RAM according to the value of addr too. For data 0x0000007F, we can see that it's written into corresponding address, and is read from it. The simulation waves prove that the data memory works correctly.

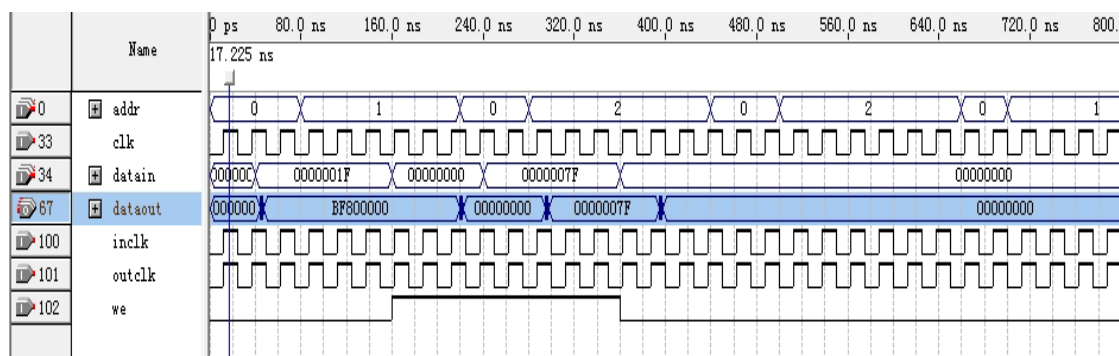


Figure 5-6 : Memory module verification

- WB stage

The function of MEM stage is to store the result calculated from EXE or data in MEM stage. Since there are many data paths, we use mux to achieve that. Because the circuit of this stage is simple, it's unnecessary to run simulation specifically in this chapter.

- pipeline integral verification

- x. Verification program

See appendix A.

- xi. Verification waves

As Figure 5-7 shows, clock is system clock. memclock is ram's clock. Signal inst is the corresponding instruction in a clock cycle. Ealu, malu, walu is the pipeline register value of the output of alu. we can see that corresponding instructions are put into the pipeline with the change of PC. And the output of Alu value has been sent to the next level with the pipeline. The simulation waves prove that the pipeline works correctly.

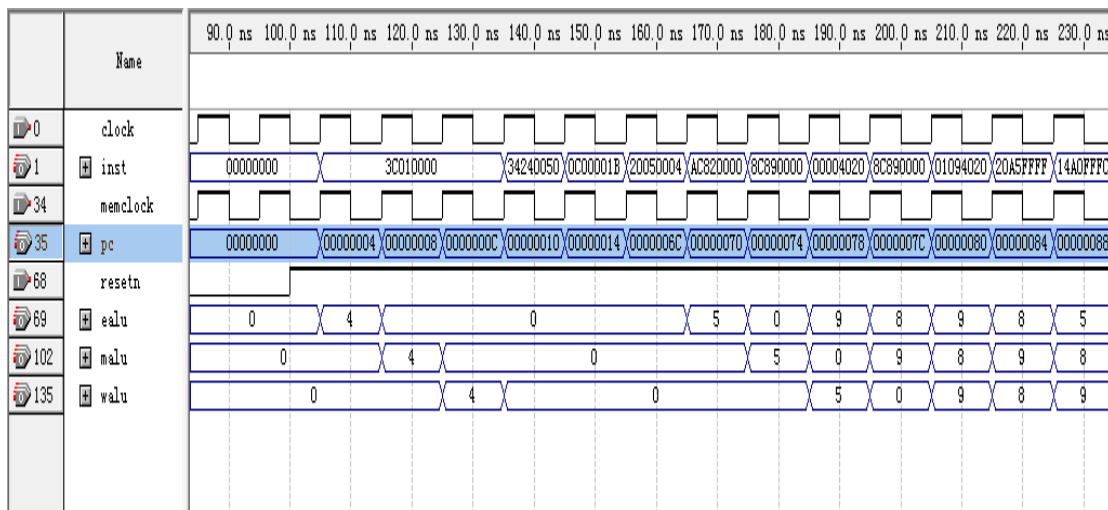


Figure 5-7 : Pipeline integral verification

b) Interrupt and exception circuit verification

i. Verification program

See appendix B.

ii. Verification waves

As Figure 5-8 shows, clock is system clock. memclock is ram's clock. Signal inst is the corresponding instruction in a clock cycle. Ealu, malu, walu is the pipeline register value of the output of alu. The output of Alu value has been sent to the next level with the pipeline. The simulation waves prove that the Interrupt and exception circuit works correctly.

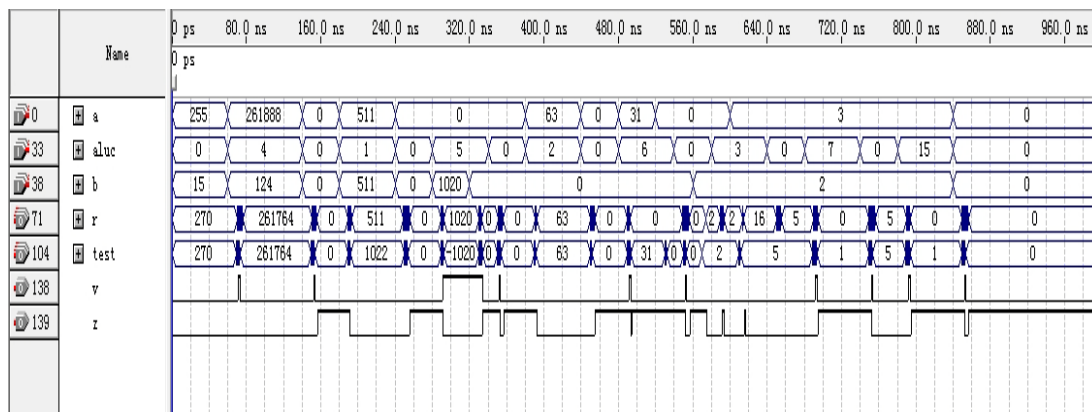


Figure 5-8 : Interrupt and exception circuit verification

VI. SUMMARY AND FUTURE WORK

Through this thesis, Verilog HDL code for a Altera cycloe4 FPGA board was developed, on which a pipeline CPU runs. In chapter 1 we make a brief introduction of the research background. It mainly introduces the background and related research status and CPU's integrated circuit industry. In chapter 2 we describe the development platform and MIPS architecture. There we also introduce the software and hardware development platform for the project and FPGA's design. At the same time, we describe the registers and instructions in our design and MIPS architecture. In chapter 3 we firstly discuss the design of pipeline data path. And we work out some methods for solving pipeline hazards. In Chapter three we describe the design of pipeline data path. It introduces the pipeline design method, the composition of the pipeline and design and verification of associated component. After that, we make the interrupt circuit and whole verification.

In the future, there are two ways to improve this simple CPU. Firstly we will add some modules to the original design, including timer, bus, and the whole CPO coprocessor. Secondly we will improve the circuit thus to make the whole circuit run in a higher frequency. Generally speaking, more performance analysis such as studying the impact of exceptions on the core performance can be done further to see some interesting and important results.

REFERENCES RÉFÉRENCES REFERENCIAS

1. N. K. Choudhary, S. V. Wadhavkar, T. A. Shah, H. Mayukh, J. Gandhi, B. H. Dwiel, S. Navada, H. H. Najaf-abadi, and E. Rotenberg, "FabScalar: Composing Synthesizable RTL Designs of Arbitrary Cores Within a Canonical Superscalar Template", Proceedings of the 38th Annual International Symposium on Computer Architecture. ACM, 2011, pp. 11. [Online]. Available: <http://doi.acm.org/10.1145/2000064.2000067>
2. Naoki Fujieda, Takefumi Miyoshi, and Kenji Kise "SimMips A MIPS System Simulator".
3. MIPS® Architecture For Programmers Volume III-A: Introduction to the MIPS32® Architecture.
4. MIPS® Architecture For Programmers Volume I-A: Introduction to the MIPS32® Architecture.
5. MIPS® Architecture For Programmers Volume II-A: Introduction to the MIPS32® Architecture.
6. Linux Porting Guide. [Online]. Available: <http://www.embedded.com/design/embedded/4023297/Linux-Porting-Guide>
7. Simple Scalar Simulator Toolset. [Online]. Available: <http://www.simplescalar.com/>
8. M. A. Khalighi, N. Schwartz, N. Aitamer, and S. Bourennane, "Fading reduction by aperture averaging and spatial diversity in optical wireless systems," *Journal of Optical Communications and Networking, IEEE/OSA* vol. 1, pp. 580-593, 2009.
9. A. O. Aladeloba, A. J. Phillips, and M. S. Woolfson, "Performance evaluation of optically preamplified digital pulse position modulation turbulent free-space optical communication systems," *IET Optoelectronics*, vol. 6, pp. 66-74, February 2012.
10. L. C. Andrews, R. L. Phillips, and C. Y. Hopen, "Aperture averaging of optical scintillations: power fluctuations and the temporal spectrum," *Waves Random Media*, vol. 10, pp. 53-70, 2000.
11. S. Bloom, E. Korevaar, J. Schuster, and H. A. Willebrand, "Understanding the performance of free-space optics," *Journal of Optical Networking*, vol. 2, pp. 178-200, June 2003.
12. D. K. Borah and D. G. Voelz, "Pointing error effects on free-space optical communication links in the presence of atmospheric turbulence," *Journal of Lightwave Technology*, vol. 27, pp. 3965-3973, 2009.
13. A. A. Farid and S. Hranilovic, "Outage capacity optimization for free-space optical links with pointing errors," *Journal of Lightwave Technology*, vol. 25, pp. 1702-1710, 2007.
14. H. G. Sandalidis, T. A. Tsiftsis, G. K. Karagiannidis, and M. Uysal, "BER performance of FSO links over

- strong atmospheric turbulence channels with pointing errors," *IEEE Communications Letters*, vol. 12, pp. 44-46, 2008.
15. A. J. Phillips, "Power penalty for burst mode reception in the presence of interchannel crosstalk," *IET Optoelectronics*, vol. 1, pp. 127-134, 2007.
 16. K. W. Cattermole and J. J. O'Reilly, *Mathematical topics in telecommunications volume 2: problems of randomness in communication engineering*, Pentech Press Limited, Plymouth, 1984.
 17. I. T. Monroy and E. Tangdionga, *Crosstalk in WDM communication networks*, Kluwer Academic Publishers, Norwell, Massachusetts, USA, 2002.
 18. J. O'Reilly and J. R. F. Da Rocha, "Improved error probability evaluation methods for direct detection optical communication systems," *IEEE Transactions on Information Theory*, vol. 33, pp. 839-848, 1987.
 19. L. F. B. Ribeiro, J. R. F. Da Rocha, and J. L. Pinto, "Performance evaluation of EDFA preamplified receivers taking into account intersymbol interference," *Journal*
 20. Mishra Prabhat, Dutt Nikil, Nicolau Alex. Specification of Hazards, Stalls, Interrupts, and Exceptions in Expression, Technical Report #01-05, Dept. of Information and Computer Science, University of California, Irvine, CA 92697, 2001, USA
 21. Smith James.E., Plezskun Andrew R, Implementing Precise Interrupts in Pipelined Processors, *IEEE Transactions on Computers*. 1988, 37(5), pp. 562-573
 22. Wang Chia-Jiu, Emmett Frank. Implementing Precise Interrupts in Pipelined RISC Processors. *IEEE, Micro*. 1993,13(4) , pp. 36-43
 23. KE Xi-ming. Implementation Mechanism of Precise Interrupts in Microprocessors, *High Performance Computing Technology*. 2003, 160, pp. 45~48
 24. XI Chen, ZHANG Sheng-bing, SHEN Xu-bang, et al. New precise interrupt mechanism based on backup- buffer. *Computer Engineering and Applications*. 2007,43(6) , pp. 95- 98.
 25. Liu Shibin, GaoDeyuan, Fan Xiaoya, et al. Design of Instruction Decoder for Use in China for an Embedded MPU, *Journal of Northwestern Polytechnic University*. 2001,19(1) , pp. 1-5
 26. D. W. Anderson, F. 1. Sparacio, and F. M. Tomasulo, 'The IBM system1360 Model 91 : Machine philosophy and instruction handling, "IBM 1. Res. Develop., vol. 11, pp. 8-24, Jan. 1967.
 27. Ozer, E.; Sathaye, S.W.; Menezes, K.N.; Banerjia, S.; Jennings, M.D.; Conte, T.M.; "A fast interrupt handling scheme for VLIW processors", *Parallel Architectures and Compilation Techniques*, 1998. Proceedings. 1998 International Conference on Digital Object Identifier: IO.I109IPACT.1998.727184 Publication Year: 1998 , Page(s): 136 - 141
 28. 1. E. Smith and A. R. Pleszkun, "Implementing precise interrupts in pipelined processors," *IEEE Trans. Comput.*, vol. C-37, no. 5, pp. 562-573, May 1988.
 29. W-M.W. Hwu and Y.N. Patt, "Checkpoint Repair for Out-of-Order Execution Machines," *IEEE Trans. Computers*, Vol. C-36, No. 12, Dec. 1987, pp. 1,515-1,522.
 30. Pericas, M., Cristal, A., Gonzalez, R., Jimenez, D.A., Valero, M., "A decoupled KILO-instruction processor", *High-Performance Computer Architecture*, 2006. The Twelfth International Symposium on, On page(s)
 31. Dominic Sweetman, See MIPS Run, Academic Press, 2002.
 32. David A Patterson, John L Hennessy, *Computer Organization and Design: The Hardware/Software Interface*, Morgan Kaufmann Publishers, Inc. 1998
 33. Stephen Brown, Zvonko Vranesic. *Fundamentals of Digital Logic with VHDL Design*, McGraw-Hill, 2000.
 34. Zhu Ziyu, Li Yamin. *CPU Chip Logic Design*. Tsinghua University Press. 2005
 35. "Altera university program—Learning through innovation," Altera Corporation, San Jose, CA, 2011 [Online]. Available: <http://www.altera.com/education/univ/unv-index.html>
 36. A. Clements, "The undergraduate curriculum in computer architecture," *IEEE Micro*, vol. 20, no. 3, pp. 13–21, May–Jun. 2000.
 37. J. Djordjevic, B. Nikolic, T. Borozan, and A. Milenkovic, "CAL2: Computer aided learning in computer architecture laboratory," *Comput. Appl. Eng. Educ.*, vol. 16, pp. 172–188, 2008.
 38. B. Nikolic, Z. Radivojevic, J. Djordjevic, and V. Milutinovic, "A survey and evaluation of simulators suitable for teaching courses in computer architecture and organization," *IEEE Trans. Educ.*, vol. 52, no. 4, pp. 449–458, Nov. 2009.
 39. H. Oztekin, F. Temurtas, and A. Gulbag, "BZK.SAU: Implementing a hardware and software-based computer architecture simulator for educational purpose," in *Proc. 2nd Int. Conf. Comput. Design Appl.*, 2010, pp. 490–497.
 40. V. Gustin and P. Bulic, "Learning computer architecture concepts with the FPGA-based "Move" microprocessor," *Comput. Appl. Eng. Educ.*, vol. 14, pp. 135–141, 2006.

