MULTIMEDIA UNIVERSITY

TRIMESTER 1, SESSION 2023/2024

FYP Research Proposal

Design a 32-bit MIPS ALU based on RISC

EPE4036 – PROJECT

Name	Student ID	Faculty
Saeb K. H. Naser	1191302764	FOE

Supervisor: Dr. Chinnaiyan Senthilpari

TABLE OF CONTENTS

INTRODUCTION

In the ever-evolving realm of computer engineering, the design of processors plays a pivotal role in shaping the performance and efficiency of computing systems. This research seeks to make a significant contribution to this dynamic field by meticulously designing and implementing a 32-bit Arithmetic Logic Unit (ALU) grounded in the principles of Reduced Instruction Set Computing (RISC) within the Microprocessor without Interlocked Pipelined Stages (MIPS) architecture.

As a fundamental component responsible for executing arithmetic and logic operations, the ALU's significance in contemporary computing cannot be overstated. This project focuses on the intricacies of ALU design within the esteemed 32-bit MIPS architecture, renowned for its elegance and simplicity, and a longstanding contributor to efficient processor design.

The MIPS architecture's commitment to RISC principles has made it a benchmark in processor development, inspiring subsequent architectures to optimize instruction execution. The dedication to a reduced set of instructions, each executed in a single clock cycle, aligns seamlessly with the industry's pursuit of streamlined and high-performance processor architectures.

Amidst technological advancements and the emergence of new computing paradigms, the demand for processors capable of handling intricate operations efficiently becomes more pronounced. This research project addresses this need by concentrating on the ALU's design intricacies, seeking to enhance its capabilities within the proven framework of the MIPS architecture.

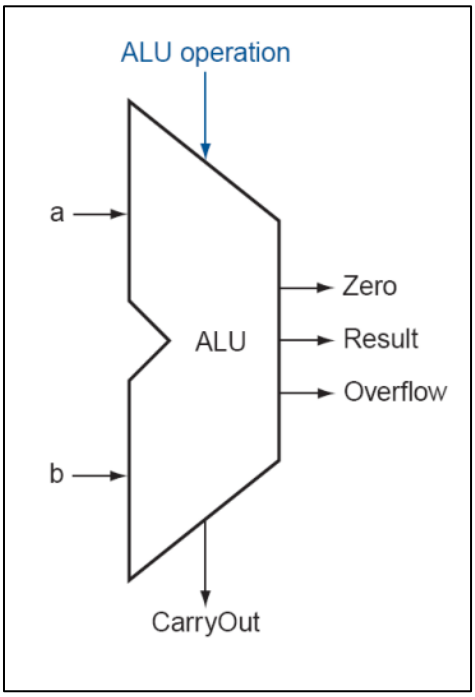

Figure 1: A symbolic representation of an ALU

AIM OF THE PROJECT

The primary aim of this research is to design and implement a 32-bit MIPS ALU that not only aligns with the principles of the MIPS architecture but also embodies the efficiency and simplicity inherent in the RISC ISA. Focusing specifically on the ALU, the core of arithmetic and logic operations, this project contributes to the ongoing discourse on processor optimization. Through a meticulous exploration of the problem statement, literature review, design methodology, and expected outcomes, the goal is to offer a comprehensive understanding of the rationale, methodology, and potential impact of designing a 32-bit MIPS ALU based on RISC ISA.

PROBLEM STATEMENT

In the contemporary landscape of computer engineering, the demand for processors capable of handling increasingly complex operations has intensified. While advancements in technology have paved the way for enhanced computational capabilities, challenges persist in optimizing the efficiency of processor architectures, particularly in executing intricate arithmetic and logic operations.

The prevailing design paradigms face constraints in achieving a harmonious balance between speed, complexity, and energy efficiency. As computing tasks become more sophisticated, the need for processors that can execute operations swiftly without compromising power efficiency becomes paramount. Traditional processor architectures, although robust, may struggle to meet these evolving demands.

The heart of any processor, the Arithmetic Logic Unit (ALU), is central to the execution of fundamental operations that underpin computing tasks. The intricacies of ALU design, particularly within the context of established architectures like the 32-bit MIPS, present a nuanced challenge. While MIPS architecture is renowned for its elegance and efficiency, there exists an opportunity to further enhance the ALU's capabilities to meet the demands of modern computing.

This research project identifies the gap in contemporary processor architectures, particularly in the design nuances of the ALU, and seeks to address this gap through a focused exploration within the 32-bit MIPS architecture. The specific challenges include optimizing the ALU's performance to align seamlessly with the principles of Reduced Instruction Set Computing (RISC) while accommodating the intricacies of modern computational tasks.

