32-bit ALU Design and Verification Report

Author: Noridel Herron

Date: 08/14/2025

1. Overview

The **Arithmetic Logic Unit** (ALU) is a combinational logic block that performs arithmetic and logical operations as part of the CPU datapath. This ALU accepts two 32-bit operands (A and B), along with a 4-bit operation code (alu_op), and produces a 32-bit result. It also generates four status flags: Z (Zero), N (Negative), C (Carry), and V (Overflow).

2. Features

Feature	Description
Operand Width	32 bits
Operations Supported	ADD, SUB, XOR, OR, AND, SLL, SRL, SRA, SLT, SLTU
Status Flags	Z, N, C, V
Overflow detection	For both addition and subtraction using signed arithmetic rules
Operation Codes	Parameterizable via localparam constants

3. I/O Ports

Signal	Direction	Width	Description
A	Input	32	Operand A
B	Input	32	Operand B
alu_op	Input	4	Operation selector
result	Output	32	ALU result

Z	Output	1	Zero flag
N	Output	1	Negative flag
C	Output	1	Carry flag
V	Output	1	Overflow flag

4. Operation Codes

Name	Value (hex)	Description
ALU_ADD	0x0	Addition
ALU_SUB	0x1	Subtraction
ALU_XOR	0x2	Bitwise XOR
ALU_OR	0x3	Bitwise OR
ALU_AND	0x4	Bitwise AND
ALU_SLL	0x5	Shift left logical
ALU_SRL	0x6	Shift right logical
ALU_SRA	0x7	Shift right arithmetic
ALU_SLT	0x8	Set less than (signed)
ALU_SLTU	0x9	Set less than (unsigned)

5. Design Details

The ALU uses an always_comb block with a case statement on alu_op.

Each case computes the result and flags accordingly:

- C (Carry):

  For ADD, taken from the MSB carry-out

  (C = temp[32] where temp = {1'b0, A} + {1'b0, B}).

  For SUB, represents 'no borrow' and is computed as the inverted MSB borrow bit

  (C = ~temp[32] where temp = {1'b0, A} - {1'b0, B}).

- V (Overflow):

  For ADD, set when operands have the same sign but the result's sign differs from A

  (V = (A[31] == B[31]) && (result[31] != A[31])).

  For SUB, set when operands have different signs but the result's sign differs from A

  (V = (A[31] != B[31]) && (result[31] != A[31])).

- Z (Zero): High when result equals zero.
- N (Negative): Reflects the sign bit of result (N = result[31]).

6. Verification Strategy

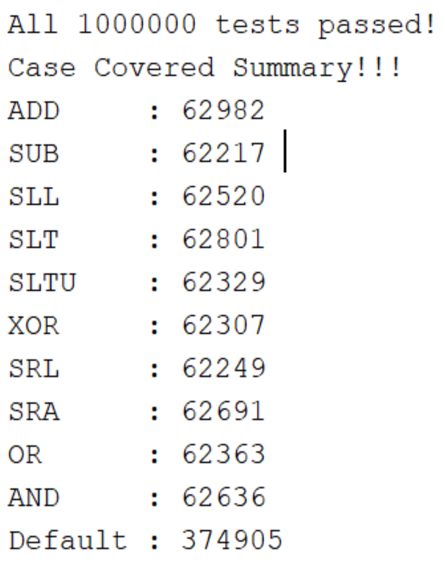
A SystemVerilog testbench was developed with the following features:
• Randomized Testing: 1,000,000 iterations with random 32-bit inputs and random operation codes.
• Expected Model: A variable tracks the expected result for each operation.
• Scoreboard-like Comparison: Actual DUT outputs compared against expected outputs every iteration.
• Debug Variable: An additional variable tracks the current operation index for pinpointing bugs quickly.
• Bug Narrowing: When a mismatch occurs, the debug variable helps identify the exact iteration and input set that caused it.

7. Future Improvements

• Parameterize operand width for more flexibility.
• Add multiply and divide operations.
• Implement a formal verification suite in addition to random simulation.
• Integrate with a CPU pipeline and test with instruction-level workloads.

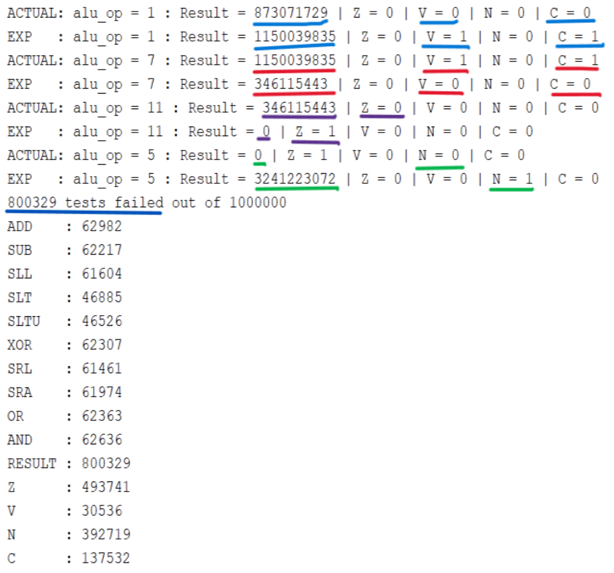
8. Simulation Output

In the TCL console, the testbench is structured so that if no errors occur, a case coverage summary is displayed (Figure 1A). If errors are detected, a different report is generated to pinpoint the exact issue (Figure 1B).



```
All 1000000 tests passed!
Case Covered Summary!!!
ADD     : 62982
SUB     : 62217
SLL     : 62520
SLT     : 62801
SLTU    : 62329
XOR     : 62307
SRL     : 62249
SRA     : 62691
OR      : 62363
AND     : 62636
Default : 374905
```

