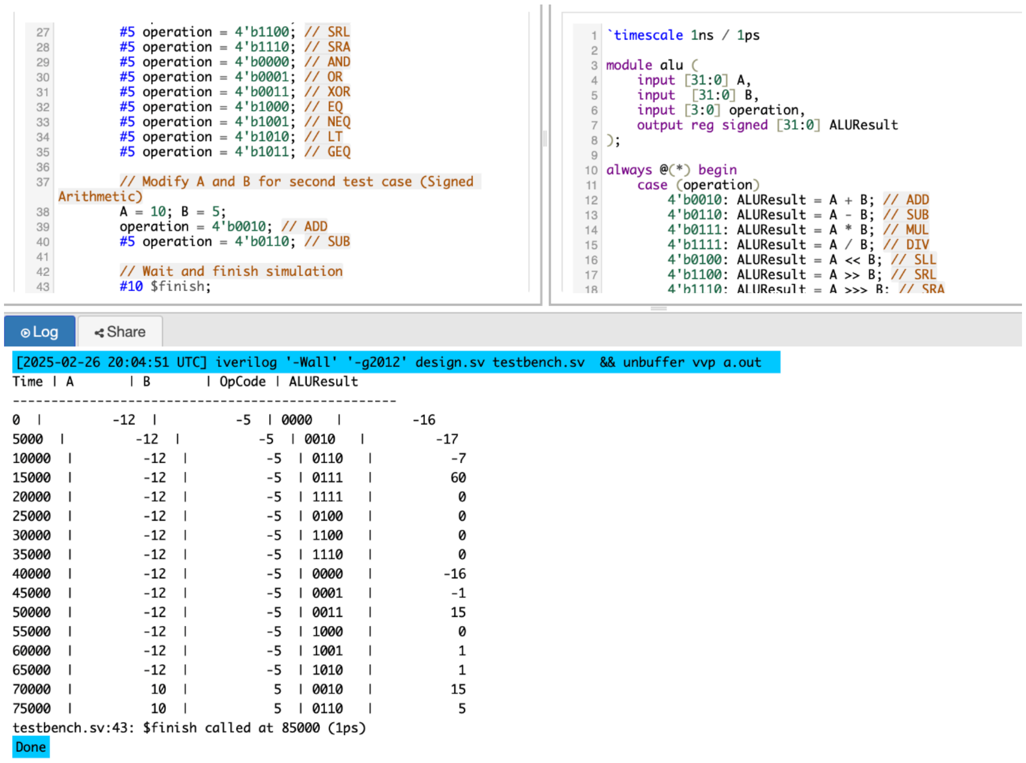
Task 1

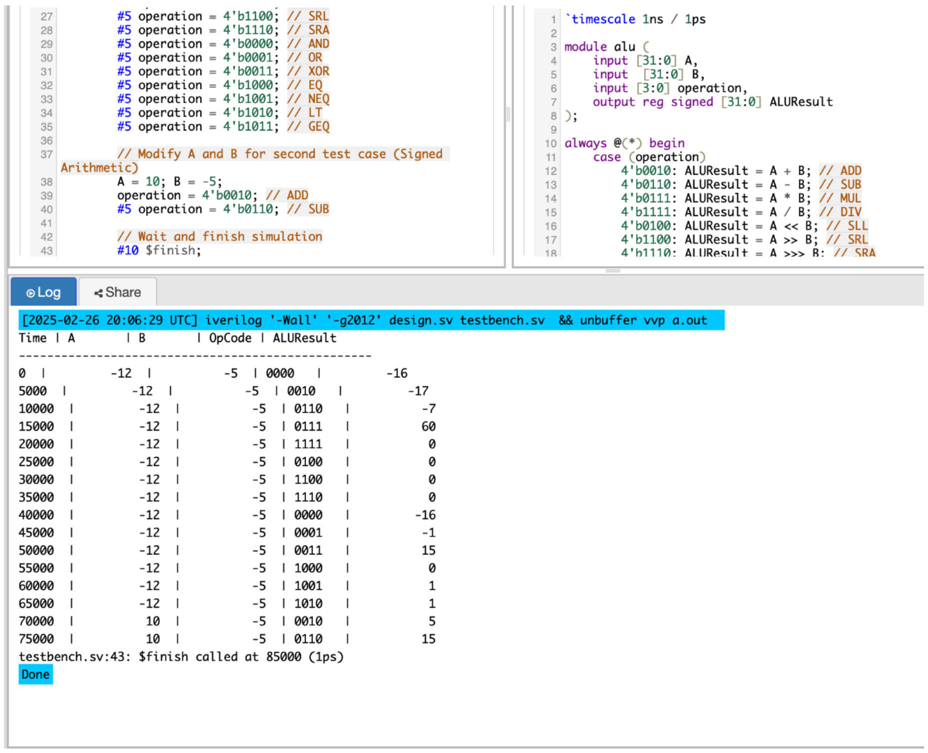


```
        #5 operation = 4'b1100; // SRL
        #5 operation = 4'b1110; // SRA
        #5 operation = 4'b0000; // AND
        #5 operation = 4'b0001; // OR
        #5 operation = 4'b0011; // XOR
        #5 operation = 4'b1000; // EQ
        #5 operation = 4'b1001; // NEQ
        #5 operation = 4'b1010; // LT
        #5 operation = 4'b1011; // GEQ

        // Modify A and B for second test case (Signed
Arithmetic)
        A = 10; B = 5;
        operation = 4'b0010; // ADD
        #5 operation = 4'b0110; // SUB

        // Wait and finish simulation
        #10 $finish;
```

```
`timescale 1ns / 1ps

module alu (
    input [31:0] A,
    input  [31:0] B,
    input [3:0] operation,
    output reg signed [31:0] ALUResult
);

always @(*) begin
    case (operation)
        4'b0010: ALUResult = A + B; // ADD
        4'b0110: ALUResult = A - B; // SUB
        4'b0111: ALUResult = A * B; // MUL
        4'b1111: ALUResult = A / B; // DIV
        4'b0100: ALUResult = A << B; // SLL
        4'b1100: ALUResult = A >> B; // SRL
        4'b1110: ALUResult = A >>> B; // SRA
```

```
[2025-02-26 20:04:51 UTC] iverilog '-Wall' '-g2012' design.sv testbench.sv  && unbuffer vvp a.out
Time | A        | B        | OpCode | ALUResult
--------------------------------------------------
0  |        -12  |          -5  | 0000   |          -16
5000  |        -12  |          -5  | 0010   |          -17
10000  |        -12  |          -5  | 0110   |           -7
15000  |        -12  |          -5  | 0111   |           60
20000  |        -12  |          -5  | 1111   |            0
25000  |        -12  |          -5  | 0100   |            0
30000  |        -12  |          -5  | 1100   |            0
35000  |        -12  |          -5  | 1110   |            0
40000  |        -12  |          -5  | 0000   |          -16
45000  |        -12  |          -5  | 0001   |           -1
50000  |        -12  |          -5  | 0011   |           15
55000  |        -12  |          -5  | 1000   |            0
60000  |        -12  |          -5  | 1001   |            1
65000  |        -12  |          -5  | 1010   |            1
70000  |         10  |           5  | 0010   |           15
75000  |         10  |           5  | 0110   |            5
testbench.sv:43: $finish called at 85000 (1ps)
Done
```

