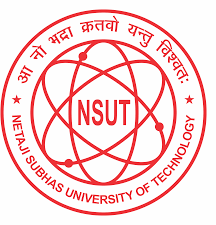
**Digital System Design**

****

**Netaji Subhas University of Technology (Formerly known as Netaji Subhas Institute of Technology)**

**Sector - 3, Dwarka, New Delhi-110078**

**SUBMITTED TO: - SUBMITTED BY: -**

Dr. Kunwar Singh Name: Deepak Kumar

Dept. of Electronics and Roll No.: 2023PEV3227

Communication Engineering Course: MTech. (E-VLSI)

**AIM**

Implementation of 8-bit RISC Processor architecture.

**THEORY**

A RISC-SPM is a type of computer architecture that uses a reduced instruction set computer (RISC) processor, a controller, and a memory. The processor consists of an arithmetic and logic unit (ALU), a set of registers, and a multiplexer. The controller is responsible for fetching, decoding, and executing instructions from the memory, and controlling the data paths and the memory access. The memory stores both the program instructions and the data.

A RISC-SPM can perform various operations, such as arithmetic, logic, memory access, branching, and looping, using a small and simple set of instructions. A RISC-SPM is designed to simplify the hardware and improve the performance and efficiency of the computer.

A RISC-SPM has three functional units: a processor, a controller, and a memory. The processor has an ALU, a set of registers, and a multiplexer. The controller has a finite state machine (FSM) that controls the instruction sequence and the data paths. The memory has a word-addressable memory that stores the instructions and the data.