Figure 1A



```
ACTUAL: alu_op = 1 : Result = 873071729 | Z = 0 | V = 0 | N = 0 | C = 0
EXP   : alu_op = 1 : Result = 1150039835 | Z = 0 | V = 1 | N = 0 | C = 1
ACTUAL: alu_op = 7 : Result = 1150039835 | Z = 0 | V = 1 | N = 0 | C = 1
EXP   : alu_op = 7 : Result = 346115443 | Z = 0 | V = 0 | N = 0 | C = 0
ACTUAL: alu_op = 11 : Result = 346115443 | Z = 0 | V = 0 | N = 0 | C = 0
EXP   : alu_op = 11 : Result = 0 | Z = 1 | V = 0 | N = 0 | C = 0
ACTUAL: alu_op = 5 : Result = 0 | Z = 1 | V = 0 | N = 0 | C = 0
EXP   : alu_op = 5 : Result = 3241223072 | Z = 0 | V = 0 | N = 1 | C = 0
800329 tests failed out of 1000000
ADD    : 62982
SUB    : 62217
SLL    : 61604
SLT    : 46885
SLTU   : 46526
XOR    : 62307
SRL    : 61461
SRA    : 61974
OR     : 62363
AND    : 62636
RESULT : 800329
Z      : 493741
V      : 30536
N      : 392719
C      : 137532
```

Figure 1B

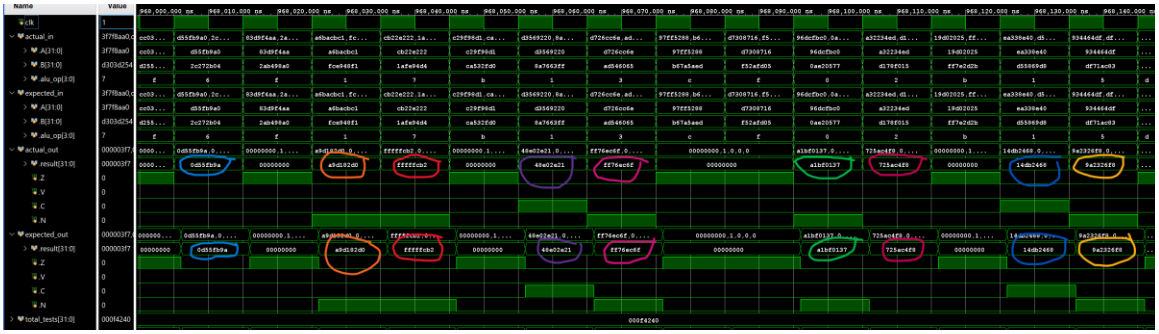
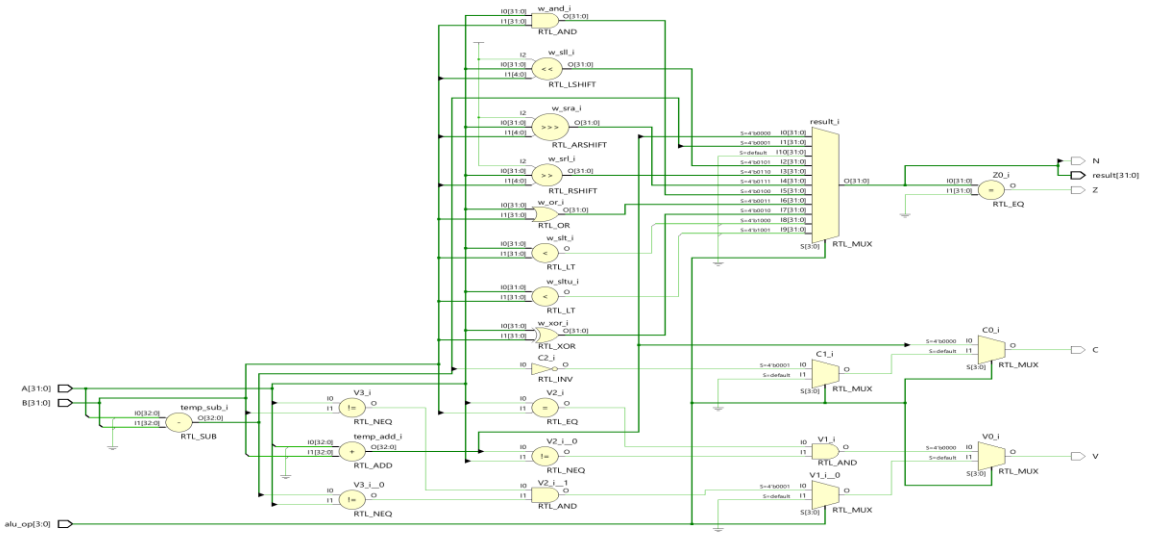


Figure 1C



9. Code File Names

The design and verification sources are organized as follows:

- Module: Verilog Folder -> v_alu.v
- Testbench: test_benches Folder -> alu_tb.sv