In light of these considerations, the problem addressed by this research is the need for a 32-bit MIPS ALU that not only adheres to the established principles of the MIPS architecture but also surpasses current design limitations. This includes optimizing the ALU's efficiency, minimizing power consumption, and enhancing its capacity to handle increasingly intricate arithmetic and logic operations. By identifying and addressing these challenges, this research aims to contribute to the ongoing discourse on processor optimization and efficient ALU design.

LITERATURE REVIEW

"Synthesis and Simulation of a 32Bit MIPS RISC Processor using VHDL"

The paper titled "Synthesis and Simulation of a 32Bit MIPS RISC Processor using VHDL" [1] by Mr. S. P. Ritpurkar, Prof. M. N. Thakare, and Prof. G. D. Korde explores the design and simulation of a 32-bit MIPS RISC Processor using Very High-Speed Integrated Circuit Hardware Description Language (VHDL). The primary objective is to analyze the components of the processor, including the Instruction fetch module, Decoder module, and Execution module, which encompasses a 32-bit Floating-point ALU, Flag register, MIPS Instruction Set, and 32-bit general-purpose registers. The design theory is based on the 32-bit MIPS RISC Processor, incorporating a pipeline concept for improved efficiency.

The paper begins by distinguishing between Complex Instruction Set Computer (CISC) and Reduced Instruction Set Computer (RISC) architectures. It provides insights into the characteristics of CISC, exemplifying processors like System 360–IBM, VAX–Digital Equipment Corporation, and Intel x86 architecture. The focus then shifts to RISC, emphasizing its aim for simplicity and speed, with examples such as Apple iPods, iPhones, and MIPS Technologies' prevalence in home electronics.

The MIPS Instruction Set Architecture is introduced, highlighting four formats: R-Type, I-Type, J-Type, and I/O-Type. Each format is explained, depicting their structure and purpose. The paper underscores the advantages of having a smaller number of instructions, uniform instruction sizes, and an organized register file, contributing to the ease of design and cost-effectiveness.

The authors delve into the pipeline design, explaining the five stages: Instruction Fetch (IF), Instruction Decoder (ID), Execution (EXE), Memory and I/O (MEM), and Write-Back (WB). Each stage's role is detailed, emphasizing the decomposition of the pipeline to enhance instruction throughput.

The experimental section presents the Register Transfer Logic (RTL) view of the 32-bit MIPS RISC Processor, showcasing units such as Instruction Fetch, Instruction Decoder, Execution, and Memory. Simulation results for different instruction formats, including R-Type, I-Type, J-Type, and I/O-Type, are provided, demonstrating the successful execution of instructions and the efficiency of the proposed design.

A performance table is presented, comparing the proposed MIPS RISC processor with various latest RISC processors, showcasing its efficiency and competitiveness in terms of delay and maximum operating frequency.

The paper concludes by summarizing the design methodology, VHDL implementation, and successful synthesis and simulation results. The proposed 32-bit MIPS RISC Processor is lauded for its high performance, achieving a delay of 0.741ns and a maximum operating frequency of 1.350 GHz. The clear hierarchy of the design, ease of editing, and successful implementation contributes to the paper's overall conclusion.

LITERATURE REVIEW

"Implementation of a 32-bit MIPS Based RISC Processor using Cadence"

The paper titled "Implementation of a 32-bit MIPS Based RISC Processor using Cadence" [2] by Topiwala and Saraswathi, presented at the 2014 IEEE International Conference on Advanced Communication Control and Computing Technologies (ICACCCT), focuses on the implementation of a 32-bit MIPS-based RISC processor. RISC architecture, known for its streamlined instruction set and emphasis on performance, has been a significant area of research in computer architecture.

The authors highlight the importance of RISC processors in achieving higher speeds through a reduced set of instructions. The chosen approach involves a 5-stage pipelined architecture, comprising Instruction Fetch (IF), Instruction Decode (ID), Execution (EX), Memory Access (MEM), and Write Back (WB) stages. This structure aligns with the MIPS philosophy, emphasizing efficiency in instruction execution.

The study underscores the significance of load/store architecture in MIPS design, emphasizing the advantages of on-chip registers for faster operations compared to memory locations. To enhance the efficiency of the pipeline, the authors introduce a Hazard Detection Unit and a Data Forwarding Unit. These units play a crucial role in identifying and mitigating data hazards, ensuring a smooth execution flow through the pipeline.

The paper acknowledges the growing relevance of RISC processors in embedded systems, particularly in the "system-on-a-chip" paradigm. The authors discuss the trade-offs involved in integrated circuits, considering power consumption, area utilization, and propagation delay. The implementation is done using Verilog-HDL and Cadence tools, with a focus on optimizing power usage.