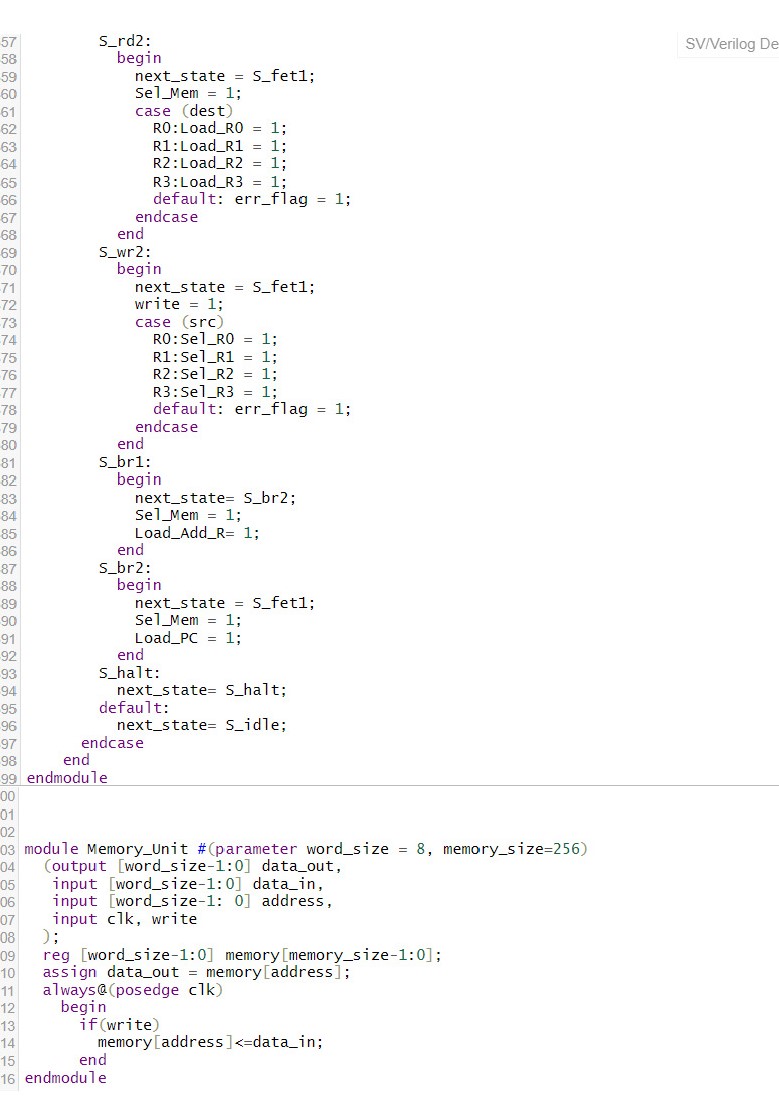
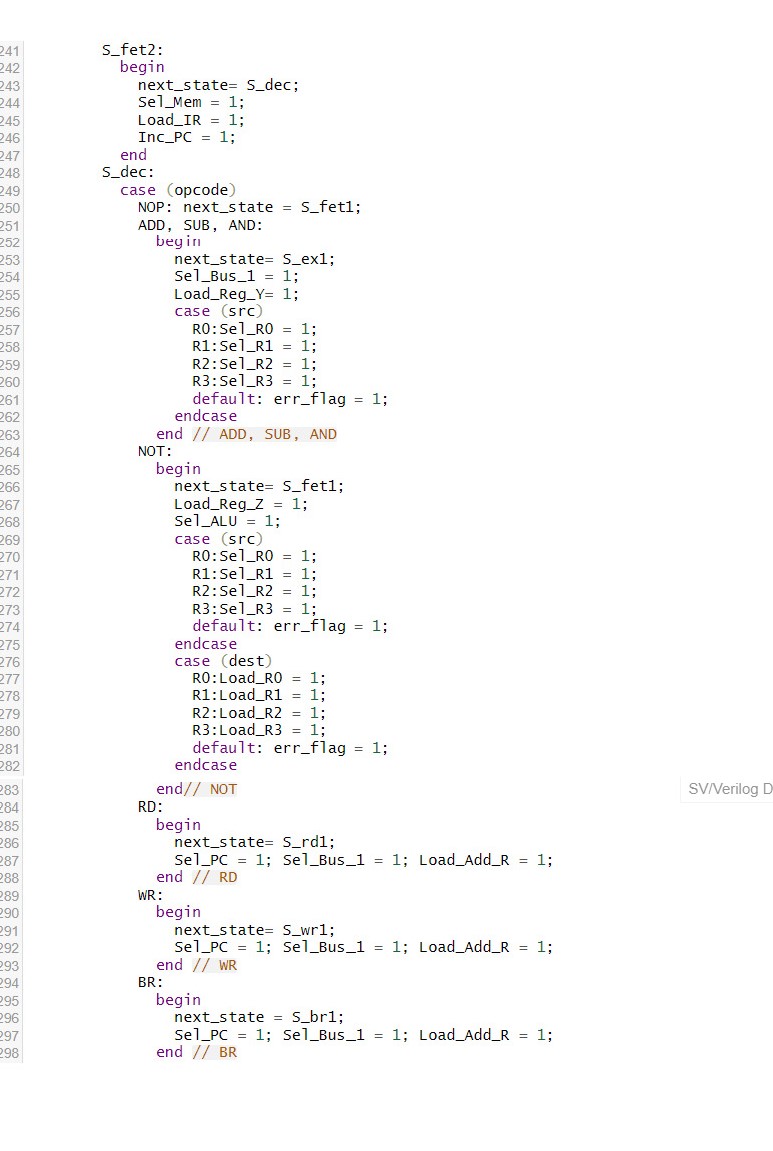
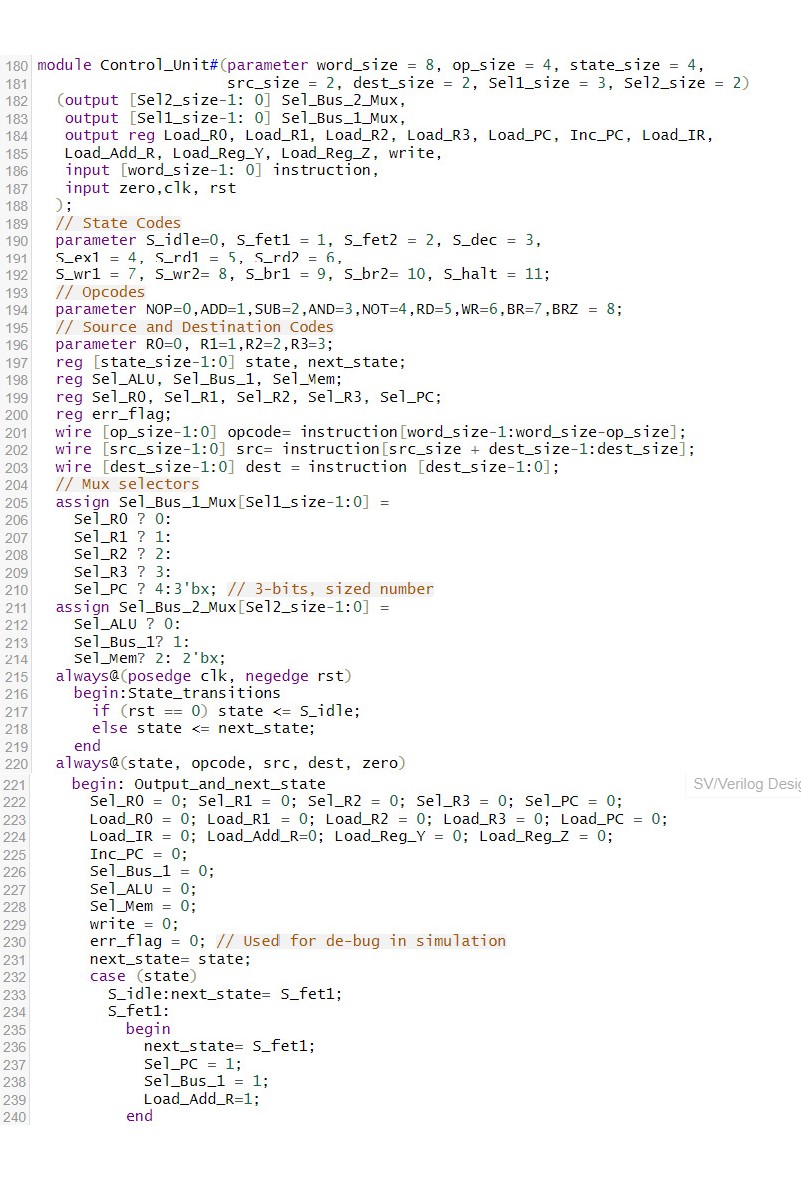
