

Verification Plan for Sequential ALU

1. Inputs and Outputs

- **Inputs:**
 - **clk**: Clock signal for sequential operation.
 - **rst**: Reset signal to initialize the ALU.
 - **opcode**: Control signal to select the operation (e.g., addition, subtraction).
 - **A, B**: Operands for the ALU.
- **Outputs:**
 - **Result**: Output result of the ALU operation.

2. Test Plan

2.1 Test Scenarios

1. Reset Condition:

- Ensure the ALU resets to 0 when the reset signal is high.

2. Addition Operation:

- Verify the correctness of the addition operation ($A + B$).
- Test edge cases, such as when **A** or **B** are 0, FF, and overflow conditions.

3. Subtraction Operation:

- Test $A - B$ for different operand values.
- Check underflow conditions (i.e., $A < B$).

4. AND Operation:

- Test $A \& B$ for different values of **A** and **B**.

5. OR Operation:

- Test $A \mid B$ for different values of **A** and **B**.

6. XOR Operation:

- Test $A \wedge B$ for different operand values.

7. Sequential Execution:

- Verify the ALU executes operations sequentially in each clock cycle.

2.2 Corner Cases

- Test when **A** and **B** are both zero.

- Test operations with maximum and minimum possible values for operands.
- Verify behavior when an undefined opcode is provided.

3. Pass/Fail Criteria

The test will be considered **pass** if:

- The ALU performs all operations correctly as per the opcode.
- Functional coverage shows 100% for all scenarios.
- There are no assertion failures during the simulation.

The test will be considered **fail** if any of the following conditions occur:

- Incorrect ALU results for valid operations.
- The ALU does not reset correctly or fails to respond to the clock signal.