In terms of instruction types, the authors delve into the MIPS architecture, covering Register Type (R-Type), Immediate Type (I-Type), and Jump Type (J-Type). The discussion provides a comprehensive overview of the instruction set and its impact on the overall design.

The paper addresses power reduction challenges, emphasizing the importance of minimizing unnecessary switching activity to achieve lower dynamic power consumption. Simulation results are presented, and performance metrics such as area, power dissipation, and propagation delay are analyzed using Cadence RTL Compiler with TSMC 0.18μm technology libraries.

In conclusion, Topiwala and Saraswathi's work significantly contributes to the understanding of 32-bit MIPS-based RISC processor design. The incorporation of pipeline architecture, hazard detection, and data forwarding reflects contemporary trends in computer architecture. The study serves as a valuable resource for researchers and practitioners in the fields of VLSI design and computer architecture, offering insights into optimizing performance and power efficiency.

LITERATURE REVIEW

“Design and Implementation of 32-bit Functional Unit for RISC architecture applications”

The paper titled "Design and Implementation of 32-bit Functional Unit for RISC architecture applications" [3] by Rashmi Samanth, Ashwini Amin, and Subramanya G. Nayak presents the design and implementation of a 32-bit functional unit for RISC-based processors. The functional unit includes modules such as Arithmetic and Logic Unit (ALU) that performs operations like addition, subtraction, multiplication, shifting, and code conversion. These operations are controlled by suitable control units and data paths, utilizing multiplexers for selecting various operations based on control inputs. The design is carried out using Hardware Description Language (HDL), and simulation and synthesis are performed using Xilinx ISE to analyze the results.

The introduction outlines the significance of high-speed data path logic system design in VLSI. The authors specifically target the MIPS architecture, a RISC-based processor. The paper aims to enhance the efficiency of the processor by designing a 32-bit ALU using behavioral and structural modeling.

The RISC architecture adopted follows a simple load/store architecture. Operations are performed based on operand registers, and memory access is restricted to load/store instructions. The architecture comprises an ALU, multiplexers, multipliers, and register files. Functional units play a crucial role in manipulating data in the processor's data path.

The proposed 32-bit ALU architecture includes logical units, arithmetic units, multipliers, shifters, and code converters. Operations are controlled by multiplexers that govern memory address computations, arithmetic, and logical operations. The design is implemented in Verilog HDL for RISC-based applications, and simulation is carried out using Xilinx ISE. The comparison with a conventional MIPS ALU demonstrates a reduction in power dissipation by 30.44%, area by 6%, and delay by 34.49%.

In conclusion, the paper successfully implements the functional unit using behavioral and structural modeling. The design, simulated using Xilinx ISE and Cadence RTL Compiler, showcases reduced power, area, and delay compared to a conventional MIPS design. The focus on an energy-efficient ALU suggests potential improvements using low-power techniques.

The references cited in the paper cover various aspects of processor design, providing a comprehensive background for the proposed work.

DESIGN METHOD

Design Principles:

In designing the 32-bit MIPS ALU based on RISC, we adhere to key principles to ensure efficiency, simplicity, and compatibility with the MIPS architecture:

Efficiency:
The ALU is designed to execute instructions in a single clock cycle, optimizing overall processor performance.

RISC Adherence:
A reduced set of instructions is implemented to maintain simplicity and align with RISC principles, promoting faster execution.

MIPS Compatibility:
The design strictly follows the 32-bit MIPS architecture, ensuring seamless integration within MIPS-based systems.

Enhanced Capabilities:
The ALU is tailored to meet the demands of modern computing, handling intricate arithmetic and logic operations efficiently.

Modular Design Approach:

The design embraces a modular approach, breaking down the ALU into distinct modules:

Arithmetic Module:
Incorporates functions for addition, subtraction, and multiplication, ensuring comprehensive arithmetic capabilities.

Logic Module:
Implements logical operations such as AND, OR, and XOR to cover a broad spectrum of computational needs.

Control Module:
Manages control signals, enabling efficient opcode decoding and ensuring smooth operation of the ALU.

Hardware Description Language (HDL):

VHDL is chosen as the Hardware Description Language for its suitability in accurately representing hardware behavior. Key considerations include:

Concurrency:
Leveraging VHDL's strengths in concurrent modeling to accurately capture the parallel nature of hardware operations.

Clarity:
Ensuring the VHDL code is clear, well-structured, and adheres to industry coding standards for maintainability.