VII. ACKNOWLEDGEMENTS

Foremost, I would like to express my sincere gratitude to my Supervisor Prof.Liu for the continuous support of my thesis work, for his patience, motivation, enthusiasm, and immense knowledge. His guidance helped me to perform my thesis work and in writing this thesis report.

I also would like to say a big thank you to all my colleagues, and the researchers in the Computer Engineering Center at the Guangdong University of

Technology, China, for their support and for making my time in Guangzhou an enjoyable one.

Most importantly, I would like to thank my wife Fathai and my children - for their wonderful support throughout my study. This dissertation is dedicated to them as a token of my gratitude.

学位论文独创性声明

本人郑重声明：所呈交的学位论文是我个人在导师的指导下进行的研究工作及取得的研究成果。尽我所知，除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明，并表示了谢意。本人依法享有和承担由此论文所产生的权利和责任。

论文作者签名： 日期：

学位论文授权使用授权声明

本学位论文作者完全了解学校有关保存、使用学位论文的规定，同意授权广东工业大学保留并向国家有关部门或机构送交该论文的印刷本和电子版本，允许该论文被查阅和借阅。同意授权广东工业大学可以将本学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印、扫描或数字化等其他复制手段保存和汇编本学位论文。保密论文在解密后遵守此规定。

论文作者签名： 日期：

指导教师签名： 日期：

APPENDIX A PIPELINE VERIFICATION PROGRAM

```

0: main :lui r1, 0 # address of data[0]
1: ori r4, r1, 80 # address of data[0]
2: call : jal sum # call function
3: dslot1: addi r5, r0, 4 # counter, DELAYED SLOT(DS)
4: return: sw r2, 0(r4) # store result
5: lw r9, 0(r4) # check sw
6: sub r8, r9, r4 # sub: r8 ← r9 - r4
7: addi r5, r0, 3 # counter
8: loop: addi r5, r5, -1 # counter - 1
9: ori r8, r5, 0xffff # zero-extended : 0000ffff
A: xori r8, r8, 0x5555 # zero-extended : 0000aaaa
B: addi r9, r0, -1 # sign-extended : ffffffff
C: andi r10, r9, 0xffff # zero-extended : 0000ffff
D: or r6, r10, r9 # or: ffffffff
E: xor r8, r10, r9 # xor: ffff0000
F: and r7, r10, r6 # and: 0000ffff
10: beq r5, r0, shift # if r5 = 0, goto shift
11: dslot2: nop # DS
12: j loop2 # jump loop2
13: dslot3: nop # DS
14: shift: addi r5, r0, -1 # r5 = ffffffff
15: sll r8, r5, 15 # <<15 = ffff8000
16: sll r8, r8, 16 # <<16 = 80000000
17: sra r8, r8, 16 # >>16 = ffff8000(arith)
18: srl r8, r8, 15 # >>15 = 0001ffff(logic)
19: finish: j finish # dead loop
20: dslot4: nop # delay slot

```

APPENDIX B INTERRUPT AND EXCEPTION VERIFICATION PROGRAM

```

0: reset : j start # entry on reset
1: nop
2:EXC_BASE: mfc0 r26, C0_CAUSE # read cp0 Cause reg
3: andi r27, r26, 0xc # get ExcCode, 2 bits here
4: lw r27, j_table (r27) # get address from table
5: nop
6: jr r27 # jump to that address
7: nop
c: int_entry: nop #0.interrupt handler deal with interrupt here
d: eret # return from interrupt
e: nop
f:sys_entry: nop # SysCall handler
10: epc_plus4: mfc0 r26, C0_EPC # get EPC
11: addi r26, r26, 4 #EPC + 4
12: mtc0 r26, C0_EPC #EPC ← EPC +4
13: eret #return from exception
14: nop
15: uni_entry: nop
16: j epc_plus4 #return
17: nop
1a: ovf_entry: nop #overflow handler
1b: j epc_plus4 #return
1c: nop
1d: start: addi r8, r0, 0xf # IM[3:0] ← 1111
1e: mtc0 r8, C0_STATUS # exc/intr enable

```





Design and Implementation of Data Scrambler & Descrambler System using VHDL

By Naina K. Randive & Prof. G. P. Borkhade

Sant Gadge Baba Amravati University, India

Abstract- Multimedia data security is very important for multimedia commerce on the internet and real time data multicast. An striking solution for encrypting data with adequate message security at low cost is the use of Scrambler/Descrambler. Scramblers are necessary components of physical layer system standards besides interleaved coding and modulation. Scramblers are well used in modern VLSI design especially those are used in data communication system either to secure data or re-code periodic sequence of binary bits stream. However, it is necessary to have a descrambler block on the receiving side while using scrambling data in the transmitting end to have the actual input sequence on the receiving end. Scrambling and De-scrambling is an algorithm that converts an input string into a seemingly random string of the same length to avoid simultaneous bits in the long format of data. Scramblers have accomplish of uses in today's data communication protocols. On the other hand, those methods that are theoretical proposed are not feasible in the modern digital design due to many reasons such as slower data rate, increasing information, circuit hazards, uncountable hold-up etc. Therefore it is requisite for the modern digital design to have modified architecture to meet the required goal. We will recommend here modified scrambler design which is perfectly suitable for any industrial design.

Keywords: *scrambler, descrambler, VHDL, and FPGA.*

GJCST-A Classification : *B.5.2 B.6.3*



Strictly as per the compliance and regulations of:



RESEARCH | DIVERSITY | ETHICS

Design and Implementation of Data Scrambler & Descrambler System using VHDL

Naina K. Randive ^α & Prof. G. P. Borkhade ^σ

Abstract Multimedia data security is very important for multimedia commerce on the internet and real time data multicast. An striking solution for encrypting data with adequate message security at low cost is the use of Scrambler/Descrambler. Scramblers are necessary components of physical layer system standards besides interleaved coding and modulation. Scramblers are well used in modern VLSI design especially those are used in data communication system either to secure data or re-code periodic sequence of binary bits stream. However, it is necessary to have a descrambler block on the receiving side while using scrambling data in the transmitting end to have the actual input sequence on the receiving end. Scrambling and De-scrambling is an algorithm that converts an input string into a seemingly random string of the same length to avoid simultaneous bits in the long format of data. Scramblers have accomplish of uses in today's data communication protocols. On the other hand, those methods that are theoretical proposed are not feasible in the modern digital design due to many reasons such as slower data rate, increasing information, circuit hazards, uncountable hold-up etc. Therefore it is requisite for the modern digital design to have modified architecture to meet the required goal. We will recommend here modified scrambler design which is perfectly suitable for any industrial design.

Keywords: scrambler, descrambler, VHDL, and FPGA.

I. INTRODUCTION

In telecommunications, a scrambler is a device that transposes or inverts signals or otherwise encodes a message at the transmitter to make the message unintelligible at a receiver not equipped with an appropriately set descrambling device. While encryption usually refers to operations carried out in the digital domain, scrambling typically refers to operations carried out in the analog domain. Scrambling is consummate by the addition of components to the original signal or the changing of some important component of the original signal in order to make extraction of the original signal complex. To improve the degree of data security in a conventional Scrambler the number of stages of the shift register needs to be enhanced. This conversely increases error propagation. A uncomplicated method for ensuring security is to encrypt the data. The pseudo-noise (PN) key generation is of paramount importance

for any secure communication system. PN sequences base on Linear Feedback Shift Registers (LFSR) and non linear combination based implementations are simplest to give moderate level of security. Chaos base encryption techniques have proved fruitful, but complexity of such systems is important. The complex system generated is used to scramble incoming plain text. At the receiving end, the same code be generated and successfully used to decrypt the transmitted data. The ease of the circuit along with the complexity of the generated codes makes the circuit striking for secure message communication applications.

II. PROPOSED WORK

The entire operation is proposed using Modelsim and Xilinx blocks goes through three phases.

1. Architecture of Scrambler & Descrambler
2. Block diagram of Scrambler & Descrambler
3. Overview of Scrambler & Descrambler

a) Architecture of Scrambler & Descrambler

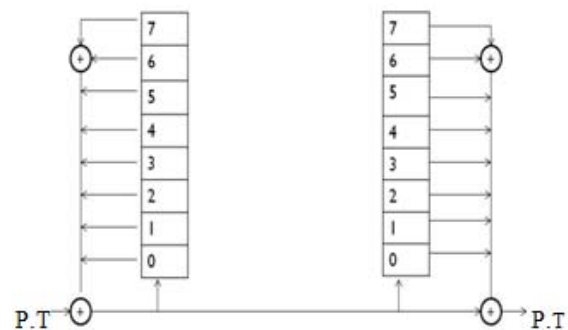


Figure 1 : Architecture of Scrambler and Descrambler

A data scramble & descramble are shown in fig. The scramble operates in the following manner. The initial shift register contents are random but prespecified and fixed to the same in both the scramble and descramble. The initial bit sequence of location 6 & 7 in the shift register X-OR is placed in shift register stage 0. The generated bit sequence is the sum with plain text, then it becomes the bit sequence is crypto word. As this bit is presented to the channel the contents of the shift register are shifted up one stage as follows: 7→out, 5→6, 4→5, 3→4, 2→3, 1→2.

The descramble operates as follows. The initial shift register contents are random but prespecified and fixed to the same in both the scramble and descramble.

Author ^α ^σ: Dept. of Electronics and Telecommunications
Dept. of Electronics and Telecommunications P.R. Pote (Patil) college
of Engineering and, Management, Amravati, India.
e-mails: naina0689@gmail.com, gauri.borkhade@gmail.com

The initial bit sequence of location 6 & 7 in the shift register X-OR is placed in shift register stage 0. The generated bit sequence is the sum with crypto word then it becomes the bit sequence is plain text. As this bit is presented to the channel the contents of the shift register are shifted up one stage as follows: 7→out, 5→6, 4→5, 3→4, 2→3, 1→2.

b) Block diagram Scrambler & Descrambler

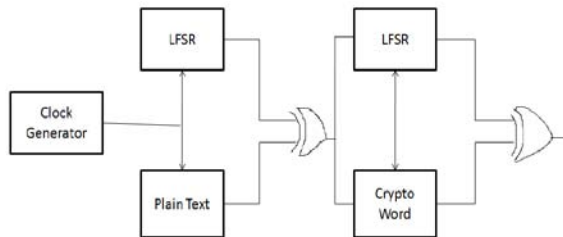


Figure 2 : Block diagram of scrambler And descrambler

Block diagram of scramble & descramble represented in Figure. Scrambler is performed in sequence X-OR the 8-bit plain text (D0-D7) character with the 8-bit (D0-D7) output of the LFSR. An output of the LFSR is XOR with plain text of the data to be processed. The LFSR and data register are then successively advanced and the output processing is repeated for D1 through D7.

Descrambler is performed in order XOR the 8-bit crypt word (D0-D7) character with the 8-bit (D0-D7) output of the LFSR. An output of the LFSR is XOR with crypto word of the data to be processed. The LFSR and data register are then consecutively advanced and the output processing is repeated for D1 through D7.

c) Overview of Scrambler and Descrambler

In the transmitter, a pseudorandom cipher sequence is added (modulo 2) to the data (or control) sequence to produce a scrambled data (or control) sequence.

In the receiver, the same pseudorandom cipher sequence is subtracted (modulo 2) from the scrambled data (or control) sequence to recover the transmitted data (or control) sequence, as illustrated in figure.

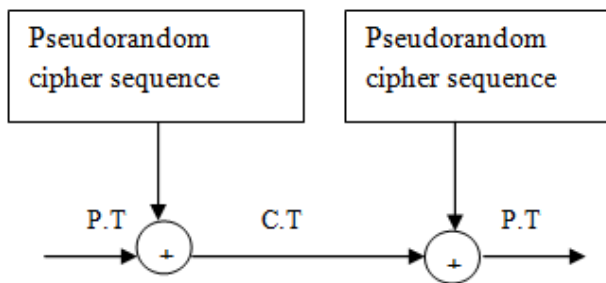


Figure 3 : Overview of scrambler and descrambler

III. RESULTS

The proposed Fpga implementation of various outputs is done using Modelsim and Xilinx. Both hardware and software implementation of various output is tabulated below.

a) For 8 bit Scrambler

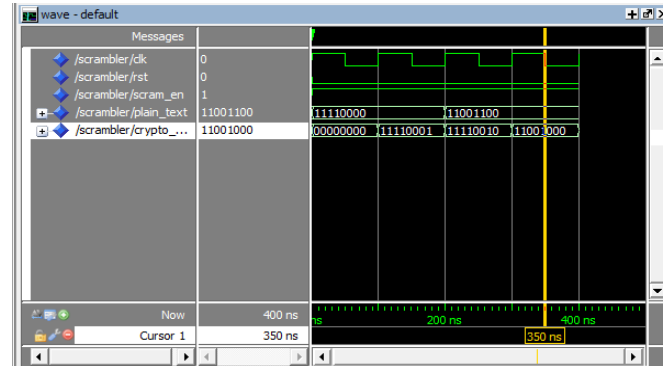


Figure 4 : Wave output for Scrambler

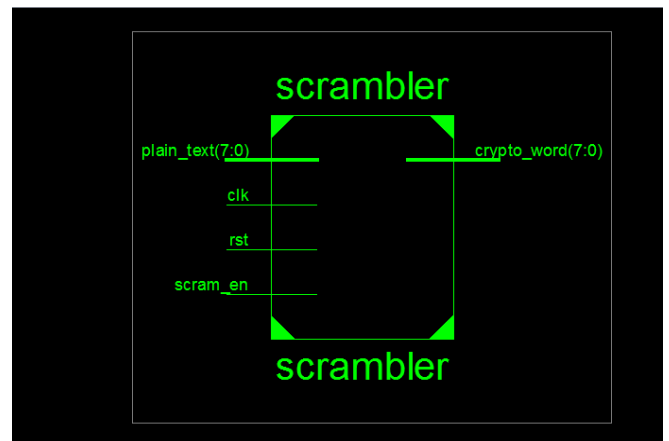


Figure 5 : RTL Schematic for Scrambler

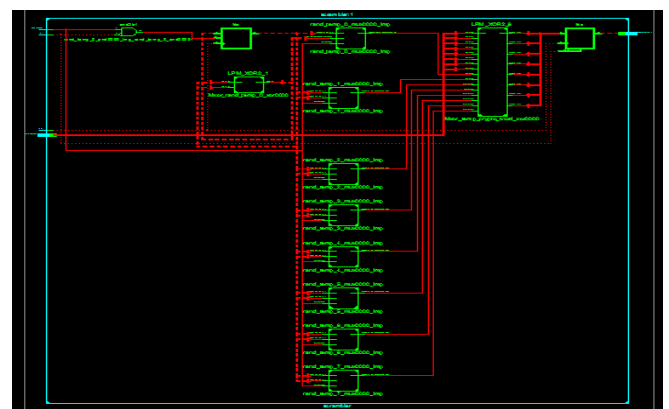


Figure 6 : Internal View of RTL Schematic for Scrambler

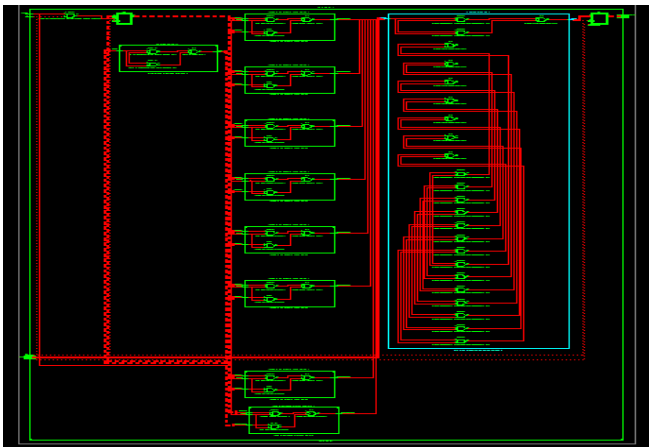


Figure 7 : Internal View of RTL Schematic for Scrambler

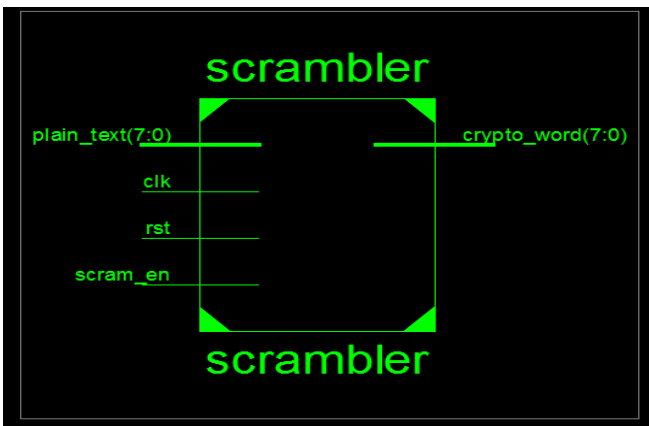


Figure 8 : View Technology Schematic for Scrambler

Device utilization summary:

