



Aiden Contini- s3780445

Daniel brennan- s3850688

EEET2261- Computer Architecture and Organisation

Instruction set architecture and cpu design

*Engineers Without Borders Challenge*

**Contents**

[Introduction 2](#_Toc135409741)

[Background 2](#_Toc135409742)

[Design 3](#_Toc135409743)

[Instructions 3](#_Toc135409744)

[J-Type 3](#_Toc135409745)

[B-Type 3](#_Toc135409746)

[I-Type 3](#_Toc135409747)

[S-Type 3](#_Toc135409748)

[Minor Building Blocks 3](#_Toc135409749)

[Comparator 3](#_Toc135409750)

[Adder 3](#_Toc135409751)

[Major Building Blocks 3](#_Toc135409752)

[ALU 3](#_Toc135409753)

[Register File 3](#_Toc135409754)

[Program Memory 3](#_Toc135409755)

[Program Counter 3](#_Toc135409756)

[Instruction Memory 3](#_Toc135409757)

[Control Unit 3](#_Toc135409758)

[CPU Design 3](#_Toc135409759)

[Results 3](#_Toc135409760)

[Conclusion 4](#_Toc135409761)

# Introduction

Our group project focuses on the design and implementation of a CPU and instruction set capable of performing a linear search to find the minimum value in a given array. In this project, we aim to apply our knowledge and skills gained from previous laboratory experiments, where we designed and simulated components such as the ALU, program counters, multiplexers, register file, and adders. By undertaking this project, we aim to deepen our understanding of instruction set architecture (ISA) and computer organization, and apply this knowledge to create a functional CPU that meets the specified criteria.

# Background

To successfully accomplish our project goals, we need to develop a solid understanding of the theoretical principles and technical aspects of instruction set architecture (ISA) and computer organization. By comprehending these fundamental concepts, we can make informed decisions during the design process and create an effective instruction set architecture that supports our specific application.

Additionally, our previous laboratory experiments have equipped us with practical knowledge and experience in designing and simulating individual components of a digital system, such as the ALU, program counters, multiplexers, register file, and adders. This prior knowledge will be valuable as we integrate these components into a complete CPU datapath for our project.

The design of our instruction set architecture must consider the requirements of the linear search operation. We need to carefully select instructions that enable memory access, perform logical and arithmetic operations, and facilitate control transfers like conditional branching and jumping. Our instruction set should provide efficient functionality for conducting the linear search operation on the supplied array.

Furthermore, the design and implementation of a functional CPU datapath are crucial aspects of our project. We will leverage the design principles and knowledge gained from our previous work on components like the ALU, program counters, multiplexers, register file, and adders. While we won't directly use all of the same specific components, we will apply similar design principles to integrate these individual elements into a cohesive CPU architecture. By building upon our previous experience, we can ensure that the components work harmoniously together to form a functional and efficient CPU.

This CPU datapath should be capable of executing multiple instructions, including fetching and decoding instructions, performing memory read and write operations, carrying out logical and arithmetic operations, and supporting control transfer instructions.

To facilitate the design and verification processes, we will utilize the Intel Quartus Prime software suite, which offers a range of tools for schematic capture, functional simulation, synthesis, optimization, and verification. This software will enable us to simulate the behaviour of our CPU design, ensuring that it aligns with our intended objectives and meets the specified constraints.

# 

# Design

## Instructions

### J-Type

#### JAL

#### JALR

### B-Type

#### BEQ

#### BLT

### I-Type

#### LW

#### ADDI

### S-Type

#### SW

## Minor Building Blocks

### Comparator

### Adder

## Major Building Blocks

### ALU

### Register File

### Program Memory

### Program Counter

### Instruction Memory

### Control Unit

## CPU Design

# Results

# Conclusion