The VHDL implementation follows a systematic process:

Behavioral Modeling:
Describing ALU functions in VHDL to accurately represent the intended behavior of each module.

Module Files:
Creating separate VHDL files for each module, promoting modularity and ease of maintenance.

Coding Standards:
Adhering to industry coding standards to ensure clarity, readability, and consistency across the entire codebase.

ALU Architecture Design:

The high-level architecture of the ALU is defined with a focus on data paths, control signals, and registers:

Data Paths:
Identifying and implementing distinct paths for arithmetic and logic operations, optimizing for efficiency.

Control Signals:
Establishing control signals for each module to manage the flow of operations within the ALU.

Registers:
Integrating registers strategically for temporary data storage during computation.

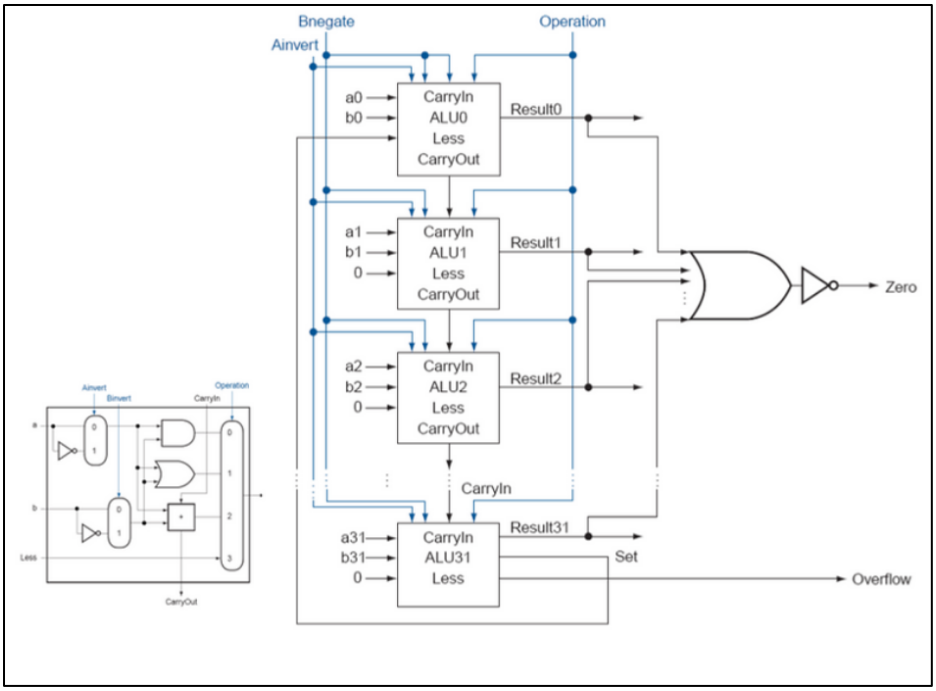

Figure 2: A Symbolic Representation of a 32-bit MIPS ALU Design

Synthesis and Simulation:

The synthesis and simulation steps are crucial for validating the design and preparing it for hardware implementation:

Synthesis:
Using synthesis tools to convert the VHDL code into a netlist, mapping the design onto the target FPGA or ASIC.

Simulation:
Conducting thorough simulations to verify the functionality of the ALU under various scenarios, ensuring accuracy and reliability.

Simulation Software: ModelSim - Intel FPGA Starter Edition

The selection of ModelSim - Intel FPGA Starter Edition for simulation is underpinned by several compelling reasons. First and foremost, it is a cost-free option for students, aligning seamlessly with academic projects. Its robust support for VHDL ensures compatibility with our chosen Hardware Description Language, an essential aspect for our project. Additionally, being a product developed by Intel, ModelSim is well-tailored for FPGA-based projects, making it an optimal choice for testing on **Cyclone V** devices. In the simulation process, ModelSim will be employed for compiling the VHDL code, simulating the behavior of the 32-bit MIPS ALU, and conducting waveform analysis. This comprehensive approach aims to validate the correctness and functionality of our designed ALU under different scenarios.

Testing:

The testing phase is crucial for verifying the functionality and correctness of the ALU design:

Scenario Testing:
Simulating various scenarios to verify the ALU's response to different inputs and conditions.

Waveform Analysis:
Analyzing simulation waveforms to identify any unexpected behavior or issues within the ALU design.

Performance Metrics:

Performance metrics are established to quantitatively assess the effectiveness of the ALU design:

Speed:
Evaluating the execution speed of the ALU to ensure it meets or exceeds performance expectations.

Resource Utilization:
Assessing the efficient use of resources, including FPGA or ASIC resources, to optimize for space and cost.