Selected Device : 3s500efg320-4

Number of Slices:	9 out of 4656	0%
Number of Slice Flip Flops:	16 out of 9312	0%
Number of 4 input LUTs:	10 out of 9312	0%
Number of IOs:	19	
Number of bonded IOBs:	19 out of 232	8%
Number of GCLKs:	1 out of 24	4%

Figure 9 : Area for Scrambler

Timing summary for Scrambler

Minimum period= 3.424ns(Maximum Frequency 292.056MHz)

Throughput for 8 bit scrambler:- Maximum Freq*No of Bit/No of cycle

$$\begin{aligned}
 &= 292.056\text{MHz} * 8/1 \\
 &= 2336.448\text{MHz} \\
 &\sim 2.4\text{GHz}
 \end{aligned}$$

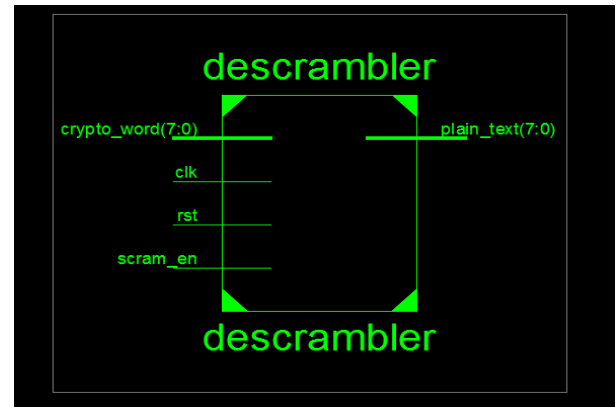


Figure 10 : RTL Schematic for Descrambler

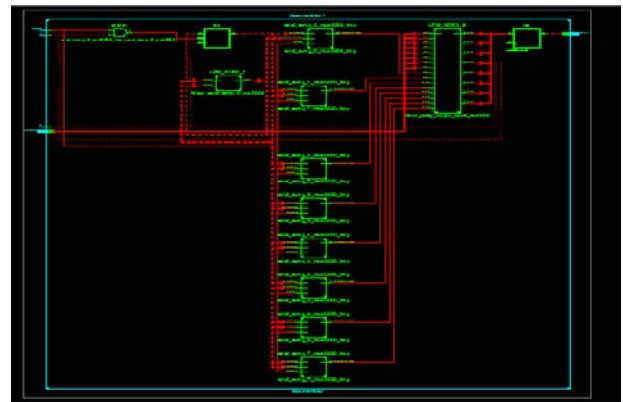


Figure 11 : Internal View of RTL Schematic for Descrambler

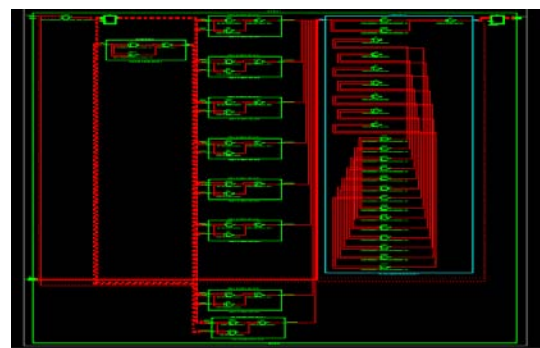


Figure 12 : Internal View of RTL Schematic for Descrambler

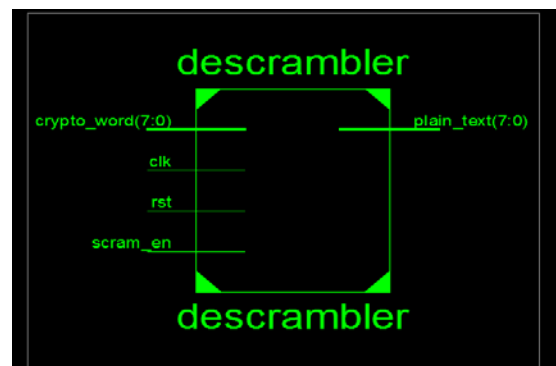


Figure 13 : View Technology Schematic for Descrambler

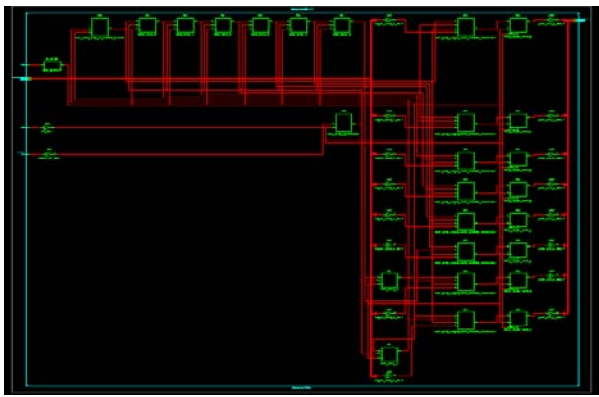


Figure 14 : Internal View Of View technology Schmatic for Descrambler

c) *Maximum Length polynomial for Descrambler*
For Enhanced Security using polynomial equation
 $X^7 + X^6 + X^4 + X^3 + 1 = 0$

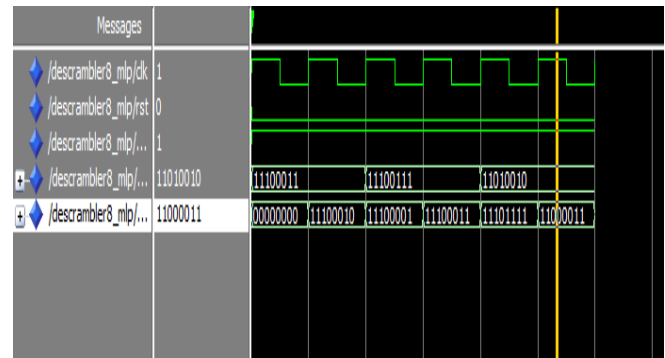


Figure 17 : Wave output for Maximum Length Polynomial for Descrambler

d) *For 16 Bit Scrambler*
Area for 16bit Scrambler

```
Device utilization summary:
-----
Selected Device : 3s500efg320-4

Number of Slices:          9 out of 4656  0%
Number of Slice Flip Flops: 16 out of 9312 0%
Number of 4 input LUTs:   10 out of 9312 0%
Number of IOs:            19
Number of bonded IOBs:    19 out of 232  8%
Number of GCLKs:          1 out of 24    4%
```

Figure 15 : Area for Scrambler

Timing summary for Scrambler

Minimum period= 3.424ns(Maximum Frequency 292.056MHz)

Throughput for 8 bit scrambler:- Maximum Freq*No of Bit/No of cycle

$$= 292.056\text{MHz} \times 8/1 \\ = 2336.448\text{MHz} \\ \sim 2.4\text{GHz}$$

b) *Maximum Length polynomial for Scrambler*
For Enhanced Security using polynomial equation
 $X^7 + X^6 + X^4 + X^3 + 1 = 0$

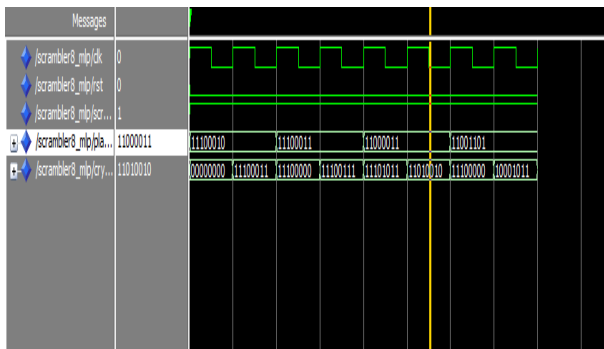


Figure 16 : Wave output for Maximum Length polynomial for Scrambler

```
Device utilization summary:
-----
Selected Device : 3s500efg320-4

Number of Slices:          37 out of 4656  0%
Number of Slice Flip Flops: 64 out of 9312 0%
Number of 4 input LUTs:   34 out of 9312 0%
Number of IOs:            67
Number of bonded IOBs:    67 out of 232  28%
Number of GCLKs:          1 out of 24    4%
```

Figure 18 : Area for 16Bit Scrambler

Timing Summary:

Minimum period=3.424ns (Maximum Frequency 292.056MHz)

e) *For 32 Bit Scrambler*
Area for 32 Bit Scrambler

```
Device utilization summary:
-----
Selected Device : 3s500efg320-4

Number of Slices:          37 out of 4656  0%
Number of Slice Flip Flops: 64 out of 9312 0%
Number of 4 input LUTs:   34 out of 9312 0%
Number of IOs:            67
Number of bonded IOBs:    67 out of 232  28%
Number of GCLKs:          1 out of 24    4%
```

Figure 19 : Area for 32Bit Scrambler

Timing Summary:

Minimum period=3.424ns(Maximum Frequency 292.056 MHz)

f) For 16 Bit Scrambler

Area for 16bit Scrambler

```
Device utilization summary:
-----
Selected Device : 3s500efg320-4

Number of Slices:          37 out of 4656   0%
Number of Slice Flip Flops: 64 out of 9312   0%
Number of 4 input LUTs:    34 out of 9312   0%
Number of IOs:              67
Number of bonded IOBs:      67 out of 232   28%
Number of GCLKs:            1 out of 24     4%
```

Figure 20 : Area for 16Bit Scrambler

Timing Summary:

Minimum period=3.424ns(Maximum Frequency 292.056MHz)

g) For 32 Bit Scrambler

Area for 32 Bit Scrambler

```
Device utilization summary:
-----
Selected Device : 3s500efg320-4

Number of Slices:          37 out of 4656   0%
Number of Slice Flip Flops: 64 out of 9312   0%
Number of 4 input LUTs:    34 out of 9312   0%
Number of IOs:              67
Number of bonded IOBs:      67 out of 232   28%
Number of GCLKs:            1 out of 24     4%
```

Figure 21 : Area for 32Bit Scrambler

Timing summary:

Number Of Bit	Maximum Frequency
8 Bit	292.056MHz
16 Bit	292.056MHz
32 Bit	292.056 MHz
8 Bit	449.438MHz

Timing Summary:

Minimum period=3.424ns(Maximum Frequency 292.056 MHz)

h) Implementation

```
Device utilization summary:
-----
Selected Device : 3s500efg320-4

Number of Slices:          37 out of 4656   0%
Number of Slice Flip Flops: 64 out of 9312   0%
Number of 4 input LUTs:    34 out of 9312   0%
Number of IOs:              67
Number of bonded IOBs:      67 out of 232   28%
Number of GCLKs:            1 out of 24     4%
```

Figure 22 : Device utilization Summary

Timing Summary:

Speed Grade: -4

Minimum period: 2.225ns (Maximum Frequency: 449.438MHz)
 Minimum input arrival time before clock: 4.366ns
 Maximum output required time after clock: 4.283ns
 Maximum combinational path delay: No path found

Figure 23 : Timing Summary

IV. CONCLUSION

A new modified scheme for complex PN-code based data scrambler and descrambler has been presented. A scrambler & descrambler accepts information in intelligible form and through intellectual transformation assure data quality with fastest rate without any error or dropping occurrence. We used our proposed and modified design in our present universal serial bus architecture. Moreover, this current design is very efficient, more securable, high speed, low power and lower area used & it has lots of scope to improved.

V. ACKNOWLEDGMENT

I have taken efforts in this project. Though it would not have been possible without the kind support and help of many individuals and organizations. I would like to make bigger my sincere thanks to all of them. I am highly indebted to Prof .G. P. Borkhade for their guidance and constant supervision as well as for providing necessary information regarding the project & also for their support in completing the project.

I would like to convey my pleasure towards my guide & member of Electronics and Telecommunication Engineering for their kind co-operation and encouragement which help me in completion of this project.

I would like to express my special gratitude and thanks to institute persons for giving me such attention and time. My thanks and appreciations also go to my colleague in developing the project and people who have willingly helped me out with their abilities.

REFERENCES RÉFÉRENCES REFERENCIAS

1. Rajib Imranand Monirul Islam, 2013, Indurtial Modified Digital scrambler and descrambler system.
2. Davinder Pal Sharma, 2013, Data scrambler of ultra-wide band communication system.
3. G.M. Bhat, M. Mustafa, Shabir Ahmad, and Javaid Ahmad, 2009, VHDL modeling and simulation of data scrambler and descrambler for secure data communication.

4. Sharma, D. P. Singh J, Simulation and spectral analysis of the scrambler for 56Kbps modem. The Journal of Signal Processing Systems. 67, 269-277 (2012).
5. Sharma, D. P. Singh J. DSP based implementation of scrambler for 56Kbps modem. Signal Processing – An International Journal. 4, 85-96 (2010).
6. Hethan Kumar, M Praveen Kumar Y G, Dr. M volume 3 Issue 4, April 2014, “Design and implementation of Logical Scrambler Architecture for OTN Protocol”.
7. Xiao-Bei Liu, Soo Ngee Koh, Chee-Cheon Chui, and Xin-Wen Wu, “A Study of Reconstruction of Linear Scrambler using Dual Words of Channel Encoder” March 2013.