Task 2

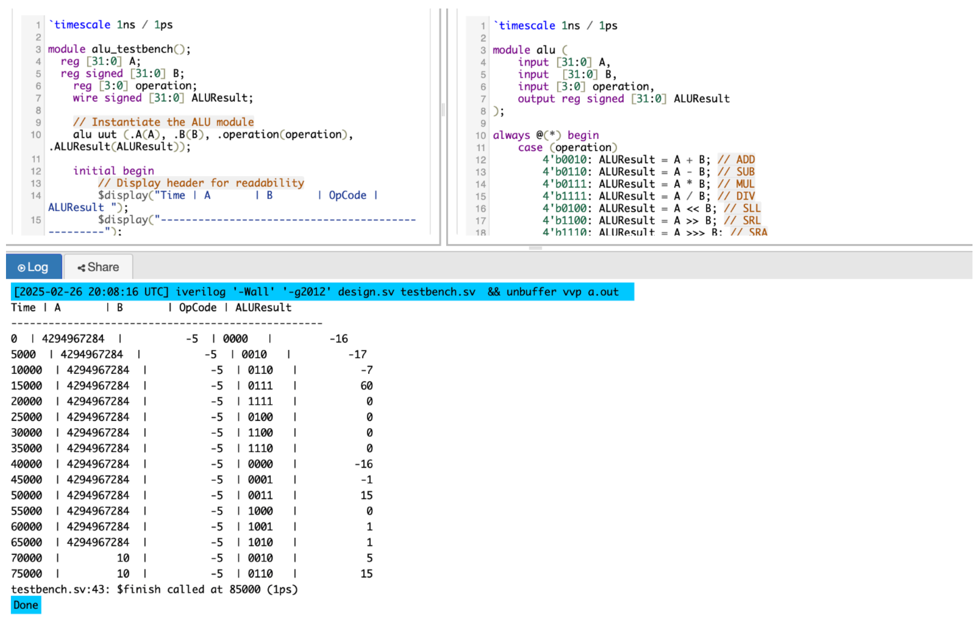


```
        #5 operation = 4'b1100; // SRL
        #5 operation = 4'b1110; // SRA
        #5 operation = 4'b0000; // AND
        #5 operation = 4'b0001; // OR
        #5 operation = 4'b0011; // XOR
        #5 operation = 4'b1000; // EQ
        #5 operation = 4'b1001; // NEQ
        #5 operation = 4'b1010; // LT
        #5 operation = 4'b1011; // GEQ

        // Modify A and B for second test case (Signed
Arithmetic)
        A = 10; B = -5;
        operation = 4'b0010; // ADD
        #5 operation = 4'b0110; // SUB

        // Wait and finish simulation
        #10 $finish;
```

```
`timescale 1ns / 1ps

module alu (
    input [31:0] A,
    input  [31:0] B,
    input [3:0] operation,
    output reg signed [31:0] ALUResult
);

always @(*) begin
    case (operation)
        4'b0010: ALUResult = A + B; // ADD
        4'b0110: ALUResult = A - B; // SUB
        4'b0111: ALUResult = A * B; // MUL
        4'b1111: ALUResult = A / B; // DIV
        4'b0100: ALUResult = A << B; // SLL
        4'b1100: ALUResult = A >> B; // SRL
        4'b1110: ALUResult = A >>> B; // SRA
```

```
[2025-02-26 20:06:29 UTC] iverilog '-Wall' '-g2012' design.sv testbench.sv  && unbuffer vvp a.out
Time | A        | B        | OpCode | ALUResult
--------------------------------------------------
0  |        -12  |          -5  | 0000   |          -16
5000  |        -12  |          -5  | 0010   |          -17
10000  |        -12  |          -5  | 0110   |           -7
15000  |        -12  |          -5  | 0111   |           60
20000  |        -12  |          -5  | 1111   |            0
25000  |        -12  |          -5  | 0100   |            0
