SAP – 1

ARCHITECTURE

LAB PROJECT REPORT

Fall 2023

CSE-304L

Computer Organization & Architecture Lab

Submitted by:

AIMAL KHAN [21PWCSE1996],

HAMZA [21PWCSE1990],

SARWAT [21PWCSE2014]

Class Section: **A**

Submitted to:

Dr. Bilal Habib

Thursday, February 1, 2024

Department of Computer Systems Engineering

University of Engineering and Technology, Peshawar

Overview of the project:

This project explores the Simple-As-Possible-1 (SAP-1) computer architecture using Verilog in the Modelsim Toolchain. The objective is a hands-on investigation of SAP-1's architecture, instruction set, microprogramming, and control unit implementation. Through practical application, this project bridges theory and implementation, providing valuable insights into digital design and computer architecture.

Introduction to SAP-1:

The Simple-As-Possible-1 (SAP-1) computer architecture, crafted by Albert Paul Malvino and Donald Leach, serves as an entry point into the world of computers for learners. It's a basic model designed to make complex concepts understandable.

Key components:

Arithmetic and Logic Unit (ALU):

The ALU is the computational heart of SAP-1. It performs basic arithmetic operations (addition and subtraction) .

Accumulator (AC): The accumulator is a register that stores the results of arithmetic and logical operations. It is a central component for temporary data storage during computation.

Memory Address Register (MAR):

The MAR holds the address of the memory location from which data is to be fetched or to which data is to be written.

Instruction Register (IR):

The IR holds the current instruction being executed by the SAP-1. It is used to decode the instruction and determine the required operations.

Program Counter (PC):

The PC keeps track of the memory address of the next instruction to be fetched. Instruction Decoder: The instruction decoder interprets the instruction stored in the IR and generates control signals to coordinate the operation of other components accordingly.

Control Unit: The control unit manages the sequencing and coordination of various operations within the SAP-1. It generates control signals to synchronize the activities of different components.

W Bus (Working Bus): The W bus serves as a data bus that facilitates the transfer of data between different components.

Fetch and execution cycle:

Fetch Cycle:

Program Counter Increment:

Program Counter (PC) increments to the next memory address.

Memory Access:

Memory reads the instruction at the incremented address.

Execution Cycle:

Instruction Decoding:

Control unit decodes the instruction in the IR.

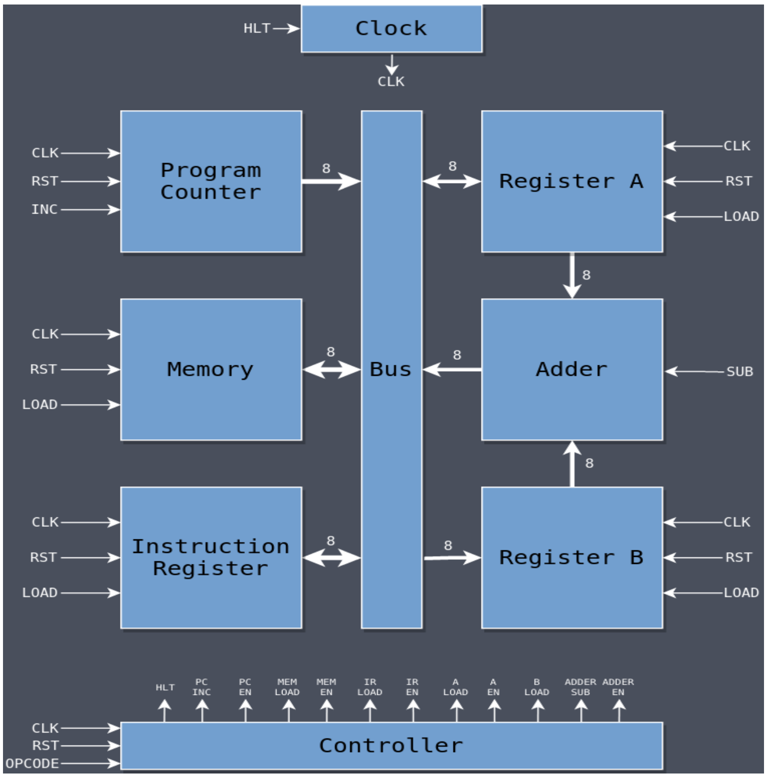
ALU Operations:

ALU performs arithmetic/logic operations based on the decoded instruction.

Data Transfer:

Data may be transferred between registers.

Output: Results may be transferred to output registers for display or further processing.



Anticipated Results:

Operational SAP-1 Prototype:

We aim to achieve a fully functional SAP-1 model within the Modelsim environment, demonstrating the execution of instructions according to SAP-1 specifications.

Enhanced Comprehension of Computer Architecture:

The hands-on implementation of SAP-1 will contribute to a deeper understanding of fundamental Computer Organization and Architecture (COA) principles.