GLOBAL JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY: A
HARDWARE & COMPUTATION

Volume 15 Issue 2 Version 1.0 Year 2015

Type: Double Blind Peer Reviewed International Research Journal

Publisher: Global Journals Inc. (USA)

Online ISSN: 0975-4172 & Print ISSN: 0975-4350

The Implementation of DMA Controller on Navigation baseband SoC

By Abdulraqueeb Alnabihi & Prof Liu Yijun

Guangdong University of Technology, China

Abstract- This article discusses the architecture design of DMA controller on high performance GPS receiver based on RTEMS. We achieve the optimal integration of DMA IP and navigation baseband system. We designed the hardware architecture of DMA IP and make full use of hardware performance with the idea of multiplexing. We use register and FIFO buffer to achieve read-write control. And we design the DMA controller with Verilog HDL. Finally we verify the design on Altera Cyclone4 FPGA. The result demonstrates that DMA controller can ease the CPU's burden and shorten the acquisition & tracking time thus improving the performance of the whole system.

Keywords: DMA controller; AHB bus; GPS(Global Positioning System); System on Chip; FPGA (Field –Programmable Gate Array)

GJCST-A Classification : B.3.2 B.5.1



THE IMPLEMENTATION OF DMA CONTROLLER ON NAVIGATION BASEBAND SOC

Strictly as per the compliance and regulations of:



RESEARCH | DIVERSITY | ETHICS

The Implementation of DMA Controller on Navigation baseband SoC

Abdulraqueb Alnabihi ^α & Prof. Liu Yijun^σ

Abstract- This article discusses the architecture design of DMA controller on high performance GPS receiver based on RTEMS. We achieve the optimal integration of DMA IP and navigation baseband system. We designed the hardware architecture of DMA IP and make full use of hardware performance with the idea of multiplexing. We use register and FIFO buffer to achieve read-write control. And we design the DMA controller with Verilog HDL. Finally we verify the design on Altera Cyclone4 FPGA. The result demonstrates that DMA controller can ease the CPU's burden and shorten the acquisition & tracking time thus improving the performance of the whole system.

Keywords: DMA controller; AHB bus; GPS(Global Positioning System); System on Chip; FPGA (Field –Programmable Gate Array).

I. INTRODUCTION

DMA controller is the core component of the SoC. Through controlling the data transfer from memory to peripheral independently, it can greatly alleviate the CPU's burden and improve the efficiency of data processing. Therefore the design of DMA controller directly affects the overall performance of SoC chip. Similar to the design of other integrated circuit IP, DMA design should also consider the reusability of DMA IP and balance the operation speed and circuit area, at the same time achieving good integration with SoC. According to the design goal, the domestic and foreign scholars have conducted in-depth research. Literature [2] designs a configurable multi-channel DMA controller, but it doesn't consider the size of FIFO BUFFER, which may causes the waste of resources. The design in literature [3] can alleviate the CPU's burden and improve the data transmission rate between peripherals, but it is only a single channel structure thus lack of generality. Literature [4] designs a specific data path for DMA data transfer. Although it avoids the limitation of AHB bus, achieving parallel transmission of multi-path data, but making the internal SoC bus timing complicated and consideration must be given to the arbitration of memory R/W between bus and special data bus. In this paper, according to the large amount of data of high real-time navigation applications, we design the DMA controller implementation in navigation baseband SoC. The design shortens the first positioning time of the chip and improves its whole performance.

Author ^α: Guangdong, Guangzhou, China.

Author ^σ: School of Computer Science and Technology, Guangdong University of Technology 510006. e-mail: 970133919@qq.com

Our Navigation Baseband SoC chooses LEON3 processor from Aeroflex Gaisler Corporation and it's based on SPARC V8 architecture. The on-chip bus is the AMBA2.0 of ARM Corporation, which ensuring the data transmission flowing. In order to realize the fast acquisition and tracking of navigation signal, the chip has 2 built-in acquisition and tracking module. The chip uses the on-chip SRAM controller to realize the real-time data storage. We use RTEMS 4.10 as the operation system, making the system with good real-time performance. At the same time, the driver program of RTEMS relative to LINUX and other embedded operating system is simpler.

II. DESIGN OF THE SYSTEM ARCHITECTURE

The whole system architecture uses classical SoC architecture based on AMBA bus, as Figure 1 shows. AMBA2.0 AHB bus connects on-chip CPU, memory and other high-speed devices. It achieves W/R operation between SRAM, SDRAM and FLASH ROM through memory controller. Acquisition and tracking module are connected to the DMA controller, and they store the generated data to RAM through DMA module.

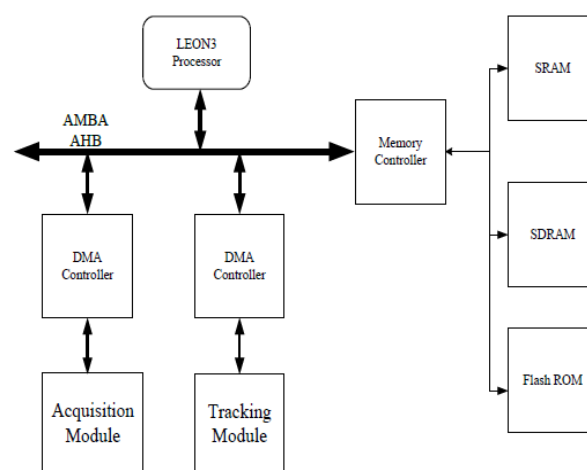


Fig.1 : Navigation Baseband Architecture

III. DESIGN OF THE SYSTEM ARCHITECTURE

As Figure 2 shows, DMA module is mainly made of buffer, interrupt generating logic module, register, arbitration module, bus W/R module and corresponding interfaces. Acquisition and tracking module connects into DMA module through parallel data interfaces, and store the generating data to each

The DMA controller has three groups of control registers. Each group includes a 32 bit control register, source address register, destination address register and data register. It achieves memory W/R operation through Memory Mapping by AHB bus and AMBA bus controller. Each signal generated by control register is sent to fixed priority arbitrator for arbitration first, the arbiter controls a mux to generate corresponding control signal.

IV. DESIGN OF THE FIFO MODULE

IV. DESIGN OF THE FIFO MODULE

The structure of the FIFO is shown as Figure 3. Because it uses different clocks to read and write, so we use Dual-port RAM as memory module. The word length and depth of storage of dual port RAM is according to the AD parameter RF front-end sampling. In order to solve the problem of metastable clock domain data transmission, FIFO read and write pointer is encoded with the gray code. Through the multi-level register transfer DMA controller can reduce the probability of metastable.

gray code. The first two bits are not the same. When the FIFO is full the two LSB gets the same. When the pointer is exactly equal, FIFO is empty.

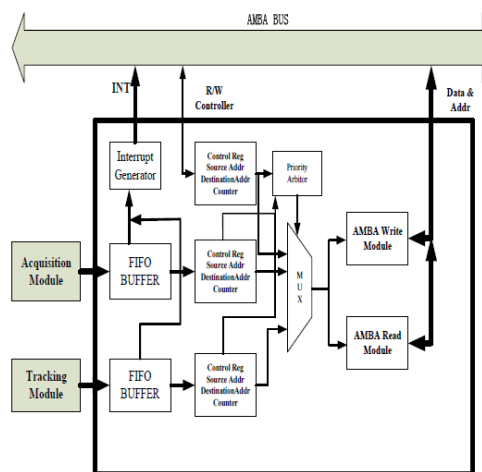


Fig.3 : FIFO module architecture

V. THE WORKFLOW OF DMA CONTROLLER

Initially, the bus is idle, waiting for the DMA transmission request. When the control register Read or Write bit is 1, the DMA controller goes into corresponding reading or writing ready state. According to the setting value of the counter, DMA then decided to adopt the single or burst mode to read and write bus. After read and write data process the controller goes into finish state. Finally it determines whether goes into the read and write cycle according to the value of control register. The workflow is shown as Figure4.

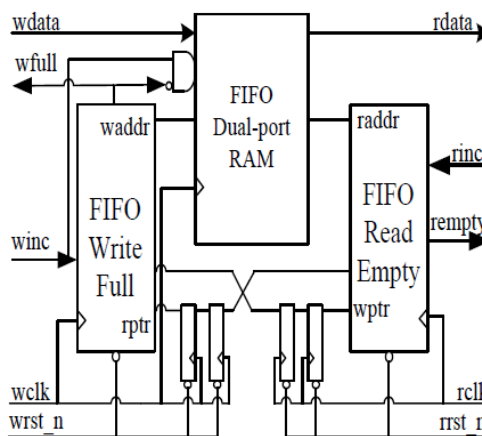


Fig. 4 : FSM Diagram

VI. THE SOFTWARE AND HARDWARE CO-VERIFICATION OF DMA IP

© 2015 Global Journals Inc. (US)

synthesize. We simulate the data transmission process by software interruption on RTEMS. Figure 5 shows the real-time read-burst mode waveform through Altera's Signal Tap on FPGA. We can see that DMA module successfully achieves data R/W in burst mode.



Fig. 5 : read burst module verification waveform

VII. APPLICATION

In order to investigate the effect of DMA solution brings, in this paper, we research in DMA and non DMA (FIFO interrupt) two cases the first positioning time (TTFF) baseband SoC requires. Through the test to different S/N ratio of the received signal and the dynamic condition, we can get the experiment data and conclusion.

a) Experiment Design

The first positioning time of navigation baseband SoC is also called cold start time. The system powers on and initialize without any historical information under a cold start condition. And then it attempts to locate and lock the satellite. Because of the lack of prior information, it will take a long time. System uses a method similar to the polling, locking the signals from all the satellites.

The system performs four steps from power on to the output of position data: system initialization – acquisition – tracking – position calculation. At the same time it interact with the user. In the four step, the processor is mainly responsible for initializing the operating system, position solution and human-computer interactive task. The DMA controller can be used instead of the processor, responsible for store the data generated by acquisition and tracking module into RAM. Therefore the processor can focus on the position solution and human-computer interactive task and the first positioning time (TTFF) is shortened.

In order to comprehensively study the performance of the system, the test experiment selects three different IF GPS signal as signal source. The GPS S/N ratio is 42db-Hz under the first condition. The system acceleration is 0g. The GPS S/N ratio is 42db-Hz under the first condition. But the system acceleration is 20g. The GPS S/N ratio is 34db-Hz under the first

condition. The system acceleration is 0g. The test results is shown in Table 1 and Table 2.

Table 1: Non-DMA Mode TTFF statistics

GPS S/N dB/Hz	system acceleration / m/s ²	TTFB / s
42	0	26.4
42	20	30.2
34	0	28.6

Table 2 : DMA Mode TTFF statistics

GPS S /N dB/Hz	system acceleration / m/s ²	TTF / s
42	0	25.3
42	20	28.1
34	0	26.8

b) Data analysis

We can draw from Table 1 and Table 2 that: Under the first condition, the system is under a low dynamic environment with high signal-to-noise ratio, since that the TTFF both in DMA and non-DMA mode is relatively short. And the DMA mode takes 1.1 s less time than non-DMA mode. So, DMA controller sets CPU free from busy data transforming task, making it to have more free time to perform other tasks and shortening the TTFF.

Under the second condition, the system is under a high dynamic extreme environment, which greatly increases the acquisition and tracking difficulty. Here the DMA mode takes 2.1 s less time than non-DMA mode. Compared to the first condition, Using DMA method to shorten the TTFF effect is more obvious.

Under the third condition, the system is under a environment with low signal-to-noise ratio, which also increases the acquisition and tracking difficulty. TTFF in DMA mode is 28.6s, in non-DMA mode is 26.8s, which is both longer than the first condition. Here the DMA mode takes 1.8 s less time than non-DMA mode. Implementing DMA also has the obvious effect on reducing TTFF.

The experiment proves that: The DMA controller can effectively reduce navigation baseband SoC CPU's load, shorten TTFE.

VIII. CONCLUSION

SoC technology has developed rapidly in recent years, represent the future development trend of IC. As a complex system composed by multiple IP, the single component design of GPS baseband SoC must fully consider the overall system hardware and software

integration in order to play the best performance of the whole system. In this paper, the DMA IP from the system view, it uses FIFO BUFFER and interrupting mechanism to realize the good combination of acquisition and tracking module. With the control register group developer can easily program the DMA controller under the RTMES real-time operating system. The FPGA Experiments show that this design can realize the data read and write control based on AMBA bus, reduce the burden of CPU.

14. Clifford E. Cummings. Simulation and Synthesis Techniques for Asynchronous FIFO Design [J]. Snug, 2002, 01:2-23.
15. Xia Yu-wen. Verilog Digital System Design Tutorial (2nd Edition) [M]. BEIHANG UNIVERSITY PRESS 2008.

REFERENCES RÉFÉRENCES REFERENCIAS

1. Bai Zhong-ying. Principle of Computer Organization [M]. Beijing: Science Press 2005.
2. CHEN Shuang-yan, WANG Dong-hui, HOU Chao-huan. Design and Implementation of a Configurable Multi-Channel DMA Controller Based on SystemC [J]. Microelectronics & Computer, 2007, 24 (5): 48-51.
3. Shi Bing, Ding Zhi-gang, Zhang Wei-hong. NAND Flash DMA Application Based on PXA3xx Processor [J]. Journal of Computer Applications, 2009, 29(8): 2136-2138.
4. Zhang Lu-yu, Li Li, Pan Hong-bing, Wang Kun, Li Wei. Design of a multi-interface DMA controller based on SoC [J]. Electronic Measurement Technology, 2014, 37(9): 32-36.
5. Sun Zhong-xiu, Fei Xiang-lin, Luo Bin. Operating System Course [M]. 3rd Edition. Higher Education Press, 2003.
6. Xie Yong, Shen Ming, Zheng Jian-hong. The Hardware Application of High Performance DMA Controller in AMBA Bus Architecture [J]. Journal of Chongqing Institute of Technology: Computer and Automation Column, 2006, 20(8): 72-74.
7. Hao Jun. The Simulation and Verification of DMA Controller [D]. Master Dissertation. Xidian University. 2013.
8. Geng Jian-bo. The Design and Verification of DMA Controller based on AMBA Bus [D]. Master Dissertation. Xidian University. 2013.
9. Zhao Qiang. The Design of DMA Controller based on AHB Bus Specification [D]. Master Dissertation. Xidian University. 2014.
10. Wei Yun. The Design of DMA Platform for PCIe Bus Based on FPGA [D]. Wuhan University of Technology. 2013.
11. Li Mu-guo, Huang Ying, Liu Yu-zhi. The Design of DMA for PCIe Bus Based on FPGA [J]. Computer Measurement & Control, 2013, 01: 46-49.
12. Aeroflex Gaisler. GRLIB IP Library User's Manual [Z]. Sweden: Aeroflex Gaisler Ltd. 2013. 10-13
13. ARM L td. AMBA Bus Specification [Z]. UK: ARM Ltd., 1999. 2-18.