30000  |        -12  |          -5  | 1100   |            0
35000  |        -12  |          -5  | 1110   |            0
40000  |        -12  |          -5  | 0000   |          -16
45000  |        -12  |          -5  | 0001   |           -1
50000  |        -12  |          -5  | 0011   |           15
55000  |        -12  |          -5  | 1000   |            0
60000  |        -12  |          -5  | 1001   |            1
65000  |        -12  |          -5  | 1010   |            1
70000  |         10  |          -5  | 0010   |            5
75000  |         10  |          -5  | 0110   |           15
testbench.sv:43: $finish called at 85000 (1ps)
Done
```

Task 3



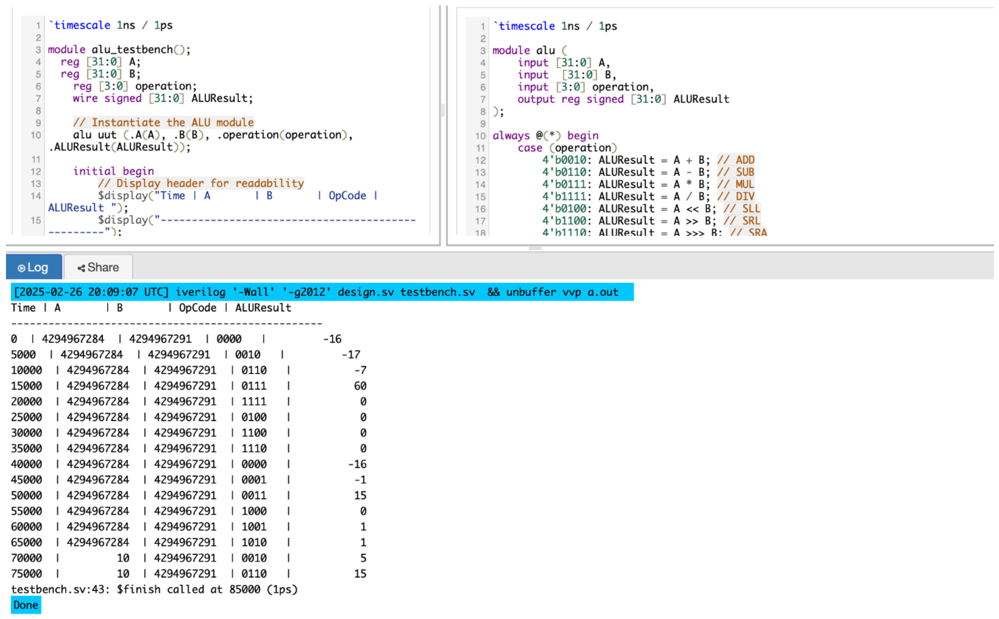
```
`timescale 1ns / 1ps

module alu_testbench();
  reg [31:0] A;
  reg signed [31:0] B;
    reg [3:0] operation;
    wire signed [31:0] ALUResult;

    // Instantiate the ALU module
    alu uut (.A(A), .B(B), .operation(operation),
.ALUResult(ALUResult));

    initial begin
        // Display header for readability
        $display("Time | A        | B        | OpCode |
ALUResult ");
        $display("----------------------------------------
---------");
```

```
`timescale 1ns / 1ps

module alu (
    input [31:0] A,
    input  [31:0] B,
    input [3:0] operation,
    output reg signed [31:0] ALUResult
);

always @(*) begin
    case (operation)
        4'b0010: ALUResult = A + B; // ADD
        4'b0110: ALUResult = A - B; // SUB
        4'b0111: ALUResult = A * B; // MUL
        4'b1111: ALUResult = A / B; // DIV
        4'b0100: ALUResult = A << B; // SLL
        4'b1100: ALUResult = A >> B; // SRL
        4'b1110: ALUResult = A >>> B; // SRA
```

