A black background with white text

Description automatically generated

Development Phase Report - JoSDC’24

|  |  |  |
| --- | --- | --- |
| Team Name | | |
| The Plumbers | | |
| Team Members | | |
| # | Name | Email |
| 1 | Saleh Ibrahim Al-ghobari | salehghobari@outlook.com |
| 2 | Ahmad Khalil Fratekh | ahmadkhalil.af@gmail.com |
| 3 | Omar Kasasbeh | omar.kasasbeh.02@gmail.com |
| 4 |  |  |
| 5 |  |  |

|  |
| --- |
| A group of men with tools and computer  Description automatically generated |

Jordan National Semiconductor Design Competition (JOSDC’2024)

Project Title

By

Ahmad Fratekh

Computer Engineering

[Ahmadkhalil.af@gmail.com](mailto:Ahmadkhalil.af@gmail.com)

Saleh Ghobari

Computer Engineering

salehghobari@outlook.com

Omar Kasasbeh

Electrical Engineering

Omart.kasasbeh.02@gmail.com

Amman, Jordan

12 / 2024

Acknowledgments

Recognition or favorable notice for people.

Abstract

In this project we present the design and implementation of a **MIPS processor**, showcasing two distinct architectures: a single-cycle design and a 5-stage pipelined design.

The single-cycle design executes each instruction in one clock cycle, offering simplicity in design and control logic but at the cost of longer clock cycles due to the need to accommodate the slowest instruction. In contrast, the 5-stage pipelined design splits instruction execution into five stages: fetch, decode, execute, memory access, and write-back. This approach improves performance by enabling multiple instructions to be processed simultaneously, effectively increasing throughput. However, it introduces complexities such as data hazards, control hazards, and the need for forwarding, stalling, and branch prediction mechanisms.

The project includes detailed explanations of both architectures, their control paths, and data paths, as well as the implementation challenges faced during the design process. A comparative analysis highlights the trade-offs between simplicity and performance, illustrating how the pipelined design achieves higher instruction throughput at the expense of increased design complexity.

Simulation results demonstrate the functionality and efficiency of both designs, providing insights into their suitability for various computational workloads. This work serves as a foundational exploration of MIPS processor architecture, contributing to understanding modern processor design principles.

**Table of Contents**

[1 Executive Summary 8](#_Toc180434632)

[2 Introduction 9](#_Toc180434633)

[2.1 Objectives 9](#_Toc180434634)

[2.2 Design Achieved 9](#_Toc180434635)

[3 MIPS Reference Architecture 10](#_Toc180434636)

[3.1 Instruction Set 10](#_Toc180434637)

[3.2 Memory Model 10](#_Toc180434638)

[3.3 Machine Code 10](#_Toc180434639)

[4 Design 11](#_Toc180434640)

[4.1 Hardware Design and Implementation 11](#_Toc180434641)

[4.2 Coding and Software Development 11](#_Toc180434642)

[5 Results 12](#_Toc180434643)

[5.1 Experiment/Simulation Results Discussion: 12](#_Toc180434644)

[5.2 Prototype Setup 12](#_Toc180434645)

[5.3 Validation of requirements 12](#_Toc180434646)

[6 Conclusion 13](#_Toc180434647)

**List of Figures**

Figure 1.1 Figure one in chapter one ..……………………………….……………..page number

Figure 2.1 Figure one in chapter two ..……………………………….……………..page number

Figure 2.2 Figure two in chapter two ..……………………………….……………..page number

**List of Tables**

Table 1.1 Table one in chapter one ..……………………………….……………..page number

Table 2.1 Table one in chapter two ..……………………………….……………..page number

Table 2.2 Table two in chapter two ..……………………………….……………..page number

# Executive Summary

This report presents the design and implementation of a MIPS CPU, comparing two distinct architectures: a single-cycle design and a 5-stage pipelined design. The project aims to explore the fundamental principles of processor architecture, demonstrating the trade-offs between simplicity and performance. Both designs were implemented, simulated, and analyzed to highlight their technical and operational differences, focusing on key performance metrics.

## **Project Overview**

The MIPS CPU design adheres to the RISC (Reduced Instruction Set Computer) principles, emphasizing simplicity, efficiency, and scalability. The single-cycle design processes each instruction in a single clock cycle, making it straightforward and easier to implement. However, its clock cycle time is constrained by the slowest instruction, limiting its overall performance. The 5-stage pipelined design, on the other hand, breaks down instruction execution into discrete stages—instruction fetch, decode, execute, memory access, and write-back—allowing multiple instructions to be processed concurrently.

## **Design Summaries**

### Single-Cycle Design

* **Architecture:** Implements a simple control unit and a single data path.
* **Clock Cycle**: Long clock cycle to accommodate the slowest instruction.
* **Advantages**: Easy to design and debug.
* **Challenges**: Limited performance due to the lack of instruction overlap.