GLOBAL JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY: A
HARDWARE & COMPUTATION

Volume 15 Issue 2 Version 1.0 Year 2015

Type: Double Blind Peer Reviewed International Research Journal

Publisher: Global Journals Inc. (USA)

Online ISSN: 0975-4172 & Print ISSN: 0975-4350

GSM based Agriculture Monitoring and Controlling System

By DR. R. Bulli Babu, CH. JonathanSoumith, T. Cherishma Sri Lakshmi
& R. Keshav Rao

Kluniversity, India

Abstract- This paper proposes a farming environment observing framework for checking data concerning an outside by using Wireless Sensor Network (WSN) innovation. The proposed rural environment observing server framework gathers natural and soil data on the outside through WSN-based ecological and soil sensors. In this paper we are using sensors as soil moisture sensor and temperature sensor. These sensors help the field to control the water level and also temperature. Here we are using wireless sensor network as GSM (global system for mobile communication).

Keywords: AT89C51 microcontroller; SHT-71; wireless sensor networks; temperature sensor; soil moisture sensor.

GJCST-A Classification : B.4.1 C.2.1



Strictly as per the compliance and regulations of:



GSM based Agriculture Monitoring and Controlling System

DR. R. Bulli Babu ^α, CH. JonathanSoumith ^σ, T. Cherishma Sri Lakshmi ^ρ & R. Keshav Rao ^ω

Abstract- This paper proposes a farming environment observing framework for checking data concerning an outside by using Wireless Sensor Network (WSN) innovation. The proposed rural environment observing server framework gathers natural and soil data on the outside through WSN-based ecological and soil sensors. In this paper we are using sensors as soil moisture sensor and temperature sensor. This sensors help the field to control the water level and also temperature. Here we are using wireless sensor network as GSM (global system for mobile communication).

Keywords: AT89C51 microcontroller; SHT-71; wireless sensor networks; temperature sensor; soil moisture sensor.

I. INTRODUCTION

As of late, developments in data correspondence engineering have been quickening the merging between diverse commercial enterprises. The meeting and reconciliation of IT with agrarian engineering is required to be a region that could expand the included quality and benefit of agribusiness by applying the pervasive innovation to the horticultural segment which is an essential industry. To effectively develop such a u-farming environment, the improvement of crucial universal innovation improved for agribusiness, for example, sensor equipment, middleware stages, steering conventions and application administrations for agrarian situations is required. Illustrations of the union of universal innovation with horticulture, which is an essential industry, on a trial premise exist, for example, the utilization of sensor hubs in vine society destinations and applications of pervasive engineering in domesticated animals cultivating locales, and the engineering has step by step started to show up in other little territories like the increment of generation and the change of value at different farming locales.

An agrarian environment checking framework gives ecological observing administrations and office controlling administrations, and therefore keeps up the harvest becoming environment in an ideal status. This framework additionally enhances the comfort and gainfulness of clients. In any case, existing horticultural observing frameworks are generally connected and

used in shut agrarian situations, for example, nurseries, steers sheds, and so forth. as it is hard to apply rural checking frameworks in outside areas, for example, paddies, fields, plantations, and so on due to an absence of IT framework. What's more, when clients need to check the observed data in existing observing frameworks, the client should physically check the status through introduced sensors or terminals introduced in the agribusiness offices. With a specific end goal to take care of these issues, it is important to create an agrarian environment observing framework that can screen ecological data and soil data in remote area and can be utilized as a part of rural situations which need base. This paper proposes a rural environment observing server framework to screen data on the outside by using WSN (remote sensor system) engineering, which is one of such pervasive advances.

On the off chance that the proposed farming environment observing server framework is connected to an agrarian environment, ecological and soil data could be checked even at a remote site, and it is normal that this would help expanded product yields and the change of value in the rural field by supporting makers' choice making about harvest creation through the investigation on the gathered data. The rural environment checking server framework proposed in this paper gathers ecological data, for example, luminance, temperature, humidity, ph, Co2 and so on which influence development of harvests and soil data through the WSN natural sensors and soil sensors introduced outdoors, and these are interfaced to gsm module to alarm the ranchers. Moreover, the server framework is situated up to utilize force supplied through sun oriented cells so it could be utilized as a part of horticultural situations with lacking force foundation.

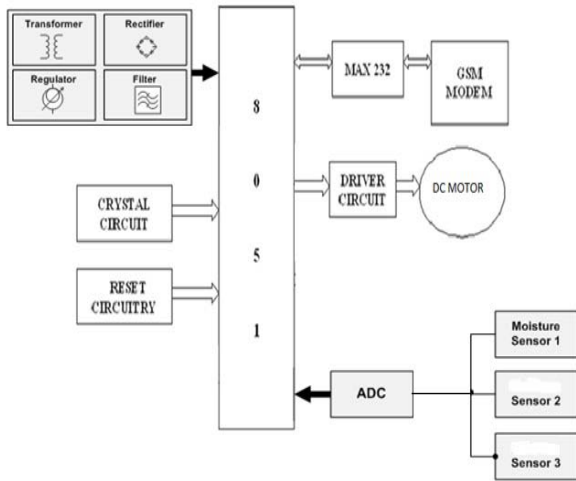
II. SYSTEM ARCHITECTURE

The primary segments in the circuit are microcontroller At89c51, Humidity sensor, soil sensor and GSM modem. The At89c51 comprises of four ports which can be utilized for both include and yield. The circuit utilizes port1 as data, port2 as in-out, port3 and port0 as yield ports. A switch is associated with pin to reset the circuit. The dirt sensor is utilized for recognizing the dampness as a part of the dirt. It is associated with port2 which go about as info to the microcontroller.

Author α: Associate Professor IV/IV B.Tech.

e-mail: bullibabu@kluniversity.in

Author σ ρ ω: Electronics and Computer Engineering Kluniversity, Guntur, A.P, India. e-mails: soumithsmith@gmail.com, cherishmasrilakshmi@gmail.com, keshav372@gmail.com



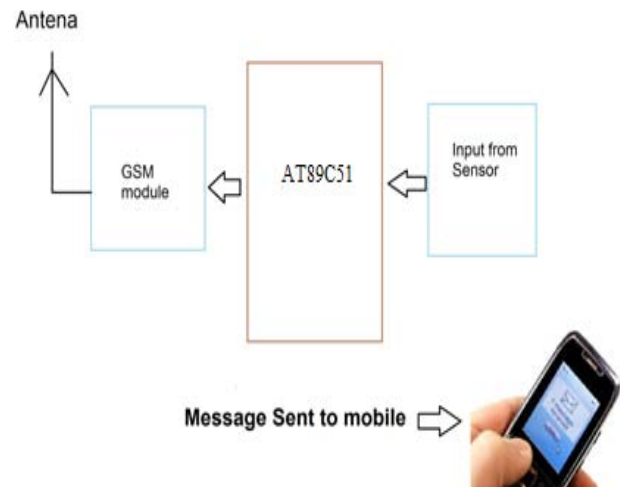
It delivers zero yield when there is dampness in the dirt and when there is no dampness it creates a yield signal which is transmitted to the microcontroller. The Environment sensor is utilized for sensing the temperature. The microcontroller acknowledges the signs which are in computerized form. GSM modem is joined with port3 which is the yield. It is interfaced to the microcontroller utilizing Max232. It is utilized to transmit the alarm message to the rancher. The correspondence in the middle of microcontroller and GSM circuit is carried out by utilizing serial correspondence. For this it utilizes Rs232 serial port. The Rs232 rationale levels and microcontroller rationale levels are not equivalent. So to match the rationale levels of microcontroller and Rs232 rationale levels Max232 is utilized to interface GSM modem.

At the port0 there are no draws up resistors inbuilt in it to deliver 5v supply. In this way, the outside forces up resistors are joined at port0 to create 5v supply. At the point when the signs from soil sensor and temperature sensor are high, The signs from the microcontroller additionally to the GSM modem as indicated by the product code dumped in the

microcontroller. At that point GSM modem then transmits the caution message to the approved persons.

III. METHODOLOGY

The main aim of this paper to atomize the irrigation system for social welfare of agriculture system. In this paper we basically observing on the some of applications like.



To continuously measuring monitor in the soil with the help of soil moisture sensor and also measuring the level of the water content using sensor and also check the temperature, humidity with the help of temperature sensor .By this if any level of water or temperature changes takes place then immediately the message is pass to the framer with the help of GSM. This GSM transmit the full of information about the field to the framer to the framer In this paper we are using keil as software and the Hard ware as the excepted output is as follows



IV. SOFTWARE DESCRIPTION

As this task is installed based undertaking, the product code is composed in implanted c which is assembled debugged and tried. It is re-enacted with the assistance of micro vision keil which control the execution of inserted c programs. Software utilized is keil programming for implanted c programming. It is conceivable to make the source documents in a word processor, for example, Notepad, run the Compiler on every C source document, indicating a rundown of controls, run the Assembler on every Assembler source record, defining an alternate rundown of controls, run either the Library Manager or Linker (again pointing out a rundown of controls) lastly running the Object-HEX.

Converter to change over the Linker yield record to an Intel Hex File. When that has been finished the Hex File can be downloaded to the target fittings and debugged. Then again KEIL can be utilized to make source records; consequently assemble, connection and clandestine utilizing choices set with a simple to utilize client interface lastly mimic or perform debugging on the equipment with access to C variables and memory. Unless you need to utilize the tolls on the charge line, the decision is clear. KEIL Greatly improves the procedure of making and testing an implanted application.

V. CONCLUSION

This paper deals with the design and development of moisture detection, temperature monitoring and alerting system for farmers. The GSM modem provides the information in case of emergency. This is a reliable and efficient system for alerting and intimation to the farmers through GSM mobile communication.

REFERENCES RÉFÉRENCES REFERENCIAS

1. Beaulah, S.A., Chalabi, Z.S., Randle, D.G., 1998. A real-time knowledge-based system for intelligent monitoring in complex, sensor-rich environments. *Comput. Electron. Agric.* 21 (1), 53–68.
2. Butler, Z., Corke, P., Peterson, R., Rus, D., 2004. Virtual fences for controlling cows. In: *Proceedings of the 2004 IEEE International Conference on Robotics and Automation*, New Orleans, LA, USA, April 26–May 1, pp. 4429–4436.
3. Q.Hao and Z.Song, "The status and development of the intelligent Automatic meter reading system," *China Science and Technology Information*, no.19, pp.72, Oct 2005.
4. Wei Xiaolong. Interface technology and examples of design of system based on MSP430 series single chip. Beijing: Beijing university of Aeronautics and Astronautics Press. 2002.

5. Hamoud, G. Chen, R.-L. Bradley, "Risk assessment of power systems SCADA," *IEEE Power Engineering Society General Meeting*, 2003, Vol.2, Jul. 2003.
6. Study on an Agricultural Environment Monitoring Server System using Wireless Sensor by Networks Jeonghwan Hwang, Changsun Shin and Hyun Yoe *
7. Detection & Current Weather Conditions; Available online: <http://alert.udcd.org> (accessed on 3 December 2010).
8. UDFC ALERT System Real-Time Flood 11. Pierce, F.J.; Elliott, T.V. Regional and on-Farm Wireless Sensor Networks for Agricultural Systems in Eastern Washington. *Computer. Electron. Agric.* 2008, 61, 32-43.



GLOBAL JOURNALS INC. (US) GUIDELINES HANDBOOK 2015

WWW.GLOBALJOURNALS.ORG

FELLOWS

FELLOW OF ASSOCIATION OF RESEARCH SOCIETY IN COMPUTING (FARSC)

Global Journals Incorporate (USA) is accredited by Open Association of Research Society (OARS), U.S.A and in turn, awards “FARSC” title to individuals. The 'FARSC' title is accorded to a selected professional after the approval of the Editor-in-Chief/Editorial Board Members/Dean.



- The “FARSC” is a dignified title which is accorded to a person’s name viz. Dr. John E. Hall, Ph.D., FARSC or William Walldroff, M.S., FARSC.

FARSC accrediting is an honor. It authenticates your research activities. After recognition as FARSC, you can add 'FARSC' title with your name as you use this recognition as additional suffix to your status. This will definitely enhance and add more value and repute to your name. You may use it on your professional Counseling Materials such as CV, Resume, and Visiting Card etc.

The following benefits can be availed by you only for next three years from the date of certification:



FARSC designated members are entitled to avail a 40% discount while publishing their research papers (of a single author) with Global Journals Incorporation (USA), if the same is accepted by Editorial Board/Peer Reviewers. If you are a main author or co-author in case of multiple authors, you will be entitled to avail discount of 10%.

Once FARSC title is accorded, the Fellow is authorized to organize a symposium/seminar/conference on behalf of Global Journal Incorporation (USA). The Fellow can also participate in conference/seminar/symposium organized by another institution as representative of Global Journal. In both the cases, it is mandatory for him to discuss with us and obtain our consent.



You may join as member of the Editorial Board of Global Journals Incorporation (USA) after successful completion of three years as Fellow and as Peer Reviewer. In addition, it is also desirable that you should organize seminar/symposium/conference at least once.

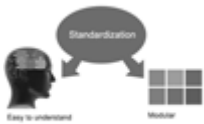
We shall provide you intimation regarding launching of e-version of journal of your stream time to time. This may be utilized in your library for the enrichment of knowledge of your students as well as it can also be helpful for the concerned faculty members.





The FARSC can go through standards of OARS. You can also play vital role if you have any suggestions so that proper amendment can take place to improve the same for the benefit of entire research community.

As FARSC, you will be given a renowned, secure and free professional email address with 100 GB of space e.g. johnhall@globaljournals.org. This will include Webmail, Spam Assassin, Email Forwarders, Auto-Responders, Email Delivery Route tracing, etc.



The FARSC will be eligible for a free application of standardization of their researches. Standardization of research will be subject to acceptability within stipulated norms as the next step after publishing in a journal. We shall depute a team of specialized research professionals who will render their services for elevating your researches to next higher level, which is worldwide open standardization.

The FARSC member can apply for grading and certification of standards of their educational and Institutional Degrees to Open Association of Research, Society U.S.A. Once you are designated as FARSC, you may send us a scanned copy of all of your credentials. OARS will verify, grade and certify them. This will be based on your academic records, quality of research papers published by you, and some more criteria. After certification of all your credentials by OARS, they will be published on your Fellow Profile link on website <https://associationofresearch.org> which will be helpful to upgrade the dignity.



The FARSC members can avail the benefits of free research podcasting in Global Research Radio with their research documents. After publishing the work, (including published elsewhere worldwide with proper authorization) you can upload your research paper with your recorded voice or you can utilize chargeable services of our professional RJs to record your paper in their voice on request.

The FARSC member also entitled to get the benefits of free research podcasting of their research documents through video clips. We can also streamline your conference videos and display your slides/ online slides and online research video clips at reasonable charges, on request.