Power Consumption:
Measuring power consumption during simulation to ensure the ALU design is power-efficient.

Optimization:

The design undergoes iterative optimization to enhance performance and efficiency:

Critical Path Minimization:
Identifying and minimizing critical paths to improve overall execution speed.

Resource Enhancement:
Optimizing resource utilization to ensure efficient use of hardware components**.**

Trade-offs:
Considering trade-offs between speed and efficiency to strike a balance that aligns with project goals.

Integration with MIPS Processor:

Efforts are directed toward seamless integration with the MIPS processor:

Seamless Integration:
Ensuring the ALU integrates seamlessly with the MIPS processor, aligning with the overall system architecture.

Communication:
Verifying effective communication between the ALU and other components within the MIPS-based system.

EXPECTED RESULTS

The successful completion of this research endeavor is anticipated to yield several noteworthy outcomes, affirming the effectiveness and significance of the proposed 32-bit MIPS ALU based on RISC architecture. The expected results encompass both quantitative and qualitative aspects, contributing to the advancement of processor design within the context of the esteemed MIPS architecture.

1. Functional Validity:

The 32-bit MIPS ALU is expected to demonstrate functional validity by accurately executing arithmetic and logic operations as per the defined MIPS instruction set. Simulation results, particularly waveform analyses, will showcase the correct operation of the ALU across various instruction formats, including R-Type, I-Type, J-Type, and I/O-Type.

2. Performance Metrics:

Performance metrics, including but not limited to power consumption, area utilization, and propagation delay, will be meticulously analyzed through simulation using tools such as ModelSim - Intel FPGA Starter Edition. The anticipated results involve achieving a balance between operational efficiency and resource utilization, showcasing competitive performance compared to existing RISC processors.

3. Comparison with Conventional Designs:

A comparative analysis between the proposed 32-bit MIPS ALU and conventional MIPS ALU designs will be conducted to validate the efficiency improvements targeted in this research. Reductions in power dissipation, area utilization, and delay, as indicated in the literature review, are expected to be confirmed through quantitative assessments.

4. Optimized Resource Utilization:

The research aims to optimize the utilization of on-chip resources, particularly registers and memory, for enhanced operational speed. Results are expected to demonstrate an efficient load/store architecture, minimizing data hazards, and enhancing overall performance.

5. Compatibility with Cyclone V Devices:

Given the focus on testing on Cyclone V devices, the expected results include seamless compatibility and functionality of the 32-bit MIPS ALU within the Cyclone V FPGA architecture.

6. Robust VHDL Implementation:

The VHDL implementation of the design is anticipated to be robust and easily comprehensible, providing a foundation for future enhancements and modifications. VHDL code analysis will affirm its adherence to best practices and standards, ensuring long-term viability and ease of maintenance.

The culmination of these expected results aims to contribute substantively to the field of computer engineering, providing insights into efficient processor design within the RISC architecture, and specifically, the venerable MIPS framework. These outcomes will serve as valuable benchmarks for evaluating the success of the proposed research.

CONCLUSION

In conclusion, this research proposal seeks to contribute significantly to the field of computer engineering by designing a 32-bit MIPS ALU based on RISC principles. Grounded in a thorough literature review, the proposed research methodology combines key insights from relevant works. The expected results aim to demonstrate the functional efficacy and performance metrics of the designed ALU, emphasizing its compatibility with Cyclone V devices. This research aspires to not only advance technical knowledge in processor design but also to address the contemporary challenges in achieving an optimal balance between efficiency and complexity. If achieved, this project promises valuable contributions to the broader discourse on processor optimization and the evolution of efficient ALU design within the context of MIPS architecture.

REFERENCES

[1] S. P. Ritpurkar, M. N. Thakare and G. D. Korde, "Synthesis and Simulation of a 32Bit MIPS RISC Processor using VHDL", 2014 International Conference on Advances in Engineering & Technology Research (ICAETR - 2014), Unnao, India, 2014, pp. 1-6, doi: 10.1109/ICAETR.2014.7012843.

[2] M. N. Topiwala and N. Saraswathi, "Implementation of a 32-bit MIPS based RISC processor using Cadence," 2014 IEEE International Conference on Advanced Communications, Control and Computing Technologies, Ramanathapuram, India, 2014, pp. 979-983, doi: 10.1109/ICACCCT.2014.7019240.

[3] R. Samanth, A. Amin and S. G. Nayak, "Design and Implementation of 32-bit Functional Unit for RISC architecture applications," 2020 5th International Conference on Devices, Circuits and Systems (ICDCS), Coimbatore, India, 2020, pp. 46-48, doi: 10.1109/ICDCS48716.2020.243545.