ALU Design and Verification Using SystemVerilog

`Submitted by:` Amr Hossam

Overview

This document provides an overview of the design and verification of an Arithmetic Logic Unit (ALU) using SystemVerilog. The verification process includes both manual and randomized testing approaches. The results from these tests are compared against a golden reference.

Design Description

The ALU design is implemented in SystemVerilog and includes the following features:

- **Inputs:**
  - `A`: 4-bit operand
  - `B`: 4-bit operand
  - `ALU_FUN`: 2-bit function select signal
  - `CLK`: Clock signal
- **Output:**
  - `ALU_OUT`: 8-bit result of the ALU operation

The ALU supports four operations based on the `ALU_FUN` input:

- `2'b00`: Addition (`ALU_OUT = A + B`)
- `2'b01`: Subtraction (`ALU_OUT = A - B`)
- `2'b10`: Multiplication (`ALU_OUT = A * B`)
- `2'b11`: Division (`ALU_OUT = A / B`)

Verification Methodology

1. Manual Tests

Manual tests were conducted by entering predefined inputs into the ALU and observing the output. The results were then compared to the expected values to validate correctness.

Verification Plan and Results:

- **Manual Test Outputs:**

```
# Choosen inputs in verification plan
# Test Passed: A = 0, B = 0, ALU_FUN = 0, Expected = 0, Actual = 0
# Test Passed: A = 1, B = 2, ALU_FUN = 0, Expected = 3, Actual = 3
# Test Passed: A = 15, B = 1, ALU_FUN = 0, Expected = 16, Actual = 16
# Test Passed: A = 4, B = 3, ALU_FUN = 1, Expected = 1, Actual = 1
# Test Passed: A = 1, B = 4, ALU_FUN = 1, Expected = 253, Actual = 253
# Test Passed: A = 2, B = 3, ALU_FUN = 2, Expected = 6, Actual = 6
# Test Passed: A = 15, B = 2, ALU_FUN = 2, Expected = 30, Actual = 30
# Test Passed: A = 4, B = 2, ALU_FUN = 3, Expected = 2, Actual = 2
# Test Passed: A = 15, B = 1, ALU_FUN = 3, Expected = 15, Actual = 15
# Test Passed: A = 15, B = 15, ALU_FUN = 0, Expected = 30, Actual = 30
# Test Passed: A = 15, B = 15, ALU_FUN = 1, Expected = 0, Actual = 0
# Test Passed: A = 15, B = 15, ALU_FUN = 2, Expected = 225, Actual = 225
# Test Passed: A = 15, B = 15, ALU_FUN = 3, Expected = 1, Actual = 1
# Test Passed: A = 0, B = 15, ALU_FUN = 0, Expected = 15, Actual = 15
# Test Passed: A = 0, B = 15, ALU_FUN = 1, Expected = 241, Actual = 241
# Test Passed: A = 0, B = 15, ALU_FUN = 2, Expected = 0, Actual = 0
```

2. Randomized Inputs

Randomized tests were performed using a generator to produce a variety of input values for the ALU.

Randomized Test Outputs:

- **Randomized Test Results:**

```
# Random inputs
# Test Passed: A = 0, B = 15, ALU_FUN = 3, Expected = 0, Actual = 0
# Test Passed: A = 10, B = 14, ALU_FUN = 1, Expected = 252, Actual = 252
# Test Passed: A = 9, B = 10, ALU_FUN = 3, Expected = 0, Actual = 0
# Test Passed: A = 14, B = 4, ALU_FUN = 2, Expected = 56, Actual = 56
# Test Passed: A = 7, B = 8, ALU_FUN = 2, Expected = 56, Actual = 56
# Test Passed: A = 4, B = 1, ALU_FUN = 0, Expected = 5, Actual = 5
# Test Passed: A = 11, B = 12, ALU_FUN = 2, Expected = 132, Actual = 132
# Test Passed: A = 2, B = 4, ALU_FUN = 1, Expected = 254, Actual = 254
# Test Passed: A = 15, B = 14, ALU_FUN = 0, Expected = 29, Actual = 29
# Test Passed: A = 4, B = 11, ALU_FUN = 2, Expected = 44, Actual = 44
# Test Passed: A = 11, B = 2, ALU_FUN = 2, Expected = 22, Actual = 22
# Test Passed: A = 6, B = 11, ALU_FUN = 1, Expected = 251, Actual = 251
# Test Passed: A = 1, B = 13, ALU_FUN = 3, Expected = 0, Actual = 0
# Test Passed: A = 6, B = 5, ALU_FUN = 3, Expected = 1, Actual = 1
# Test Passed: A = 8, B = 5, ALU_FUN = 0, Expected = 13, Actual = 13
# Test Passed: A = 0, B = 1, ALU_FUN = 2, Expected = 0, Actual = 0
# Test Passed: A = 11, B = 10, ALU_FUN = 1, Expected = 1, Actual = 1
# Test Passed: A = 8, B = 14, ALU_FUN = 0, Expected = 22, Actual = 22
# Test Passed: A = 14, B = 8, ALU_FUN = 1, Expected = 6, Actual = 6
# Test Passed: A = 2, B = 11, ALU_FUN = 2, Expected = 22, Actual = 22
# T=                 730 [Generator] Done generation of          20 items
```

Comparison with Golden Reference

The outputs from both manual and randomized tests were compared to a golden reference. This comparison helps ensure that the ALU operates as expected and meets the design specifications.