The FARSC is eligible to earn from sales proceeds of his/her researches/reference/review Books or literature, while publishing with Global Journals. The FARSC can decide whether he/she would like to publish his/her research in a closed manner. In this case, whenever readers purchase that individual research paper for reading, maximum 60% of its profit earned as royalty by Global Journals, will be credited to his/her bank account. The entire entitled amount will be credited to his/her bank account exceeding limit of minimum fixed balance. There is no minimum time limit for collection. The FARSC member can decide its price and we can help in making the right decision.

The FARSC member is eligible to join as a paid peer reviewer at Global Journals Incorporation (USA) and can get remuneration of 15% of author fees, taken from the author of a respective paper. After reviewing 5 or more papers you can request to transfer the amount to your bank account.



MEMBER OF ASSOCIATION OF RESEARCH SOCIETY IN COMPUTING (MARSC)

The ' MARSC ' title is accorded to a selected professional after the approval of the Editor-in-Chief / Editorial Board Members/Dean.

The "MARSC" is a dignified ornament which is accorded to a person's name viz. Dr. John E. Hall, Ph.D., MARSC or William Walldroff, M.S., MARSC.



MARSC accrediting is an honor. It authenticates your research activities. After becoming MARSC, you can add 'MARSC' title with your name as you use this recognition as additional suffix to your status. This will definitely enhance and add more value and reputé to your name. You may use it on your professional Counseling Materials such as CV, Resume, Visiting Card and Name Plate etc.

The following benefits can be availed by you only for next three years from the date of certification.



MARSC designated members are entitled to avail a 25% discount while publishing their research papers (of a single author) in Global Journals Inc., if the same is accepted by our Editorial Board and Peer Reviewers. If you are a main author or co-author of a group of authors, you will get discount of 10%.

As MARSC, you will be given a renowned, secure and free professional email address with 30 GB of space e.g. johnhall@globaljournals.org. This will include Webmail, Spam Assassin, Email Forwarders, Auto-Responders, Email Delivery Route tracing, etc.





We shall provide you intimation regarding launching of e-version of journal of your stream time to time. This may be utilized in your library for the enrichment of knowledge of your students as well as it can also be helpful for the concerned faculty members.

The MARSC member can apply for approval, grading and certification of standards of their educational and Institutional Degrees to Open Association of Research, Society U.S.A.



Once you are designated as MARSC, you may send us a scanned copy of all of your credentials. OARS will verify, grade and certify them. This will be based on your academic records, quality of research papers published by you, and some more criteria.

It is mandatory to read all terms and conditions carefully.



AUXILIARY MEMBERSHIPS

Institutional Fellow of Open Association of Research Society (USA)-OARS (USA)

Global Journals Incorporation (USA) is accredited by Open Association of Research Society, U.S.A (OARS) and in turn, affiliates research institutions as “Institutional Fellow of Open Association of Research Society” (IFOARS).

The “FARSC” is a dignified title which is accorded to a person’s name viz. Dr. John E. Hall, Ph.D., FARSC or William Walldroff, M.S., FARSC.



The IFOARS institution is entitled to form a Board comprised of one Chairperson and three to five board members preferably from different streams. The Board will be recognized as “Institutional Board of Open Association of Research Society”-(IBOARS).

The Institute will be entitled to following benefits:



The IBOARS can initially review research papers of their institute and recommend them to publish with respective journal of Global Journals. It can also review the papers of other institutions after obtaining our consent. The second review will be done by peer reviewer of Global Journals Incorporation (USA). The Board is at liberty to appoint a peer reviewer with the approval of chairperson after consulting us.

The author fees of such paper may be waived off up to 40%.

The Global Journals Incorporation (USA) at its discretion can also refer double blind peer reviewed paper at their end to the board for the verification and to get recommendation for final stage of acceptance of publication.



The IBOARS can organize symposium/seminar/conference in their country on behalf of Global Journals Incorporation (USA)-OARS (USA). The terms and conditions can be discussed separately.

The Board can also play vital role by exploring and giving valuable suggestions regarding the Standards of “Open Association of Research Society, U.S.A (OARS)” so that proper amendment can take place for the benefit of entire research community. We shall provide details of particular standard only on receipt of request from the Board.



Journals Research
inducing researches

The board members can also join us as Individual Fellow with 40% discount on total fees applicable to Individual Fellow. They will be entitled to avail all the benefits as declared. Please visit Individual Fellow-sub menu of GlobalJournals.org to have more relevant details.

We shall provide you intimation regarding launching of e-version of journal of your stream time to time. This may be utilized in your library for the enrichment of knowledge of your students as well as it can also be helpful for the concerned faculty members.



After nomination of your institution as “Institutional Fellow” and constantly functioning successfully for one year, we can consider giving recognition to your institute to function as Regional/Zonal office on our behalf.

The board can also take up the additional allied activities for betterment after our consultation.

The following entitlements are applicable to individual Fellows:

Open Association of Research Society, U.S.A (OARS) By-laws states that an individual Fellow may use the designations as applicable, or the corresponding initials. The Credentials of individual Fellow and Associate designations signify that the individual has gained knowledge of the fundamental concepts. One is magnanimous and proficient in an expertise course covering the professional code of conduct, and follows recognized standards of practice.



Open Association of Research Society (US)/ Global Journals Incorporation (USA), as described in Corporate Statements, are educational, research publishing and professional membership organizations. Achieving our individual Fellow or Associate status is based mainly on meeting stated educational research requirements.

Disbursement of 40% Royalty earned through Global Journals : Researcher = 50%, Peer Reviewer = 37.50%, Institution = 12.50% E.g. Out of 40%, the 20% benefit should be passed on to researcher, 15 % benefit towards remuneration should be given to a reviewer and remaining 5% is to be retained by the institution.



We shall provide print version of 12 issues of any three journals [as per your requirement] out of our 38 journals worth \$ 2376 USD.

Other:

The individual Fellow and Associate designations accredited by Open Association of Research Society (US) credentials signify guarantees following achievements:

- The professional accredited with Fellow honor, is entitled to various benefits viz. name, fame, honor, regular flow of income, secured bright future, social status etc.



- In addition to above, if one is single author, then entitled to 40% discount on publishing research paper and can get 10% discount if one is co-author or main author among group of authors.
- The Fellow can organize symposium/seminar/conference on behalf of Global Journals Incorporation (USA) and he/she can also attend the same organized by other institutes on behalf of Global Journals.
- The Fellow can become member of Editorial Board Member after completing 3yrs.
- The Fellow can earn 60% of sales proceeds from the sale of reference/review books/literature/publishing of research paper.
- Fellow can also join as paid peer reviewer and earn 15% remuneration of author charges and can also get an opportunity to join as member of the Editorial Board of Global Journals Incorporation (USA)
- • This individual has learned the basic methods of applying those concepts and techniques to common challenging situations. This individual has further demonstrated an in-depth understanding of the application of suitable techniques to a particular area of research practice.

Note :

//

- In future, if the board feels the necessity to change any board member, the same can be done with the consent of the chairperson along with anyone board member without our approval.
- In case, the chairperson needs to be replaced then consent of 2/3rd board members are required and they are also required to jointly pass the resolution copy of which should be sent to us. In such case, it will be compulsory to obtain our approval before replacement.
- In case of “Difference of Opinion [if any]” among the Board members, our decision will be final and binding to everyone.

//

PROCESS OF SUBMISSION OF RESEARCH PAPER

The Area or field of specialization may or may not be of any category as mentioned in 'Scope of Journal' menu of the GlobalJournals.org website. There are 37 Research Journal categorized with Six parental Journals GJCST, GJMR, GJRE, GJMBR, GJSFR, GJHSS. For Authors should prefer the mentioned categories. There are three widely used systems UDC, DDC and LCC. The details are available as 'Knowledge Abstract' at Home page. The major advantage of this coding is that, the research work will be exposed to and shared with all over the world as we are being abstracted and indexed worldwide.

The paper should be in proper format. The format can be downloaded from first page of 'Author Guideline' Menu. The Author is expected to follow the general rules as mentioned in this menu. The paper should be written in MS-Word Format (*.DOC,*.DOCX).

The Author can submit the paper either online or offline. The authors should prefer online submission.Online Submission: There are three ways to submit your paper:

(A) (I) First, register yourself using top right corner of Home page then Login. If you are already registered, then login using your username and password.

(II) Choose corresponding Journal.

(III) Click 'Submit Manuscript'. Fill required information and Upload the paper.

(B) If you are using Internet Explorer, then Direct Submission through Homepage is also available.

(C) If these two are not convenient, and then email the paper directly to dean@globaljournals.org.

Offline Submission: Author can send the typed form of paper by Post. However, online submission should be preferred.



PREFERRED AUTHOR GUIDELINES

MANUSCRIPT STYLE INSTRUCTION (Must be strictly followed)

Page Size: 8.27" X 11"

- Left Margin: 0.65
- Right Margin: 0.65
- Top Margin: 0.75
- Bottom Margin: 0.75
- Font type of all text should be Swis 721 Lt BT.
- Paper Title should be of Font Size 24 with one Column section.
- Author Name in Font Size of 11 with one column as of Title.
- Abstract Font size of 9 Bold, "Abstract" word in Italic Bold.
- Main Text: Font size 10 with justified two columns section
- Two Column with Equal Column with of 3.38 and Gaping of .2
- First Character must be three lines Drop capped.
- Paragraph before Spacing of 1 pt and After of 0 pt.
- Line Spacing of 1 pt
- Large Images must be in One Column
- Numbering of First Main Headings (Heading 1) must be in Roman Letters, Capital Letter, and Font Size of 10.
- Numbering of Second Main Headings (Heading 2) must be in Alphabets, Italic, and Font Size of 10.

You can use your own standard format also.

Author Guidelines:

1. General,
2. Ethical Guidelines,
3. Submission of Manuscripts,
4. Manuscript's Category,
5. Structure and Format of Manuscript,
6. After Acceptance.

1. GENERAL

Before submitting your research paper, one is advised to go through the details as mentioned in following heads. It will be beneficial, while peer reviewer justify your paper for publication.

Scope

The Global Journals Inc. (US) welcome the submission of original paper, review paper, survey article relevant to the all the streams of Philosophy and knowledge. The Global Journals Inc. (US) is parental platform for Global Journal of Computer Science and Technology, Researches in Engineering, Medical Research, Science Frontier Research, Human Social Science, Management, and Business organization. The choice of specific field can be done otherwise as following in Abstracting and Indexing Page on this Website. As the all Global

Journals Inc. (US) are being abstracted and indexed (in process) by most of the reputed organizations. Topics of only narrow interest will not be accepted unless they have wider potential or consequences.

2. ETHICAL GUIDELINES

Authors should follow the ethical guidelines as mentioned below for publication of research paper and research activities.

Papers are accepted on strict understanding that the material in whole or in part has not been, nor is being, considered for publication elsewhere. If the paper once accepted by Global Journals Inc. (US) and Editorial Board, will become the copyright of the Global Journals Inc. (US).

Authorship: The authors and coauthors should have active contribution to conception design, analysis and interpretation of findings. They should critically review the contents and drafting of the paper. All should approve the final version of the paper before submission

The Global Journals Inc. (US) follows the definition of authorship set up by the Global Academy of Research and Development. According to the Global Academy of R&D authorship, criteria must be based on:

- 1) Substantial contributions to conception and acquisition of data, analysis and interpretation of the findings.
- 2) Drafting the paper and revising it critically regarding important academic content.
- 3) Final approval of the version of the paper to be published.

All authors should have been credited according to their appropriate contribution in research activity and preparing paper. Contributors who do not match the criteria as authors may be mentioned under Acknowledgement.

Acknowledgements: Contributors to the research other than authors credited should be mentioned under acknowledgement. The specifications of the source of funding for the research if appropriate can be included. Suppliers of resources may be mentioned along with address.

Appeal of Decision: The Editorial Board's decision on publication of the paper is final and cannot be appealed elsewhere.

Permissions: It is the author's responsibility to have prior permission if all or parts of earlier published illustrations are used in this paper.

Please mention proper reference and appropriate acknowledgements wherever expected.

If all or parts of previously published illustrations are used, permission must be taken from the copyright holder concerned. It is the author's responsibility to take these in writing.

Approval for reproduction/modification of any information (including figures and tables) published elsewhere must be obtained by the authors/copyright holders before submission of the manuscript. Contributors (Authors) are responsible for any copyright fee involved.

3. SUBMISSION OF MANUSCRIPTS

Manuscripts should be uploaded via this online submission page. The online submission is most efficient method for submission of papers, as it enables rapid distribution of manuscripts and consequently speeds up the review procedure. It also enables authors to know the status of their own manuscripts by emailing us. Complete instructions for submitting a paper is available below.

Manuscript submission is a systematic procedure and little preparation is required beyond having all parts of your manuscript in a given format and a computer with an Internet connection and a Web browser. Full help and instructions are provided on-screen. As an author, you will be prompted for login and manuscript details as Field of Paper and then to upload your manuscript file(s) according to the instructions.



To avoid postal delays, all transaction is preferred by e-mail. A finished manuscript submission is confirmed by e-mail immediately and your paper enters the editorial process with no postal delays. When a conclusion is made about the publication of your paper by our Editorial Board, revisions can be submitted online with the same procedure, with an occasion to view and respond to all comments.

Complete support for both authors and co-author is provided.

4. MANUSCRIPT'S CATEGORY

Based on potential and nature, the manuscript can be categorized under the following heads:

Original research paper: Such papers are reports of high-level significant original research work.

Review papers: These are concise, significant but helpful and decisive topics for young researchers.

Research articles: These are handled with small investigation and applications.

Research letters: The letters are small and concise comments on previously published matters.

5. STRUCTURE AND FORMAT OF MANUSCRIPT

The recommended size of original research paper is less than seven thousand words, review papers fewer than seven thousands words also. Preparation of research paper or how to write research paper, are major hurdle, while writing manuscript. The research articles and research letters should be fewer than three thousand words, the structure original research paper; sometime review paper should be as follows:

Papers: These are reports of significant research (typically less than 7000 words equivalent, including tables, figures, references), and comprise:

- (a) Title should be relevant and commensurate with the theme of the paper.
- (b) A brief Summary, "Abstract" (less than 150 words) containing the major results and conclusions.
- (c) Up to ten keywords, that precisely identifies the paper's subject, purpose, and focus.
- (d) An Introduction, giving necessary background excluding subheadings; objectives must be clearly declared.
- (e) Resources and techniques with sufficient complete experimental details (wherever possible by reference) to permit repetition; sources of information must be given and numerical methods must be specified by reference, unless non-standard.
- (f) Results should be presented concisely, by well-designed tables and/or figures; the same data may not be used in both; suitable statistical data should be given. All data must be obtained with attention to numerical detail in the planning stage. As reproduced design has been recognized to be important to experiments for a considerable time, the Editor has decided that any paper that appears not to have adequate numerical treatments of the data will be returned un-refereed;
- (g) Discussion should cover the implications and consequences, not just recapitulating the results; conclusions should be summarizing.
- (h) Brief Acknowledgements.
- (i) References in the proper form.

