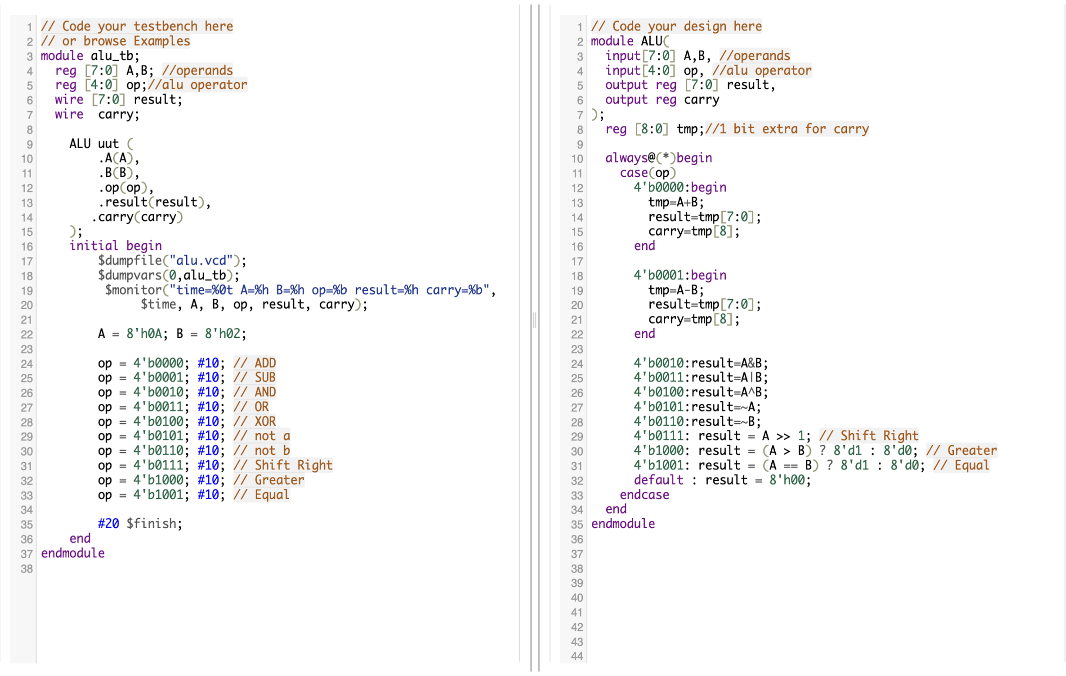
ALU Project

Name: Shreyash Jaiswal
Topic: Arithmetic Logic Unit (ALU) Design and Simulation

Generated using Verilog HDL on EDA Playground

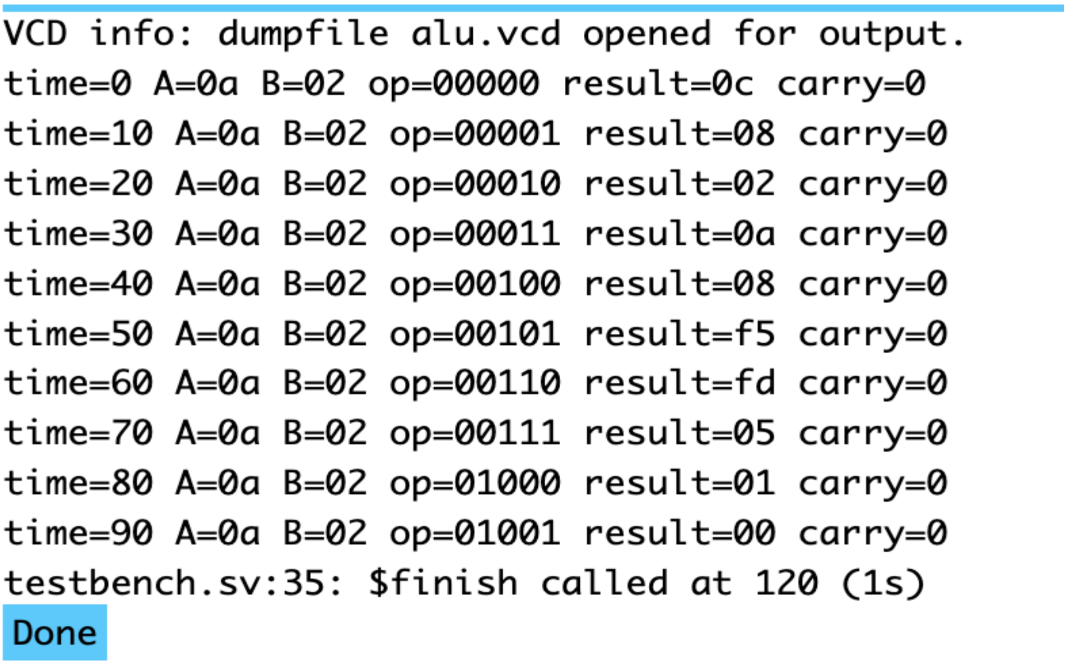
Project Description

This project focuses on the design and implementation of an 8-bit Arithmetic Logic Unit (ALU) using Verilog HDL. The ALU performs various arithmetic and logical operations such as addition, subtraction, AND, OR, XOR, NOT, shift operations, and comparison. The design was simulated on EDA Playground, and the outputs were verified using waveform analysis in EPWave. The testbench applies different inputs and verifies the correctness of the ALU design. The project demonstrates understanding of fundamental digital logic design, Verilog coding, and simulation-based verification.