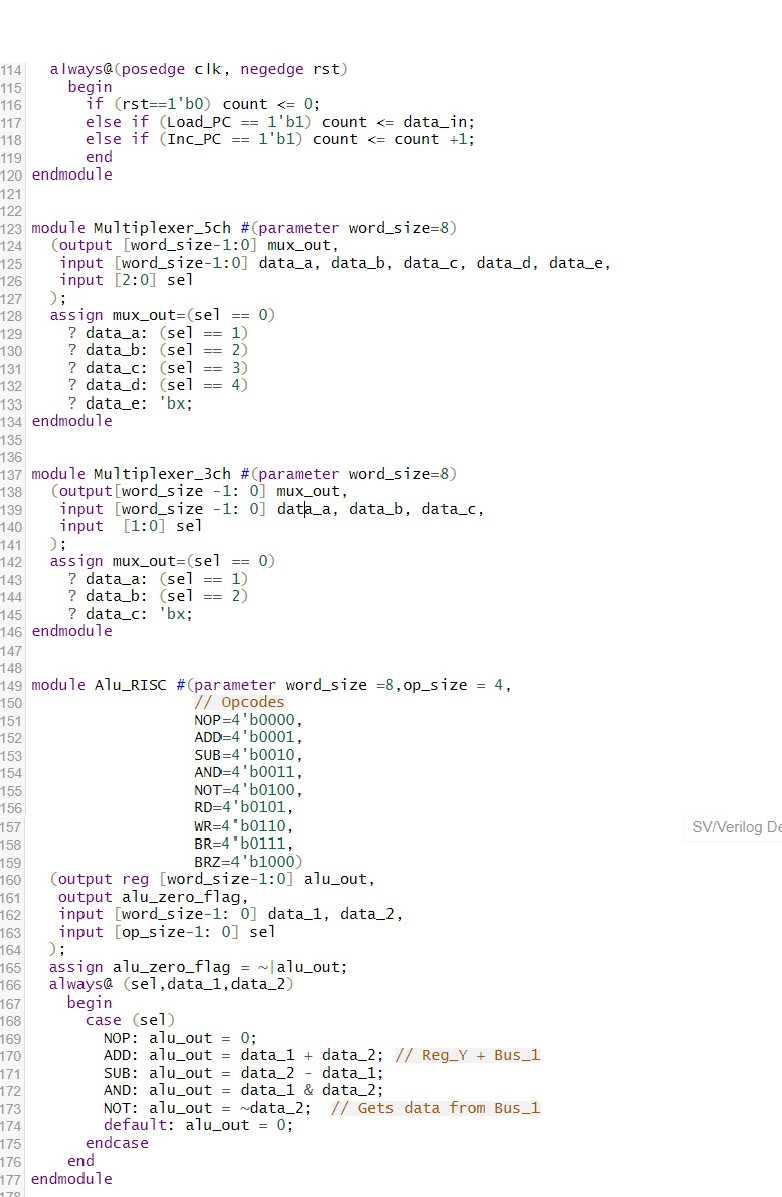
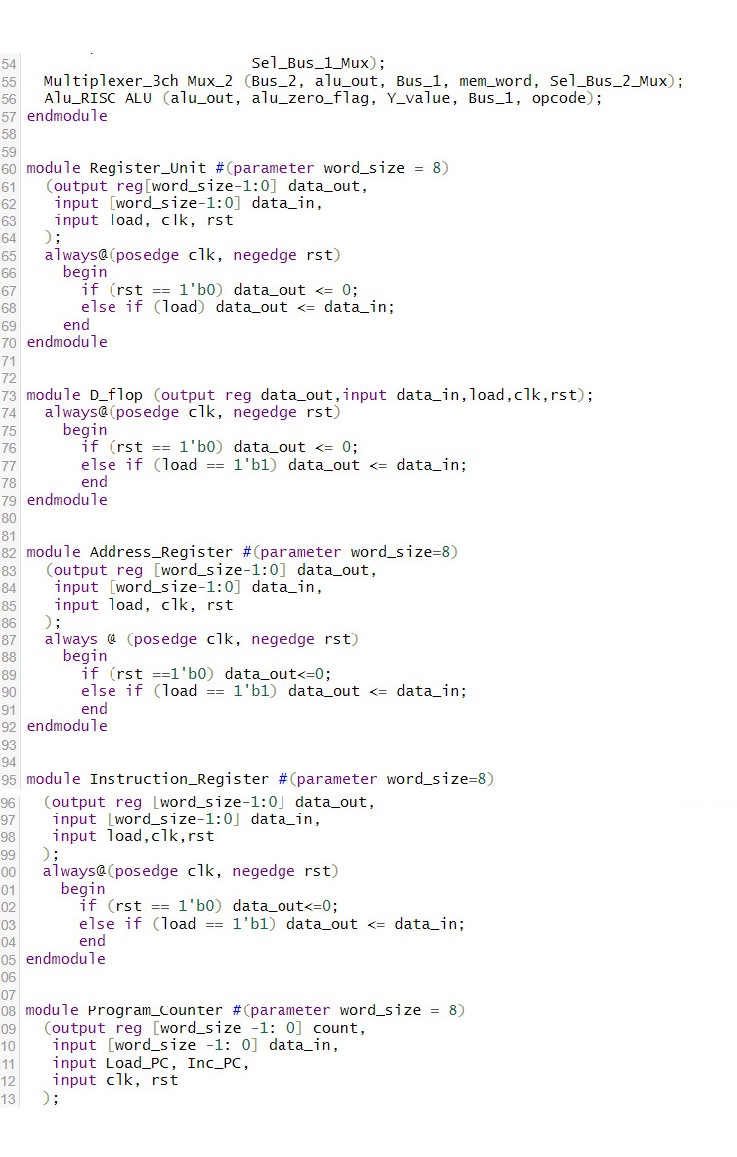
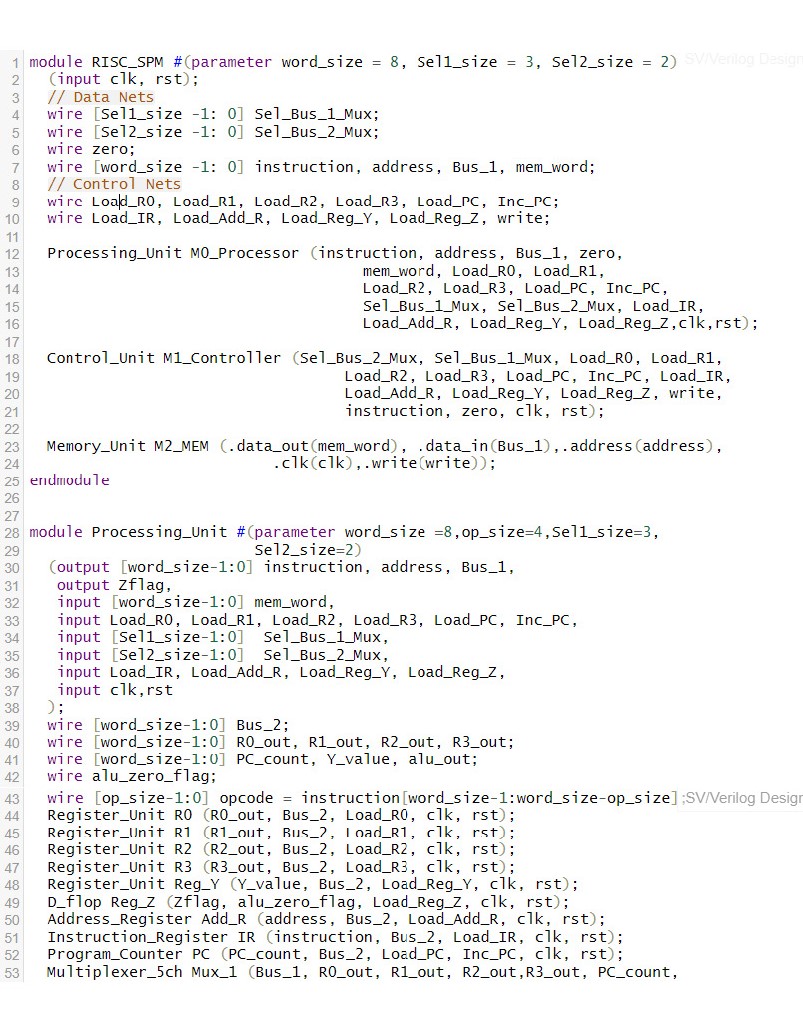
A RISC-SPM has two types of instructions: short instructions and long instructions. Short instructions are 8 bits long and perform basic arithmetic and logic operations on the registers. Long instructions are 16 bits long and perform memory access and data movement operations on the registers and the memory.