Authors should very cautiously consider the preparation of papers to ensure that they communicate efficiently. Papers are much more likely to be accepted, if they are cautiously designed and laid out, contain few or no errors, are summarizing, and be conventional to the approach and instructions. They will in addition, be published with much less delays than those that require much technical and editorial correction.



The Editorial Board reserves the right to make literary corrections and to make suggestions to improve brevity.

It is vital, that authors take care in submitting a manuscript that is written in simple language and adheres to published guidelines.

Format

Language: The language of publication is UK English. Authors, for whom English is a second language, must have their manuscript efficiently edited by an English-speaking person before submission to make sure that, the English is of high excellence. It is preferable, that manuscripts should be professionally edited.

Standard Usage, Abbreviations, and Units: Spelling and hyphenation should be conventional to The Concise Oxford English Dictionary. Statistics and measurements should at all times be given in figures, e.g. 16 min, except for when the number begins a sentence. When the number does not refer to a unit of measurement it should be spelt in full unless, it is 160 or greater.

Abbreviations supposed to be used carefully. The abbreviated name or expression is supposed to be cited in full at first usage, followed by the conventional abbreviation in parentheses.

Metric SI units are supposed to generally be used excluding where they conflict with current practice or are confusing. For illustration, 1.4 l rather than $1.4 \times 10^{-3} \text{ m}^3$, or 4 mm somewhat than $4 \times 10^{-3} \text{ m}$. Chemical formula and solutions must identify the form used, e.g. anhydrous or hydrated, and the concentration must be in clearly defined units. Common species names should be followed by underlines at the first mention. For following use the generic name should be constricted to a single letter, if it is clear.

Structure

All manuscripts submitted to Global Journals Inc. (US), ought to include:

Title: The title page must carry an instructive title that reflects the content, a running title (less than 45 characters together with spaces), names of the authors and co-authors, and the place(s) wherever the work was carried out. The full postal address in addition with the e-mail address of related author must be given. Up to eleven keywords or very brief phrases have to be given to help data retrieval, mining and indexing.

Abstract, used in Original Papers and Reviews:

Optimizing Abstract for Search Engines

Many researchers searching for information online will use search engines such as Google, Yahoo or similar. By optimizing your paper for search engines, you will amplify the chance of someone finding it. This in turn will make it more likely to be viewed and/or cited in a further work. Global Journals Inc. (US) have compiled these guidelines to facilitate you to maximize the web-friendliness of the most public part of your paper.

Key Words

A major linchpin in research work for the writing research paper is the keyword search, which one will employ to find both library and Internet resources.

One must be persistent and creative in using keywords. An effective keyword search requires a strategy and planning a list of possible keywords and phrases to try.

Search engines for most searches, use Boolean searching, which is somewhat different from Internet searches. The Boolean search uses "operators," words (and, or, not, and near) that enable you to expand or narrow your affords. Tips for research paper while preparing research paper are very helpful guideline of research paper.

Choice of key words is first tool of tips to write research paper. Research paper writing is an art. A few tips for deciding as strategically as possible about keyword search:



- One should start brainstorming lists of possible keywords before even begin searching. Think about the most important concepts related to research work. Ask, "What words would a source have to include to be truly valuable in research paper?" Then consider synonyms for the important words.
- It may take the discovery of only one relevant paper to let steer in the right keyword direction because in most databases, the keywords under which a research paper is abstracted are listed with the paper.
- One should avoid outdated words.

Keywords are the key that opens a door to research work sources. Keyword searching is an art in which researcher's skills are bound to improve with experience and time.

Numerical Methods: Numerical methods used should be clear and, where appropriate, supported by references.

Acknowledgements: Please make these as concise as possible.

References

References follow the Harvard scheme of referencing. References in the text should cite the authors' names followed by the time of their publication, unless there are three or more authors when simply the first author's name is quoted followed by et al. unpublished work has to only be cited where necessary, and only in the text. Copies of references in press in other journals have to be supplied with submitted typescripts. It is necessary that all citations and references be carefully checked before submission, as mistakes or omissions will cause delays.

References to information on the World Wide Web can be given, but only if the information is available without charge to readers on an official site. Wikipedia and Similar websites are not allowed where anyone can change the information. Authors will be asked to make available electronic copies of the cited information for inclusion on the Global Journals Inc. (US) homepage at the judgment of the Editorial Board.

The Editorial Board and Global Journals Inc. (US) recommend that, citation of online-published papers and other material should be done via a DOI (digital object identifier). If an author cites anything, which does not have a DOI, they run the risk of the cited material not being noticeable.

The Editorial Board and Global Journals Inc. (US) recommend the use of a tool such as Reference Manager for reference management and formatting.

Tables, Figures and Figure Legends

Tables: Tables should be few in number, cautiously designed, uncrowned, and include only essential data. Each must have an Arabic number, e.g. Table 4, a self-explanatory caption and be on a separate sheet. Vertical lines should not be used.

Figures: Figures are supposed to be submitted as separate files. Always take in a citation in the text for each figure using Arabic numbers, e.g. Fig. 4. Artwork must be submitted online in electronic form by e-mailing them.

Preparation of Electronic Figures for Publication

Even though low quality images are sufficient for review purposes, print publication requires high quality images to prevent the final product being blurred or fuzzy. Submit (or e-mail) EPS (line art) or TIFF (halftone/photographs) files only. MS PowerPoint and Word Graphics are unsuitable for printed pictures. Do not use pixel-oriented software. Scans (TIFF only) should have a resolution of at least 350 dpi (halftone) or 700 to 1100 dpi (line drawings) in relation to the imitation size. Please give the data for figures in black and white or submit a Color Work Agreement Form. EPS files must be saved with fonts embedded (and with a TIFF preview, if possible).

For scanned images, the scanning resolution (at final image size) ought to be as follows to ensure good reproduction: line art: >650 dpi; halftones (including gel photographs) : >350 dpi; figures containing both halftone and line images: >650 dpi.

Color Charges: It is the rule of the Global Journals Inc. (US) for authors to pay the full cost for the reproduction of their color artwork. Hence, please note that, if there is color artwork in your manuscript when it is accepted for publication, we would require you to complete and return a color work agreement form before your paper can be published.



Figure Legends: Self-explanatory legends of all figures should be incorporated separately under the heading 'Legends to Figures'. In the full-text online edition of the journal, figure legends may possibly be truncated in abbreviated links to the full screen version. Therefore, the first 100 characters of any legend should notify the reader, about the key aspects of the figure.

6. AFTER ACCEPTANCE

Upon approval of a paper for publication, the manuscript will be forwarded to the dean, who is responsible for the publication of the Global Journals Inc. (US).

6.1 Proof Corrections

The corresponding author will receive an e-mail alert containing a link to a website or will be attached. A working e-mail address must therefore be provided for the related author.

Acrobat Reader will be required in order to read this file. This software can be downloaded

(Free of charge) from the following website:

www.adobe.com/products/acrobat/readstep2.html. This will facilitate the file to be opened, read on screen, and printed out in order for any corrections to be added. Further instructions will be sent with the proof.

Proofs must be returned to the dean at dean@globaljournals.org within three days of receipt.

As changes to proofs are costly, we inquire that you only correct typesetting errors. All illustrations are retained by the publisher. Please note that the authors are responsible for all statements made in their work, including changes made by the copy editor.

6.2 Early View of Global Journals Inc. (US) (Publication Prior to Print)

The Global Journals Inc. (US) are enclosed by our publishing's Early View service. Early View articles are complete full-text articles sent in advance of their publication. Early View articles are absolute and final. They have been completely reviewed, revised and edited for publication, and the authors' final corrections have been incorporated. Because they are in final form, no changes can be made after sending them. The nature of Early View articles means that they do not yet have volume, issue or page numbers, so Early View articles cannot be cited in the conventional way.

6.3 Author Services

Online production tracking is available for your article through Author Services. Author Services enables authors to track their article - once it has been accepted - through the production process to publication online and in print. Authors can check the status of their articles online and choose to receive automated e-mails at key stages of production. The authors will receive an e-mail with a unique link that enables them to register and have their article automatically added to the system. Please ensure that a complete e-mail address is provided when submitting the manuscript.

6.4 Author Material Archive Policy

Please note that if not specifically requested, publisher will dispose off hardcopy & electronic information submitted, after the two months of publication. If you require the return of any information submitted, please inform the Editorial Board or dean as soon as possible.

6.5 Offprint and Extra Copies

A PDF offprint of the online-published article will be provided free of charge to the related author, and may be distributed according to the Publisher's terms and conditions. Additional paper offprint may be ordered by emailing us at: editor@globaljournals.org.

You must strictly follow above Author Guidelines before submitting your paper or else we will not at all be responsible for any corrections in future in any of the way.



Before start writing a good quality Computer Science Research Paper, let us first understand what is Computer Science Research Paper? So, Computer Science Research Paper is the paper which is written by professionals or scientists who are associated to Computer Science and Information Technology, or doing research study in these areas. If you are novel to this field then you can consult about this field from your supervisor or guide.

TECHNIQUES FOR WRITING A GOOD QUALITY RESEARCH PAPER:

1. Choosing the topic: In most cases, the topic is searched by the interest of author but it can be also suggested by the guides. You can have several topics and then you can judge that in which topic or subject you are finding yourself most comfortable. This can be done by asking several questions to yourself, like Will I be able to carry our search in this area? Will I find all necessary recourses to accomplish the search? Will I be able to find all information in this field area? If the answer of these types of questions will be "Yes" then you can choose that topic. In most of the cases, you may have to conduct the surveys and have to visit several places because this field is related to Computer Science and Information Technology. Also, you may have to do a lot of work to find all rise and falls regarding the various data of that subject. Sometimes, detailed information plays a vital role, instead of short information.

2. Evaluators are human: First thing to remember that evaluators are also human being. They are not only meant for rejecting a paper. They are here to evaluate your paper. So, present your Best.

3. Think Like Evaluators: If you are in a confusion or getting demotivated that your paper will be accepted by evaluators or not, then think and try to evaluate your paper like an Evaluator. Try to understand that what an evaluator wants in your research paper and automatically you will have your answer.

4. Make blueprints of paper: The outline is the plan or framework that will help you to arrange your thoughts. It will make your paper logical. But remember that all points of your outline must be related to the topic you have chosen.

5. Ask your Guides: If you are having any difficulty in your research, then do not hesitate to share your difficulty to your guide (if you have any). They will surely help you out and resolve your doubts. If you can't clarify what exactly you require for your work then ask the supervisor to help you with the alternative. He might also provide you the list of essential readings.

6. Use of computer is recommended: As you are doing research in the field of Computer Science, then this point is quite obvious.

7. Use right software: Always use good quality software packages. If you are not capable to judge good software then you can lose quality of your paper unknowingly. There are various software programs available to help you, which you can get through Internet.

8. Use the Internet for help: An excellent start for your paper can be by using the Google. It is an excellent search engine, where you can have your doubts resolved. You may also read some answers for the frequent question how to write my research paper or find model research paper. From the internet library you can download books. If you have all required books make important reading selecting and analyzing the specified information. Then put together research paper sketch out.

9. Use and get big pictures: Always use encyclopedias, Wikipedia to get pictures so that you can go into the depth.

10. Bookmarks are useful: When you read any book or magazine, you generally use bookmarks, right! It is a good habit, which helps to not to lose your continuity. You should always use bookmarks while searching on Internet also, which will make your search easier.

11. Revise what you wrote: When you write anything, always read it, summarize it and then finalize it.



12. Make all efforts: Make all efforts to mention what you are going to write in your paper. That means always have a good start. Try to mention everything in introduction, that what is the need of a particular research paper. Polish your work by good skill of writing and always give an evaluator, what he wants.

13. Have backups: When you are going to do any important thing like making research paper, you should always have backup copies of it either in your computer or in paper. This will help you to not to lose any of your important.

14. Produce good diagrams of your own: Always try to include good charts or diagrams in your paper to improve quality. Using several and unnecessary diagrams will degrade the quality of your paper by creating "hotchpotch." So always, try to make and include those diagrams, which are made by your own to improve readability and understandability of your paper.

15. Use of direct quotes: When you do research relevant to literature, history or current affairs then use of quotes become essential but if study is relevant to science then use of quotes is not preferable.

16. Use proper verb tense: Use proper verb tenses in your paper. Use past tense, to present those events that happened. Use present tense to indicate events that are going on. Use future tense to indicate future happening events. Use of improper and wrong tenses will confuse the evaluator. Avoid the sentences that are incomplete.

17. Never use online paper: If you are getting any paper on Internet, then never use it as your research paper because it might be possible that evaluator has already seen it or maybe it is outdated version.

18. Pick a good study spot: To do your research studies always try to pick a spot, which is quiet. Every spot is not for studies. Spot that suits you choose it and proceed further.

19. Know what you know: Always try to know, what you know by making objectives. Else, you will be confused and cannot achieve your target.

20. Use good quality grammar: Always use a good quality grammar and use words that will throw positive impact on evaluator. Use of good quality grammar does not mean to use tough words, that for each word the evaluator has to go through dictionary. Do not start sentence with a conjunction. Do not fragment sentences. Eliminate one-word sentences. Ignore passive voice. Do not ever use a big word when a diminutive one would suffice. Verbs have to be in agreement with their subjects. Prepositions are not expressions to finish sentences with. It is incorrect to ever divide an infinitive. Avoid clichés like the disease. Also, always shun irritating alliteration. Use language that is simple and straight forward. put together a neat summary.

21. Arrangement of information: Each section of the main body should start with an opening sentence and there should be a changeover at the end of the section. Give only valid and powerful arguments to your topic. You may also maintain your arguments with records.

22. Never start in last minute: Always start at right time and give enough time to research work. Leaving everything to the last minute will degrade your paper and spoil your work.

23. Multitasking in research is not good: Doing several things at the same time proves bad habit in case of research activity. Research is an area, where everything has a particular time slot. Divide your research work in parts and do particular part in particular time slot.

24. Never copy others' work: Never copy others' work and give it your name because if evaluator has seen it anywhere you will be in trouble.

25. Take proper rest and food: No matter how many hours you spend for your research activity, if you are not taking care of your health then all your efforts will be in vain. For a quality research, study is must, and this can be done by taking proper rest and food.

26. Go for seminars: Attend seminars if the topic is relevant to your research area. Utilize all your resources.



27. Refresh your mind after intervals: Try to give rest to your mind by listening to soft music or by sleeping in intervals. This will also improve your memory.

28. Make colleagues: Always try to make colleagues. No matter how sharper or intelligent you are, if you make colleagues you can have several ideas, which will be helpful for your research.

29. Think technically: Always think technically. If anything happens, then search its reasons, its benefits, and demerits.

30. Think and then print: When you will go to print your paper, notice that tables are not be split, headings are not detached from their descriptions, and page sequence is maintained.

