8-bit ALU Project Report

1. Introduction

This project presents an 8-bit Arithmetic Logic Unit (ALU) implemented in Verilog. The ALU performs a variety of arithmetic, logical, and bitwise operations using a 4-bit opcode, with dedicated output flags to indicate carry, overflow, zero, and sign conditions.

2. ALU Operations

Supported operations include:

- ADD (A + B)
- SUB (A - B)
- AND, OR, XOR, NOT
- SHL (Shift Left), SHR (Shift Right)
- ROL (Rotate Left), ROR (Rotate Right)
- INC (A + 1), DEC (A - 1)

Each operation is selected using a 4-bit opcode.

3. Verilog Design Overview

The ALU is implemented in a single module (`alu.v`) with combinational logic. The testbench (`alu_tb.v`) validates all operations and generates a waveform file using `$dumpfile` and `$dumpvars`. Flags (C, Z, V, S) are updated accordingly.

4. Simulation

Simulation is done using Icarus Verilog and GTKWave. The testbench exercises key cases like overflow in ADD/SUB and verifies output flags. Output waveforms help in debugging and verification.

5. Conclusion

This ALU project demonstrates fundamental digital design principles including conditional logic, flag generation, and behavioral modeling. It serves as a solid example of a modular and testable digital subsystem.