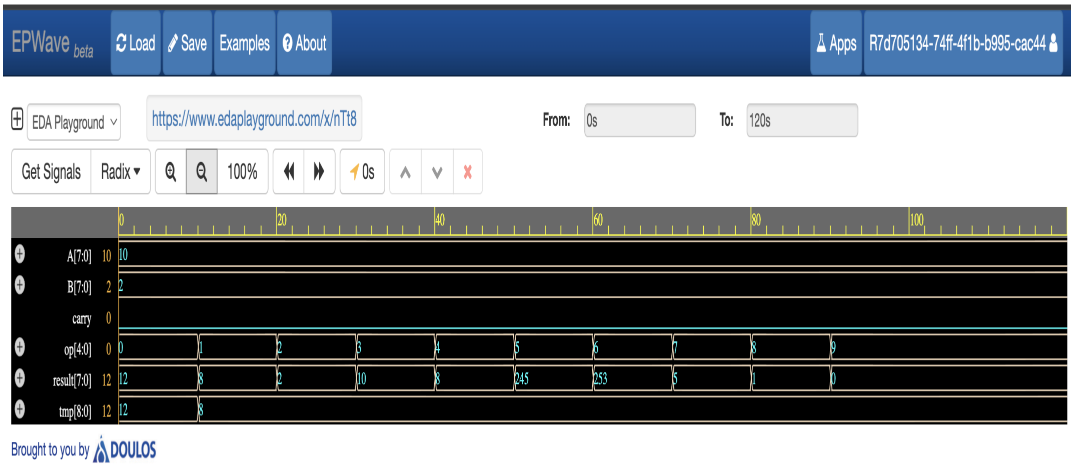


```verilog
// Code your testbench here
// or browse Examples
module alu_tb;
  reg [7:0] A,B; //operands
  reg [4:0] op;//alu operator
  wire [7:0] result;
  wire  carry;

    ALU uut (
        .A(A),
        .B(B),
        .op(op),
        .result(result),
       .carry(carry)
    );
    initial begin
        $dumpfile("alu.vcd");
        $dumpvars(0,alu_tb);
         $monitor("time=%0t A=%h B=%h op=%b result=%h carry=%b",
              $time, A, B, op, result, carry);

        A = 8'h0A; B = 8'h02;

        op = 4'b0000; #10; // ADD
        op = 4'b0001; #10; // SUB
        op = 4'b0010; #10; // AND
        op = 4'b0011; #10; // OR
        op = 4'b0100; #10; // XOR
        op = 4'b0101; #10; // not a
        op = 4'b0110; #10; // not b
        op = 4'b0111; #10; // Shift Right
        op = 4'b1000; #10; // Greater
        op = 4'b1001; #10; // Equal

        #20 $finish;
    end
endmodule
```

```verilog
// Code your design here
module ALU(
  input[7:0] A,B, //operands
  input[4:0] op, //alu operator
  output reg [7:0] result,
  output reg carry
);
  reg [8:0] tmp;//1 bit extra for carry

  always@(*)begin
    case(op)
      4'b0000:begin
        tmp=A+B;
        result=tmp[7:0];
        carry=tmp[8];
      end

      4'b0001:begin
        tmp=A-B;
        result=tmp[7:0];
        carry=tmp[8];
      end

      4'b0010:result=A&B;
      4'b0011:result=A|B;
      4'b0100:result=A^B;
      4'b0101:result=~A;
      4'b0110:result=~B;
      4'b0111: result = A >> 1; // Shift Right
      4'b1000: result = (A > B) ? 8'd1 : 8'd0; // Greater
      4'b1001: result = (A == B) ? 8'd1 : 8'd0; // Equal
      default : result = 8'h00;
    endcase
  end
endmodule
```



```
VCD info: dumpfile alu.vcd opened for output.
time=0 A=0a B=02 op=00000 result=0c carry=0
time=10 A=0a B=02 op=00001 result=08 carry=0
time=20 A=0a B=02 op=00010 result=02 carry=0
time=30 A=0a B=02 op=00011 result=0a carry=0
time=40 A=0a B=02 op=00100 result=08 carry=0
time=50 A=0a B=02 op=00101 result=f5 carry=0
time=60 A=0a B=02 op=00110 result=fd carry=0
time=70 A=0a B=02 op=00111 result=05 carry=0
time=80 A=0a B=02 op=01000 result=01 carry=0
time=90 A=0a B=02 op=01001 result=00 carry=0
testbench.sv:35: $finish called at 120 (1s)
Done
```



https://www.edaplayground.com/x/nTt8

From: 0s

To: 120s