Project Proposal: Full Implementation of MIPS on an FPGA

ES215 - Computer Organization and Architecture
Prof. Sameer Kulkarni

Group Name: RAMpage
Group Members and Roll Numbers:
Aditya Mehta (22110017)
Jiya Desai (22110107)
Pratham Sharda (22110203)
Sujal Patel (22110261)

August 31, 2024

1 Introduction

The objective of this project is to design and implement a complete MIPS (Microprocessor without Interlocked Pipeline Stages) architecture for a RISC (Reduced Instruction Set Computer) processor on an FPGA. This implementation will cover the fundamental instruction set architecture (ISA) of MIPS. The programs are meant to be stored in the FPGA's block RAM. The output will be validated using the 7-segment LED display of the FPGA, using memory-mapped I/O. The project aims to provide a detailed understanding of MIPS architecture and hands-on experience with FPGA-based processor design.

We chose to implement MIPS instead of ARM (Advanced RISC Machine), on learning that MIPS is simpler to implement and more fundamental for understanding processor design. Having worked with FPGAs in our course Digital Systems, we opted for this project to leverage our current understanding of FPGA for processor design applications.

2 Envisaged Task Breakdown

The project has been tentatively broken down into the following tasks:

1. **Understanding MIPS Architecture:** Studying the MIPS instruction set architecture, including instruction formats, types of instructions (data manipulation, data transfer, control transfer, special instructions), determining the type and number of registers and memory address modes.

2. **Designing the Data Path and Implementing Control Unit:** Creating a detailed design of the MIPS data path, including the ALU, registers, and memory interfaces, and designing a control unit to manage the execution of instructions by generating appropriate control signals.

3. **FPGA Implementation - Harnessing Block RAM:** Understanding the role of Block RAM as the memory storage unit of the FPGA.

4. **FPGA Implementation of the Design:** Mapping the MIPS architecture to the FPGA, syncing the architecture bit size with the constraints of the FPGA and developing a Python-based assembler.

5. **Memory Mapping for Input/Output Operations:** Enabling the architecture to fetch the input provided through the FPGA switches, and fetch the program output from the memory to be displayed on the 7-segment display.

6. **Testing and Validation:** Displaying the output of programs on the 7-segment LED display of the FPGA and verifying the results.
7. **Documentation:** Preparing a detailed report and presentation on the design, implementation, and results.

3 Key Milestones

The key milestones for the project are as follows:

- **Week 1-3:** Understanding MIPS Architecture; Designing the Data Path and Implementing Control Unit
- **Week 4:** FPGA Implementation - Harnessing Block RAM
- **Week 5-8:** FPGA Implementation of the Design
- **Week 8-10:** Memory Mapping for Input/Output Operations; Testing and Validation
- **Additional target (if time permits):** Development and implementation of a fully functional pipeline on the FPGA that can directly accept high-level code or assembly-level instructions as input, and perform the conversion to machine instructions and subsequent computations directly on the board (without an external assembler). (We may have to use multiple boards or a stronger FPGA board to make this possible)

4 Tools and Technologies to be Used

The following tools and technologies will be employed in this project:

- **FPGA Development Board:** BASYS-3
- **Hardware Description Language:** Verilog/VHDL
- **Synthesis and Simulation Tools:** Xilinx Vivado
- **Testing Framework:** Custom test benches in Verilog/VHDL
- **Programming Language for Dependency Programs:** Python

5 Expected Outcomes

By the end of the project, the following outcomes are expected:

- A detailed understanding of the fundamentals of processor design.
- A full FPGA implementation of MIPS architecture for a RISC processor with a functional data card, enabled to present the program output on the 7-segment LED display of the FPGA using memory-mapped I/O.
- A comprehensive report detailing the design and implementation process.

6 References

- Course Lectures, ES215 - Computer Organization and Architecture.
- Design and Implementation of RISC MIPS Processor on FPGA, IJRASET.