

MULTI CYCLE PIPELINED RISCv BASED PROCESSOR

TEAM 11

**CA Project Plan**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Document Version** | **Remarks** | **Drafted by** |
| 10/09/ 2024 | Version 1.0 | Design Overview | TEAM 11 |

**PROJECT DOCUMENT- MULTI CYCLE PIPELINED RISCv BASED PROCESSOR**

Table of Contents

[CHAPTER 1 – DESIGN OVERVIEW 4](#_Toc176875050)

[1.1 INTRODUCTION TO RISCV 4](#_Toc176875051)

[1.2 ARITHMETIC AND LOGIC UNIT 4](#_Toc176875052)

[1.2 KEY FEATURES 4](#_Toc176875053)

[1.5 PROJECT OVERVIEW 4](#_Toc176875054)

[1.7 INSTRUCTION FORMAT: 5](#_Toc176875055)

[1.6 MICRO - ARCHITECTURE: 5](#_Toc176875056)

[1.7 ARCHITECTURE COMPONENTS 6](#_Toc176875057)

[1.7.1 Program Counter 6](#_Toc176875058)

[1.7.2 Program Memory 6](#_Toc176875059)

[1.7.3 ALU 7](#_Toc176875060)

# CHAPTER 1 – DESIGN OVERVIEW

## 1.1 INTRODUCTION TO RISCV

## 

The RISC-V instruction set has a 32-bit fixed length and follows a little-endian format. It includes 32 general-purpose registers, with reg0 fixed at 0, and a 32-bit program counter that increments by one for word-addressable instruction memory. Chosen for its efficiency, RISC-V is pipeline-friendly and requires minimal hardware and power. Techniques like loop unrolling and compiler scheduling are used to optimize performance. Supports six instruction formats: R-type, U-type, I-type, B-type, J-type, and S-type.

## 1.2 ARITHMETIC AND LOGIC UNIT

An Arithmetic Logic Unit (ALU) is a digital circuit used to perform arithmetic (e.g., addition, subtraction) and logical (e.g., AND, OR, XOR) operations. It is a combinational circuit. It is a critical component within the CPU of a computer. It is responsible for performing a variety of arithmetic and logical operations on binary numbers.

## 1.2 KEY FEATURES

* Operations: ADD, SUB, AND, OR, XOR ,NOP.
* Develop a multi-cycle 5 stage pipelined processor
* For each operation implement the corresponding instruction of RISCv

## 

## 1.5 PROJECT OVERVIEW

* Designing Micro-architecture, define the project's micro-architecture and functional blocks.
* Implement ADD, SUB, AND, OR, XOR and NOP operations using corresponding RISC-V instructions.
* Create a testbench to verify the functionality of the processor and each ALU operation
* Develop a multi-cycle 5-stage pipelined processor architecture.
* Pipeline Stages: The processor pipeline is divided into different stages, Instruction Fetch (IF), Instruction Decode (ID), Execute (EX) and Write Back (WB)
* Developing various stages which includes Program Counter, Program memory, Instruction decoder, ALU and Load Block and developing their HDL codes using Verilog

## 1.7 INSTRUCTION FORMAT

A table with numbers and symbols

Description automatically generated

## 1.6 MICRO - ARCHITECTURE

A diagram of a computer

Description automatically generated

## 1.7 ARCHITECTURE COMPONENTS

### 1.7.1 Program Counter

A black background with a black square

Description automatically generated with medium confidence

* The PC, or program counter, is a key CPU register that keeps track of the order in which instructions are executed.
* The PC is in sync with the system clock and stores the memory address of the subsequent instruction that needs to be retrieved.
* Every time there is a clock pulse, the CPU retrieves the instruction from the PC's address and advances the PC to the subsequent instruction.
* Through this procedure, the CPU is guaranteed to carry out instructions in the proper order, facilitating seamless and effective system operation.

### 1.7.2 Program Memory

A black and white photo of a memory card

Description automatically generated

* The instructions that the CPU needs to carry out are kept in the program memory.
* The next instruction is fetched and transmitted to the control unit and decoder after the Program Counter (PC) locates its precise memory address.
* By disassembling the instruction into its constituent parts, such as register addresses, opcodes, and functions, the decoder is able to understand it. In order to guarantee that the command is carried out properly, the control unit subsequently routes this data to the relevant CPU components. Through the translation and orderly execution of instructions, this process makes it possible for the CPU to function smoothly.

### 1.7.3 ALU

A white rectangular object with black text

Description automatically generated

* The Arithmetic Logic Unit (ALU) of a CPU is where all arithmetic and logic operations are carried out.
* The resultant data must be stored for later use after these processes are completed.
* Through a procedure known as "write-back," the outcome is restored into the register set.
* The CPU's registers are quick, compact storage spaces that momentarily store data.

The CPU can swiftly retrieve the data for next operations by storing the results in registers,

guaranteeing effective instruction processing and execution