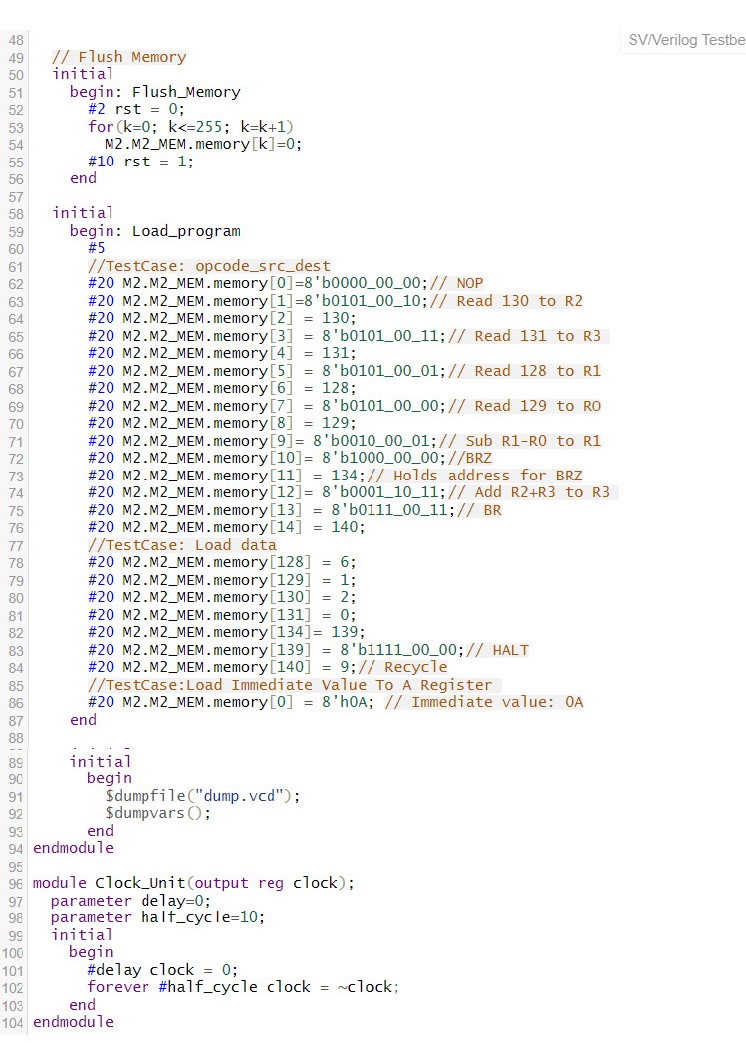
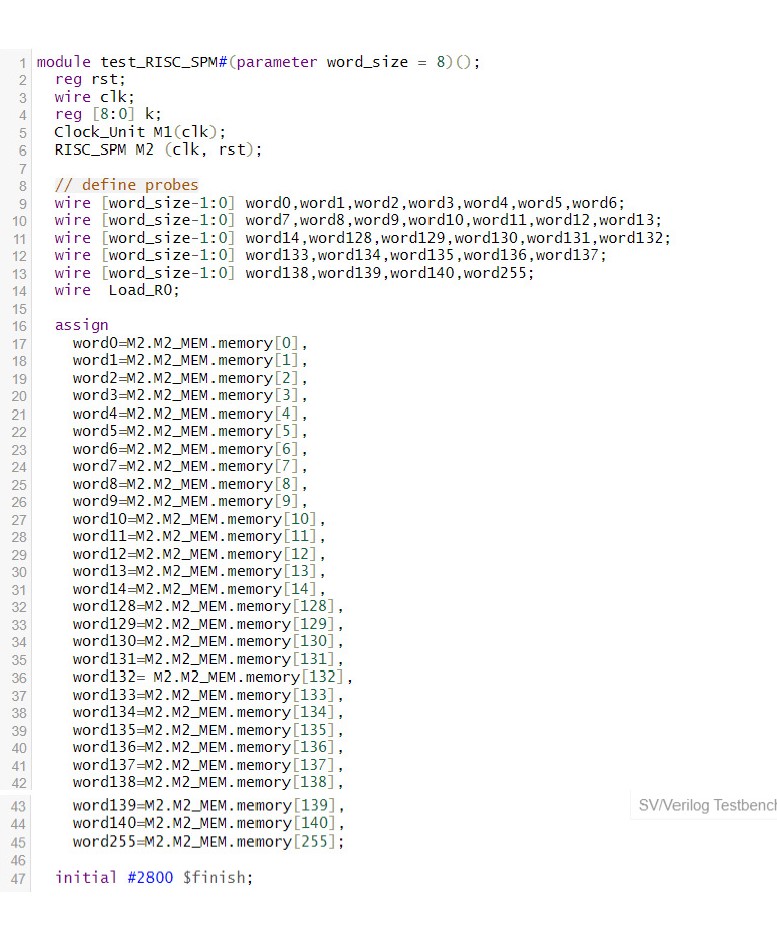
A RISC-SPM has several control signals that determine when to load registers, select the path of data through the multiplexers, write data to memory, and control the three-state busses in the architecture. The controller generates the control signals based on the instruction set and the FSM.

**Architecture**

A diagram of a computer program

Description automatically generated

**Risc\_Verilog\_code:** 

**Testbench:**

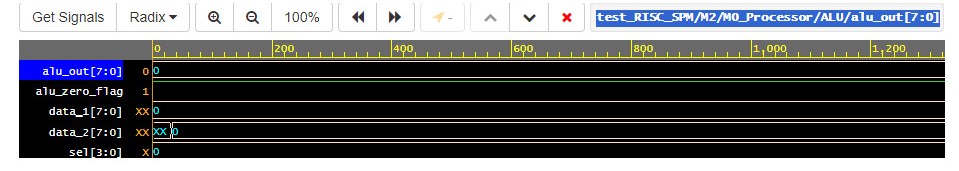
**Result**

1. **Input Data to Register\_File :**

A screenshot of a computer

Description automatically generated

1. ALU



1. M0\_processor

A screenshot of a computer

Description automatically generated

1. Memory

A screenshot of a video player

Description automatically generated

1. Controller

A screenshot of a computer

Description automatically generated