A Simple Von Neumann-Based Processor Implementation on FPGA

Enes ORHAN

Istanbul / Turkey

e-mail: enesorhan\_1366@hotmail.com

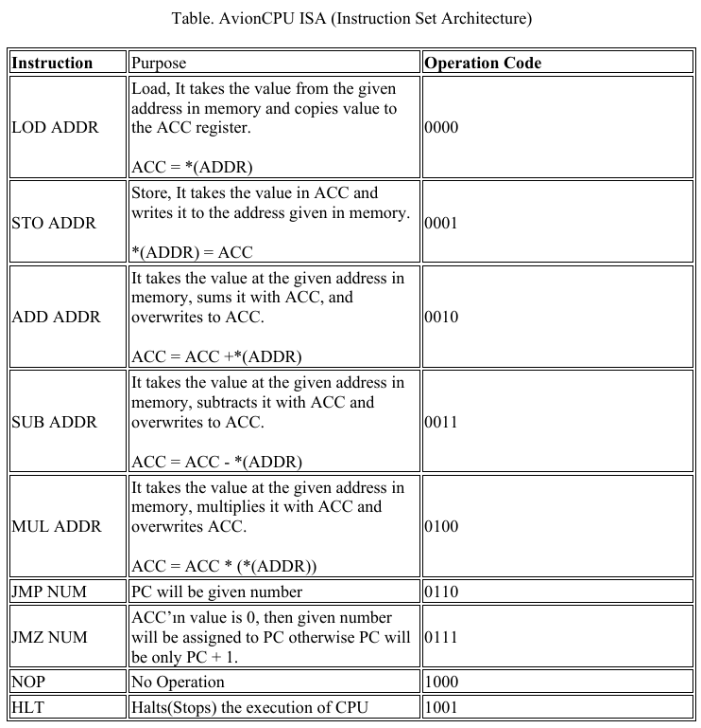
*Abstract* — In this study, a simple processor named AvionCPU was designed and implemented using Verilog HDL, based on the Von Neumann architecture. The processor supports 9 custom-defined instructions, operates through a state machine with 5 states, and includes modules such as RAM, ALU, registers, and a control unit. The functionality of the processor was verified through predefined instruction sets and waveform simulation. The project demonstrates the fundamental operation of instruction fetch, decode, execute, and memory write/read cycles.

Keywords — FPGA, CPU, Von Neumann Architecture, Verilog

# Introductıon

The purpose of this project is to design a basic processor architecture to gain hands-on experience with the working principles of embedded systems and digital processors. The processor was built in accordance with the Von Neumann architecture, where instruction and data share the same memory. This project helped reinforce the understanding of instruction execution cycles, state machine design, and ALU operations.

# System Archıtecture

The AvionCPU was developed using Verilog HDL. The development environment used was Vivado 2023.2, and simulations were carried out using the XSIM simulator.

The processor includes:

* A 10-bit instruction format with a 4-bit opcode and 6-bit address field.
* Four key modules: RAM, ALU, Control Unit, and Register Set.
* A finite state machine with five states: Fetch, Decode, Execute, etc.

The processor supports 64 addressable memory locations, and all control logic is embedded inside the state transitions.

# Software Used

Three test cases were provided to verify the correctness of the CPU:

* Test Case 1: Load two numbers from memory, add them, and store the result.
* Test Case 2: Multiply two numbers using the MUL instruction and store the result.
* Test Case 3: Multiply two numbers without using MUL, only using loops and ADD.

Waveform simulations confirmed that the processor correctly executed all instructions, updated the program counter, and interacted with RAM as expected.

# Results

The processor supports the following instructions:

LOD, STO, ADD, SUB, MUL, JMP, JMZ, NOP, HLT.

Key takeaways from this project:

* Practical implementation of instruction cycle
* Designing and simulating an ALU
* Managing control flow using finite state machines
* Understanding the relationship between hardware modules and software instruction sets

##### Project team

Enes ORHAN

A graduate student in Computer Engineering at Marmara University. Electrical and Electronics Engineering background. A highly motivated and dedicated engineer currently working as a Software Development Engineer in the R&D department of a robotics company. Strong focus on practical innovation, real-time systems, and advancing automation technologies.

##### Reference Files

YouTube Presentation Video: (to be added after upload)

GitHub Repository: https://github.com/enesorhaan/avion\_cpu