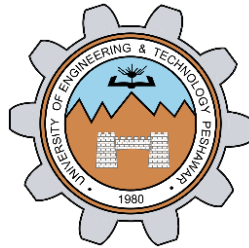


# **Project Proposal**



**Fall 2023**

**CSE-304L Computer Organization and Architecture Lab**

Submitted by:

**Shahzad Bangash (21PWCSE1980)**

**Suleman Shah (21PWCSE1983)**

**Ali Asghar (21PWCSE2059)**

**Muhammad Shahab(21PWCSE2074)**

Class Section: **C**

Submitted to:

**Dr. Bilal Habib**

Date:

**29<sup>th</sup> December 2023**

**Department of Computer Systems Engineering**  
**University of Engineering and Technology, Peshawar**

## **Problem Statement:**

The given problem is to design and implement a Simple As Possible (SAP-1) computer using ModelSim and Verilog. The SAP-1 design is a great way to understand how a computer works at the fundamental level. To implement SAP-1, there are multiple methods. One method is to model it using a hardware descriptive language like Verilog.

## **Objectives:**

- To understand the architecture of a basic computer (SAP-1).
- To design the SAP-1 computer using Verilog.
- To simulate the design using Modelsim.

## **Methodology:**

### **Design:**

The SAP-1 architecture will be designed using Verilog. This includes components such as the Program Counter, Memory Address Register, Instruction Register, Controller-Sequencer, Accumulator, Adder-Subtractor, and Output Register. Total 10 components in SAP-1.

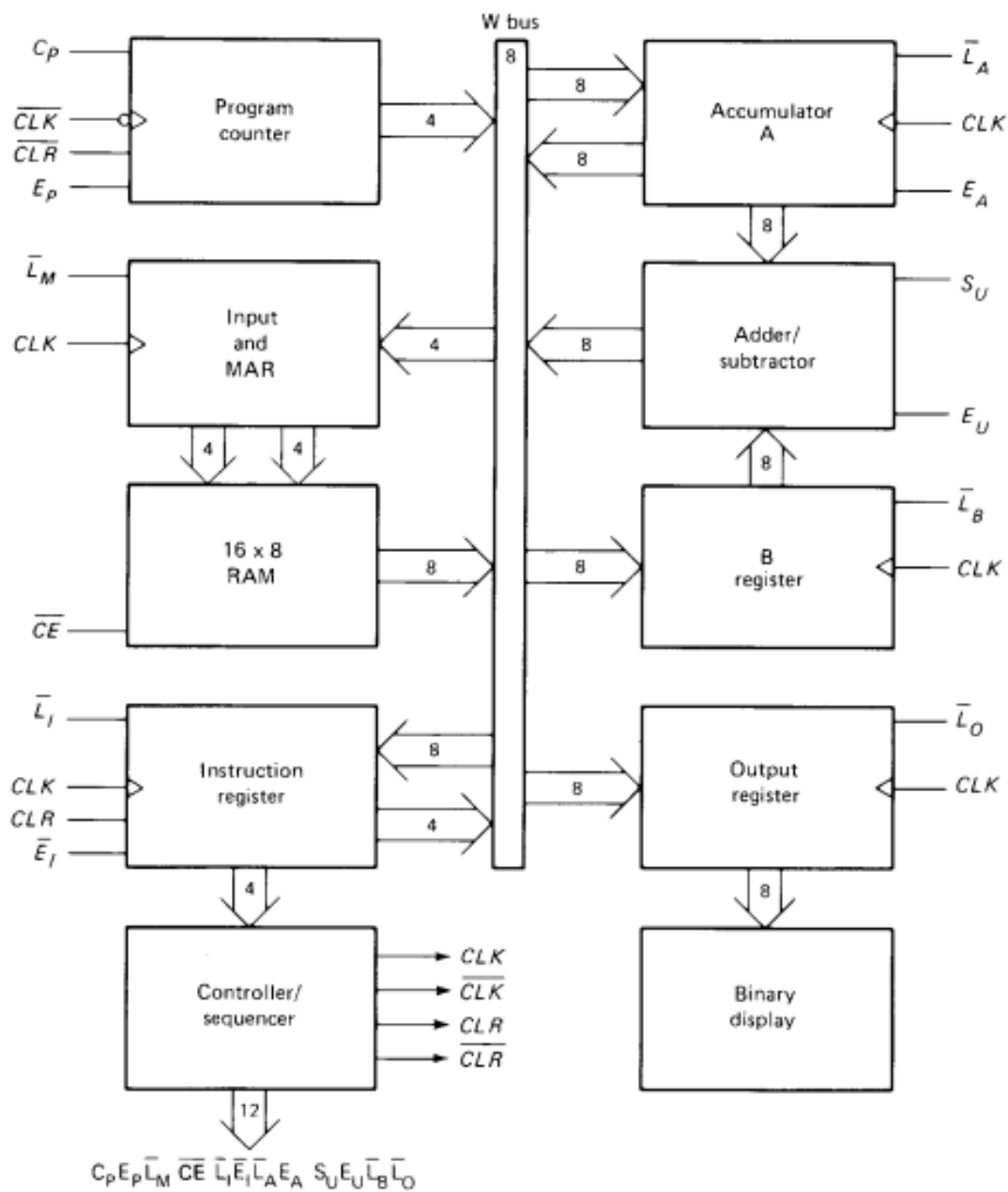
### **Simulation:**

The design will be simulated using ModelSim to verify its functionality. This includes loading instructions into memory and observing the execution of these instructions.

### **Testing:**

Various test cases will be developed to ensure the SAP-1 computer operates as expected.

## Block Diagram:



## **Instruction Set:**

There are a total of five instructions in SAP-1. They are described below one by one.

1. **LDA (Load):**

This instruction loads the data from a specific memory location into the accumulator. The memory address is specified in the operand.

2. **ADD:**

This instruction adds the data from a specific memory location to the data in the accumulator. The result is stored in the accumulator. The memory address is specified in the operand.

3. **SUB (Subtract):**

This instruction subtracts the data at a specific memory location from the data in the accumulator. The result is stored in the accumulator. The memory address is specified in the operand.

4. **OUT (Output):**

This instruction sends the contents of the accumulator to the output register, which can then be displayed.

5. **HLT (Halt):** This instruction stops the execution of the program.

## **Conclusion:**

At the end of this project, we expect to have a fully functional SAP-1 computer simulated in ModelSim. This project will provide us with a practical understanding of computer architecture and digital design.