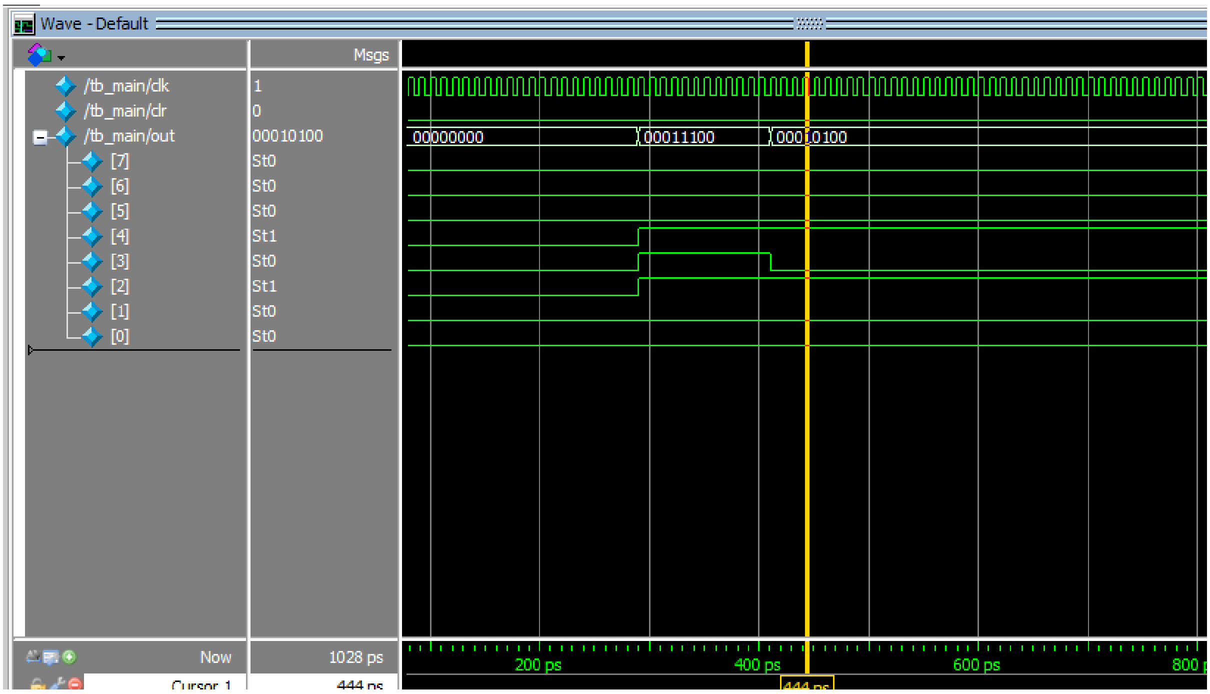
Skill Advancement:

The project is anticipated to foster improved proficiency in both Verilog and the Modelsim design suite. Furthermore, honing robust debugging skills is expected as an integral part of the learning process.

Code:

```
/ 0 to 7 is alloted for program memory and 8 to f is alloted for data
// lets implement 16 + 20 + 24 - 32.
            4'h0: data = 8'b0000_1001; // LDA 9h
            4'h1: data = 8'b0001_1010; // ADD Ah
            4'h2: data = 8'b0001_1011; // ADD Bh
            4'h3: data = 8'b0010_1100; // SUB Ch
            4'h4: data = 8'b1110_0000; // OUT
            4'h5: data = 8'b0010_1101; // SUB Dh
            4'h6: data = 8'b1110_0000; // OUT
            4'h7: data = 8'b1111_0000; // HLT
            4'h8: data = 8'b0000_0000;
            4'h9: data = 8'b0001_0000; // 16
            4'ha: data = 8'b0001_0100; // 20
            4'hb: data = 8'b0001_1000; // 24
            4'hc: data = 8'b0010_0000; // 32
            4'hd: data = 8'b0000_1000; // 8
            4'he: data = 8'b0000_0000;
            4'hf: data = 8'b0000_0000;
```

Output:



```
/tb_main/clk    1
/tb_main/clr    0
/tb_main/out    00010100
  [7]           St0
  [6]           St0
  [5]           St0
  [4]           St1
  [3]           St0
  [2]           St1
  [1]           St0
  [0]           St0
Now             1028 ps
Cursor 1        444 ps
```

Applications of SAP-1:

Educational Tool:

SAP-1 serves as a practical and hands-on educational resource for teaching computer architecture fundamentals.

Digital Design Courses:

Valuable in digital design courses, SAP-1 aids in understanding the connection between logic design, hardware components, and computer functionality.

Assembly Language Programming:

Ideal for learning assembly language programming, SAP-1 provides insights into machine instructions and hardware operations.

Historical Perspective:

Studying SAP-1 provides a historical context for early computer architectures, contributing to an understanding of their evolution.

Conclusion:

In this project, by putting SAP-1 into action using Modelsim, our goal is to connect the dots between computer architecture theory and real-world application, all within the realm of computer simulation.