31. Adding unnecessary information: Do not add unnecessary information, like, I have used MS Excel to draw graph. Do not add irrelevant and inappropriate material. These all will create superfluous. Foreign terminology and phrases are not apropos. One should NEVER take a broad view. Analogy in script is like feathers on a snake. Not at all use a large word when a very small one would be sufficient. Use words properly, regardless of how others use them. Remove quotations. Puns are for kids, not grunt readers. Amplification is a billion times of inferior quality than sarcasm.

32. Never oversimplify everything: To add material in your research paper, never go for oversimplification. This will definitely irritate the evaluator. Be more or less specific. Also too, by no means, ever use rhythmic redundancies. Contractions aren't essential and shouldn't be there used. Comparisons are as terrible as clichés. Give up ampersands and abbreviations, and so on. Remove commas, that are, not necessary. Parenthetical words however should be together with this in commas. Understatement is all the time the complete best way to put onward earth-shaking thoughts. Give a detailed literary review.

33. Report concluded results: Use concluded results. From raw data, filter the results and then conclude your studies based on measurements and observations taken. Significant figures and appropriate number of decimal places should be used. Parenthetical remarks are prohibitive. Proofread carefully at final stage. In the end give outline to your arguments. Spot out perspectives of further study of this subject. Justify your conclusion by at the bottom of them with sufficient justifications and examples.

34. After conclusion: Once you have concluded your research, the next most important step is to present your findings. Presentation is extremely important as it is the definite medium through which your research is going to be in print to the rest of the crowd. Care should be taken to categorize your thoughts well and present them in a logical and neat manner. A good quality research paper format is essential because it serves to highlight your research paper and bring to light all necessary aspects in your research.

INFORMAL GUIDELINES OF RESEARCH PAPER WRITING

Key points to remember:

- Submit all work in its final form.
- Write your paper in the form, which is presented in the guidelines using the template.
- Please note the criterion for grading the final paper by peer-reviewers.

Final Points:

A purpose of organizing a research paper is to let people to interpret your effort selectively. The journal requires the following sections, submitted in the order listed, each section to start on a new page.

The introduction will be compiled from reference matter and will reflect the design processes or outline of basis that direct you to make study. As you will carry out the process of study, the method and process section will be constructed as like that. The result segment will show related statistics in nearly sequential order and will direct the reviewers next to the similar intellectual paths throughout the data that you took to carry out your study. The discussion section will provide understanding of the data and projections as to the implication of the results. The use of good quality references all through the paper will give the effort trustworthiness by representing an alertness of prior workings.



Writing a research paper is not an easy job no matter how trouble-free the actual research or concept. Practice, excellent preparation, and controlled record keeping are the only means to make straightforward the progression.

General style:

Specific editorial column necessities for compliance of a manuscript will always take over from directions in these general guidelines.

To make a paper clear

- Adhere to recommended page limits

Mistakes to evade

- Insertion a title at the foot of a page with the subsequent text on the next page
- Separating a table/chart or figure - impound each figure/table to a single page
- Submitting a manuscript with pages out of sequence

In every sections of your document

- Use standard writing style including articles ("a", "the," etc.)
- Keep on paying attention on the research topic of the paper
- Use paragraphs to split each significant point (excluding for the abstract)
- Align the primary line of each section
- Present your points in sound order
- Use present tense to report well accepted
- Use past tense to describe specific results
- Shun familiar wording, don't address the reviewer directly, and don't use slang, slang language, or superlatives
- Shun use of extra pictures - include only those figures essential to presenting results

Title Page:

Choose a revealing title. It should be short. It should not have non-standard acronyms or abbreviations. It should not exceed two printed lines. It should include the name(s) and address (es) of all authors.



Abstract:

The summary should be two hundred words or less. It should briefly and clearly explain the key findings reported in the manuscript-- must have precise statistics. It should not have abnormal acronyms or abbreviations. It should be logical in itself. Shun citing references at this point.

An abstract is a brief distinct paragraph summary of finished work or work in development. In a minute or less a reviewer can be taught the foundation behind the study, common approach to the problem, relevant results, and significant conclusions or new questions.

Write your summary when your paper is completed because how can you write the summary of anything which is not yet written? Wealth of terminology is very essential in abstract. Yet, use comprehensive sentences and do not let go readability for briefness. You can maintain it succinct by phrasing sentences so that they provide more than lone rationale. The author can at this moment go straight to shortening the outcome. Sum up the study, with the subsequent elements in any summary. Try to maintain the initial two items to no more than one ruling each.

- Reason of the study - theory, overall issue, purpose
- Fundamental goal
- To the point depiction of the research
- Consequences, including definite statistics - if the consequences are quantitative in nature, account quantitative data; results of any numerical analysis should be reported
- Significant conclusions or questions that track from the research(es)

Approach:

- Single section, and succinct
- As an outline of job done, it is always written in past tense
- A conceptual should situate on its own, and not submit to any other part of the paper such as a form or table
- Center on shortening results - bound background information to a verdict or two, if completely necessary
- What you account in an conceptual must be regular with what you reported in the manuscript
- Exact spelling, clearness of sentences and phrases, and appropriate reporting of quantities (proper units, important statistics) are just as significant in an abstract as they are anywhere else

Introduction:

The **Introduction** should "introduce" the manuscript. The reviewer should be presented with sufficient background information to be capable to comprehend and calculate the purpose of your study without having to submit to other works. The basis for the study should be offered. Give most important references but shun difficult to make a comprehensive appraisal of the topic. In the introduction, describe the problem visibly. If the problem is not acknowledged in a logical, reasonable way, the reviewer will have no attention in your result. Speak in common terms about techniques used to explain the problem, if needed, but do not present any particulars about the protocols here. Following approach can create a valuable beginning:

- Explain the value (significance) of the study
- Shield the model - why did you employ this particular system or method? What is its compensation? You strength remark on its appropriateness from an abstract point of vision as well as point out sensible reasons for using it.
- Present a justification. Status your particular theory (es) or aim(s), and describe the logic that led you to choose them.
- Very for a short time explain the tentative propose and how it skilled the declared objectives.

Approach:

- Use past tense except for when referring to recognized facts. After all, the manuscript will be submitted after the entire job is done.
- Sort out your thoughts; manufacture one key point with every section. If you make the four points listed above, you will need a least of four paragraphs.



- Present surroundings information only as desirable in order hold up a situation. The reviewer does not desire to read the whole thing you know about a topic.
- Shape the theory/purpose specifically - do not take a broad view.
- As always, give awareness to spelling, simplicity and correctness of sentences and phrases.

Procedures (Methods and Materials):

This part is supposed to be the easiest to carve if you have good skills. A sound written Procedures segment allows a capable scientist to replacement your results. Present precise information about your supplies. The suppliers and clarity of reagents can be helpful bits of information. Present methods in sequential order but linked methodologies can be grouped as a segment. Be concise when relating the protocols. Attempt for the least amount of information that would permit another capable scientist to spare your outcome but be cautious that vital information is integrated. The use of subheadings is suggested and ought to be synchronized with the results section. When a technique is used that has been well described in another object, mention the specific item describing a way but draw the basic principle while stating the situation. The purpose is to text all particular resources and broad procedures, so that another person may use some or all of the methods in one more study or referee the scientific value of your work. It is not to be a step by step report of the whole thing you did, nor is a methods section a set of orders.

Materials:

- Explain materials individually only if the study is so complex that it saves liberty this way.
- Embrace particular materials, and any tools or provisions that are not frequently found in laboratories.
- Do not take in frequently found.
- If use of a definite type of tools.
- Materials may be reported in a part section or else they may be recognized along with your measures.

Methods:

- Report the method (not particulars of each process that engaged the same methodology)
- Describe the method entirely
- To be succinct, present methods under headings dedicated to specific dealings or groups of measures
- Simplify - details how procedures were completed not how they were exclusively performed on a particular day.
- If well known procedures were used, account the procedure by name, possibly with reference, and that's all.

Approach:

- It is embarrassed or not possible to use vigorous voice when documenting methods with no using first person, which would focus the reviewer's interest on the researcher rather than the job. As a result when script up the methods most authors use third person passive voice.
- Use standard style in this and in every other part of the paper - avoid familiar lists, and use full sentences.

What to keep away from

- Resources and methods are not a set of information.
- Skip all descriptive information and surroundings - save it for the argument.
- Leave out information that is immaterial to a third party.

Results:

The principle of a results segment is to present and demonstrate your conclusion. Create this part a entirely objective details of the outcome, and save all understanding for the discussion.

The page length of this segment is set by the sum and types of data to be reported. Carry on to be to the point, by means of statistics and tables, if suitable, to present consequences most efficiently. You must obviously differentiate material that would usually be incorporated in a study editorial from any unprocessed data or additional appendix matter that would not be available. In fact, such matter should not be submitted at all except requested by the instructor.



Content

- Sum up your conclusion in text and demonstrate them, if suitable, with figures and tables.
- In manuscript, explain each of your consequences, point the reader to remarks that are most appropriate.
- Present a background, such as by describing the question that was addressed by creation an exacting study.
- Explain results of control experiments and comprise remarks that are not accessible in a prescribed figure or table, if appropriate.
- Examine your data, then prepare the analyzed (transformed) data in the form of a figure (graph), table, or in manuscript form.

What to stay away from

- Do not discuss or infer your outcome, report surroundings information, or try to explain anything.
- Not at all, take in raw data or intermediate calculations in a research manuscript.
- Do not present the similar data more than once.
- Manuscript should complement any figures or tables, not duplicate the identical information.
- Never confuse figures with tables - there is a difference.

Approach

- As forever, use past tense when you submit to your results, and put the whole thing in a reasonable order.
- Put figures and tables, appropriately numbered, in order at the end of the report
- If you desire, you may place your figures and tables properly within the text of your results part.

Figures and tables

- If you put figures and tables at the end of the details, make certain that they are visibly distinguished from any attach appendix materials, such as raw facts
- Despite of position, each figure must be numbered one after the other and complete with subtitle
- In spite of position, each table must be titled, numbered one after the other and complete with heading
- All figure and table must be adequately complete that it could situate on its own, divide from text

Discussion:

The Discussion is expected the trickiest segment to write and describe. A lot of papers submitted for journal are discarded based on problems with the Discussion. There is no head of state for how long a argument should be. Position your understanding of the outcome visibly to lead the reviewer through your conclusions, and then finish the paper with a summing up of the implication of the study. The purpose here is to offer an understanding of your results and hold up for all of your conclusions, using facts from your research and generally accepted information, if suitable. The implication of result should be visibly described. Infer your data in the conversation in suitable depth. This means that when you clarify an observable fact you must explain mechanisms that may account for the observation. If your results vary from your prospect, make clear why that may have happened. If your results agree, then explain the theory that the proof supported. It is never suitable to just state that the data approved with prospect, and let it drop at that.

- Make a decision if each premise is supported, discarded, or if you cannot make a conclusion with assurance. Do not just dismiss a study or part of a study as "uncertain."
- Research papers are not acknowledged if the work is imperfect. Draw what conclusions you can based upon the results that you have, and take care of the study as a finished work
- You may propose future guidelines, such as how the experiment might be personalized to accomplish a new idea.
- Give details all of your remarks as much as possible, focus on mechanisms.
- Make a decision if the tentative design sufficiently addressed the theory, and whether or not it was correctly restricted.
- Try to present substitute explanations if sensible alternatives be present.
- One research will not counter an overall question, so maintain the large picture in mind, where do you go next? The best studies unlock new avenues of study. What questions remain?
- Recommendations for detailed papers will offer supplementary suggestions.

Approach:

- When you refer to information, differentiate data generated by your own studies from available information
- Submit to work done by specific persons (including you) in past tense.
- Submit to generally acknowledged facts and main beliefs in present tense.



THE ADMINISTRATION RULES

Please carefully note down following rules and regulation before submitting your Research Paper to Global Journals Inc. (US):

Segment Draft and Final Research Paper: You have to strictly follow the template of research paper. If it is not done your paper may get rejected.

- The **major constraint** is that you must independently make all content, tables, graphs, and facts that are offered in the paper. You must write each part of the paper wholly on your own. The Peer-reviewers need to identify your own perceptive of the concepts in your own terms. NEVER extract straight from any foundation, and never rephrase someone else's analysis.
- Do not give permission to anyone else to "PROOFREAD" your manuscript.
- **Methods to avoid Plagiarism is applied by us on every paper, if found guilty, you will be blacklisted by all of our collaborated research groups, your institution will be informed for this and strict legal actions will be taken immediately.)**
- To guard yourself and others from possible illegal use please do not permit anyone right to use to your paper and files.



CRITERION FOR GRADING A RESEARCH PAPER (COMPILATION)
BY GLOBAL JOURNALS INC. (US)

Please note that following table is only a Grading of "Paper Compilation" and not on "Performed/Stated Research" whose grading solely depends on Individual Assigned Peer Reviewer and Editorial Board Member. These can be available only on request and after decision of Paper. This report will be the property of Global Journals Inc. (US).

Topics	Grades		
	A-B	C-D	E-F
Abstract	Clear and concise with appropriate content, Correct format. 200 words or below	Unclear summary and no specific data, Incorrect form Above 200 words	No specific data with ambiguous information Above 250 words
Introduction	Containing all background details with clear goal and appropriate details, flow specification, no grammar and spelling mistake, well organized sentence and paragraph, reference cited	Unclear and confusing data, appropriate format, grammar and spelling errors with unorganized matter	Out of place depth and content, hazy format
Methods and Procedures	Clear and to the point with well arranged paragraph, precision and accuracy of facts and figures, well organized subheads	Difficult to comprehend with embarrassed text, too much explanation but completed	Incorrect and unorganized structure with hazy meaning
Result	Well organized, Clear and specific, Correct units with precision, correct data, well structuring of paragraph, no grammar and spelling mistake	Complete and embarrassed text, difficult to comprehend	Irregular format with wrong facts and figures
Discussion	Well organized, meaningful specification, sound conclusion, logical and concise explanation, highly structured paragraph reference cited	Wordy, unclear conclusion, spurious	Conclusion is not cited, unorganized, difficult to comprehend
References	Complete and correct format, well organized	Beside the point, Incomplete	Wrong format and structuring



INDEX

A

Aladeloba · 40

G

Gaisler · 59, 64

J

Jeonghwan · 68

L

Levelin · 30

P

Prefetch · 20

R

Rotenberg · 40

T

Tomasulo · 5, 24, 42

V

Verilog · 9, 13, 15, 40, 59, 61, 64

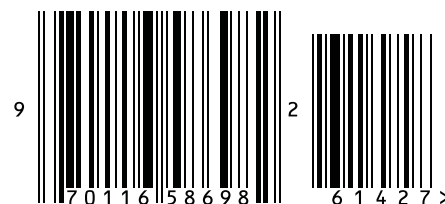


save our planet



Global Journal of Computer Science and Technology

Visit us on the Web at www.GlobalJournals.org | www.ComputerResearch.org
or email us at helpdesk@globaljournals.org



ISSN 9754350

© Global Journals Inc.