**Simulation Report for Basic ALU Design in Verilog**

Name: Md. Sameer Ahmed

Internship Domain: VLSI

Task: Basic ALU Design using Verilog

**Objective**

The objective of this task is to design a basic ALU (Arithmetic Logic Unit) that supports operations such as addition, subtraction, AND, OR, and NOT using Verilog HDL, and verify its functionality through simulation.

**Tools Used**

- Language: Verilog HDL

- Simulator: Icarus Verilog

- Waveform Viewer: GTKWave

**Design Description**

The ALU module takes two 4-bit inputs A and B, a 3-bit control input ALU\_Sel, and produces a 4-bit output ALU\_Out. Based on the value of ALU\_Sel, the ALU performs one of the following operations:

ALU\_Sel (Binary) Operation Description

000 A + B Addition

001 A - B Subtraction

010 A & B Bitwise AND

011 A | B Bitwise OR

100 ~A Bitwise NOT of A

**ALU Module Code (Verilog)**

module ALU (

input [3:0] A, // 4-bit input operand A

input [3:0] B, // 4-bit input operand B

input [2:0] ALU\_Sel, // 3-bit control signal to select ALU operation

output reg [3:0] ALU\_Out // 4-bit output from ALU (registered output)

);

always @(\*) begin

case (ALU\_Sel)

3'b000: ALU\_Out = A + B; // Addition

3'b001: ALU\_Out = A - B; // Subtraction

3'b010: ALU\_Out = A & B; // Bitwise AND

3'b011: ALU\_Out = A | B; // Bitwise OR

3'b100: ALU\_Out = ~A; // Bitwise NOT of A

default: ALU\_Out = 4'b0000; // Default output

endcase

end

endmodule

**Testbench Code (Verilog)**

module tb\_ALU;

reg [3:0] A, B;

reg [2:0] ALU\_Sel;

wire [3:0] ALU\_Out;

ALU uut (

.A(A),

.B(B),

.ALU\_Sel(ALU\_Sel),

.ALU\_Out(ALU\_Out)

);

initial begin

$display("A B ALU\_Sel Result");

$monitor("%b %b %b %b", A, B, ALU\_Sel, ALU\_Out);

A = 4'b0101; B = 4'b0011; ALU\_Sel = 3'b000; #10; // Addition

A = 4'b0101; B = 4'b0011; ALU\_Sel = 3'b001; #10; // Subtraction

A = 4'b0101; B = 4'b0011; ALU\_Sel = 3'b010; #10; // AND

A = 4'b0101; B = 4'b0011; ALU\_Sel = 3'b011; #10; // OR

A = 4'b0101; B = 4'b0011; ALU\_Sel = 3'b100; #10; // NOT

$finish;

end

endmodule

**Simulation Output**

A B ALU\_Sel Result

0101 0011 000 1000 // 5 + 3 = 8

0101 0011 001 0010 // 5 - 3 = 2

0101 0011 010 0001 // 5 & 3 = 1

0101 0011 011 0111 // 5 | 3 = 7

0101 0011 100 1010 // ~5 = 10 (in 2’s complement)

**Conclusion**

The Verilog-based ALU was successfully implemented and tested. Simulation results confirmed that the ALU correctly performs all five operations: addition, subtraction, AND, OR, and NOT. This validates the functional correctness of the ALU design.