Log   Share

```
[2025-02-26 20:08:16 UTC] iverilog '-Wall' '-g2012' design.sv testbench.sv  && unbuffer vvp a.out
Time | A        | B        | OpCode | ALUResult
--------------------------------------------------
0  | 4294967284  |          -5  | 0000   |          -16
5000  | 4294967284  |          -5  | 0010   |          -17
10000  | 4294967284  |          -5  | 0110   |           -7
15000  | 4294967284  |          -5  | 0111   |           60
20000  | 4294967284  |          -5  | 1111   |            0
25000  | 4294967284  |          -5  | 0100   |            0
30000  | 4294967284  |          -5  | 1100   |            0
35000  | 4294967284  |          -5  | 1110   |            0
40000  | 4294967284  |          -5  | 0000   |          -16
45000  | 4294967284  |          -5  | 0001   |           -1
50000  | 4294967284  |          -5  | 0011   |           15
55000  | 4294967284  |          -5  | 1000   |            0
60000  | 4294967284  |          -5  | 1001   |            1
65000  | 4294967284  |          -5  | 1010   |            1
70000  |          10  |          -5  | 0010   |            5
75000  |          10  |          -5  | 0110   |           15
testbench.sv:43: $finish called at 85000 (1ps)
Done
```

Task 4



```
`timescale 1ns / 1ps

module alu_testbench();
  reg [31:0] A;
  reg [31:0] B;
    reg [3:0] operation;
    wire signed [31:0] ALUResult;

    // Instantiate the ALU module
    alu uut (.A(A), .B(B), .operation(operation),
.ALUResult(ALUResult));

    initial begin
        // Display header for readability
        $display("Time | A        | B        | OpCode |
ALUResult ");
        $display("----------------------------------------
---------");
```

```
`timescale 1ns / 1ps

module alu (
    input [31:0] A,
    input  [31:0] B,
    input [3:0] operation,
    output reg signed [31:0] ALUResult
);

always @(*) begin
    case (operation)
        4'b0010: ALUResult = A + B; // ADD
        4'b0110: ALUResult = A - B; // SUB
        4'b0111: ALUResult = A * B; // MUL
        4'b1111: ALUResult = A / B; // DIV
        4'b0100: ALUResult = A << B; // SLL
        4'b1100: ALUResult = A >> B; // SRL
        4'b1110: ALUResult = A >>> B; // SRA
```

Log   Share

```
[2025-02-26 20:09:07 UTC] iverilog '-Wall' '-g2012' design.sv testbench.sv  && unbuffer vvp a.out
Time | A        | B        | OpCode | ALUResult
--------------------------------------------------
0  | 4294967284  | 4294967291  | 0000   |          -16
5000  | 4294967284  | 4294967291  | 0010   |          -17
10000  | 4294967284  | 4294967291  | 0110   |           -7
15000  | 4294967284  | 4294967291  | 0111   |           60
20000  | 4294967284  | 4294967291  | 1111   |            0
25000  | 4294967284  | 4294967291  | 0100   |            0
30000  | 4294967284  | 4294967291  | 1100   |            0
35000  | 4294967284  | 4294967291  | 1110   |            0
40000  | 4294967284  | 4294967291  | 0000   |          -16
45000  | 4294967284  | 4294967291  | 0001   |           -1
50000  | 4294967284  | 4294967291  | 0011   |           15
55000  | 4294967284  | 4294967291  | 1000   |            0
60000  | 4294967284  | 4294967291  | 1001   |            1
65000  | 4294967284  | 4294967291  | 1010   |            1
70000  |          10  | 4294967291  | 0010   |            5
75000  |          10  | 4294967291  | 0110   |           15
testbench.sv:43: $finish called at 85000 (1ps)
Done
```