### 5-Stage Pipelined Design

* **Architecture**: Introduces pipeline registers to separate the five stages.
* **Clock Cycle**: Shorter clock cycle, enabling higher clock frequencies.
* **Advantages**: Improved throughput with multiple instructions in execution simultaneously.
* **Challenges**: Requires handling hazards (data, control, and structural).
* **Data Hazards**: Mitigated using forwarding and stalling mechanisms.
* **Control Hazards**: Addressed using a 2-bit branch prediction unit and program counter correction.

## **Key Results and Technical Achievements**

### Performance Improvements:

* The pipelined design achieved a significant increase in instruction throughput compared to the single-cycle design. While the single-cycle CPU could execute one instruction per clock cycle, the pipelined CPU executed up to five instructions simultaneously under ideal conditions.
* Simulations indicated an average speed up of approximately 2x for workloads with minimal control hazards.

### Technical Achievements:

* **Functional Verification**: Both designs were verified using a comprehensive suite of test programs to ensure correctness (i.e. the benchmarks provided by the committee).
* **Hazard Mitigation**: Effective implementation of data forwarding and stalling ensured the correctness of the pipelined design.
* **Branch Prediction**: The inclusion of a 2-bit dynamic branch prediction unit significantly reduced control hazards, further enhancing pipelined performance.

### Resource Utilization:

* The single-cycle design required fewer hardware resources but had limited scalability.
* The pipelined design utilized additional hardware, including pipeline registers and hazard control logic, but demonstrated better resource efficiency for performance-critical applications.

## **Conclusion**

This project successfully demonstrated the implementation and comparative analysis of single-cycle and pipelined MIPS CPU designs. The single-cycle design is suitable for simplicity-focused applications or educational purposes, while the pipelined design excels in performance-intensive scenarios. The findings highlight the trade-offs inherent in CPU architecture, offering valuable insights into the design choices faced by modern processor engineers. Additionally, the project underscored the importance of hazard mitigation and branch prediction in achieving optimal pipeline performance. The integration of a 2-bit branch prediction unit marked a significant technical achievement, enhancing the pipelined architecture's reliability and efficiency. Overall, this work contributes to a deeper understanding of processor design and lays a foundation for further advancements in modern CPU architecture.

## **Recommendations**

### For further development, the following improvements are recommended:

* Explore more advanced dynamic branch prediction techniques to further reduce control hazards.
* Implement out-of-order execution in the pipelined design for greater efficiency.
* Investigate multi-core architectures to extend the scalability of the MIPS CPU.

.

# Introduction

Describe the overall components and design methodology.

## Objectives

State what the project and the report aim to achieve.

## Design Achieved

Highlight the key outcomes of the design phase, focusing on the final solution or system you've built.

# MIPS Reference Architecture

This section provides an overview of the MIPS architecture, focusing on the essential components relevant to your project. It includes the instruction set, memory model, and machine code, which are crucial for understanding how MIPS supports your design.

## Instruction Set

* Explain what the instruction set is (a group of instructions that the processor can execute).
* Explain the basic instruction formats (R-format, I-format, J-format) and their components (opcode, source registers, destination registers, etc.).

## Memory Model

* Explain how memory is organized and managed in the MIPS architecture.

## Machine Code

* Explain MIPS machine code, showing how high-level instructions are translated into binary code that the processor executes.

# Design

## Hardware Design and Implementation

* Describe the hardware design and component used.
* Explain the physical setup of the hardware.
* Discuss any challenges faced during hardware implementation.
* Include photographs, diagrams, or schematics of the hardware setup.
* **This section should include:**
  + **Single Cycle Processor and Pipelined Processor (explaining the datapath and the control unit for both).**

## Coding and Software Development

* Describe the coding and software development aspects of your project.
* Explain the programming languages and tools used.
* Provide code flow charts or algorithms related to your project.
* Discuss the coding challenges and solutions.

# Evaluation & Results

• Present the results of testing and validation procedures for both simulation and hardware.

• Include data, graphs, and tables to support your findings.

• Discuss the performance and functionality of the integrated system.

• Use a table to summarize that requirements were met.

• **This section should include:**

* **Benchmarks (including single cycle processor, pipelined processor, and the final design)**
* **Coverage**
* **Performance (Speedups)**
* **Summary: rubrics tables**

## Experiment/Simulation Results Discussion:

* + Use
    - Tables
    - Graphs
    - Waveform
    - figures

## Prototype Setup

* + Hardware
  + Software

## Validation of requirements

* + Discuss and analyze whether the requirements are met

# Conclusion

* Summarize the key findings and their implications.
* Assess whether the project's objectives were achieved.
* Provide recommendations for future work.

References

Follow a format consistent with IEEE guidelines and utilize conference and journal papers for your reference list

APPENDICES

These are detailed documentation of points mentioned in the report (e.g. technical data, questionnaires, chart …. etc.) which are considered supplementary information but too long or not quite relevant enough to include in the main body of the report.

Appendices may be labeled with letters as Appendix A, Appendix B, and so on.

Example,

Appendix A: VERILOG CODE

Appendix B: MACHINE CODE GENERATORS

Appendix C: COVERAGE TOOLS

Appendix D: VGA