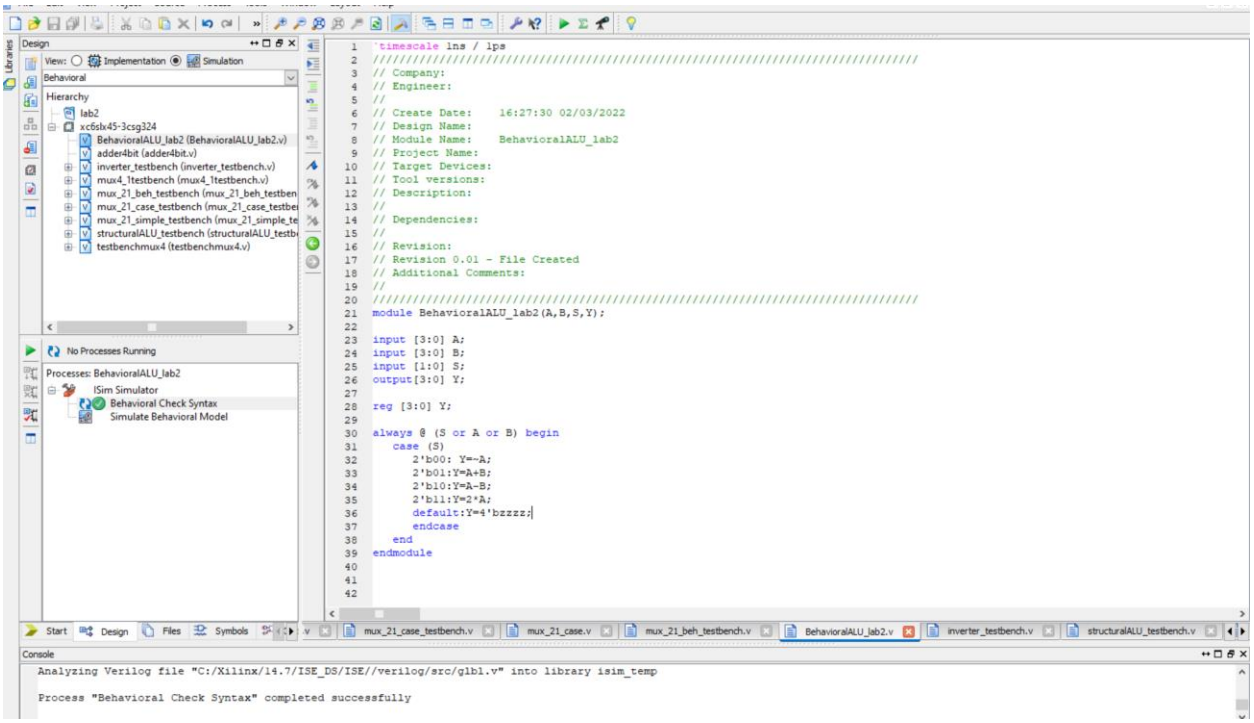
CDA 4203L
Computer System Design Lab

Lab 2 Report
Verilog Based ALU Design

Today's Date:	2/5/2022
Your Name:	Ahmed El Maliki
Your U Number:	U 18019330
No. of Hours Spent:	10H
Exercise Difficulty: (Easy, Average, Hard)	Average
Any Other Feedback:	Instructions were not that clear about the mux for part 2. It did not specify the restriction on input bits and output bits for mux to be used. To avoid any misunderstanding, I used the default 2-bit input 1 select line bit and 1 bit output to create my mux network. This is what resulted in a lot of time being spent on this project.

Problem 1: ALU Behavioral Verilog



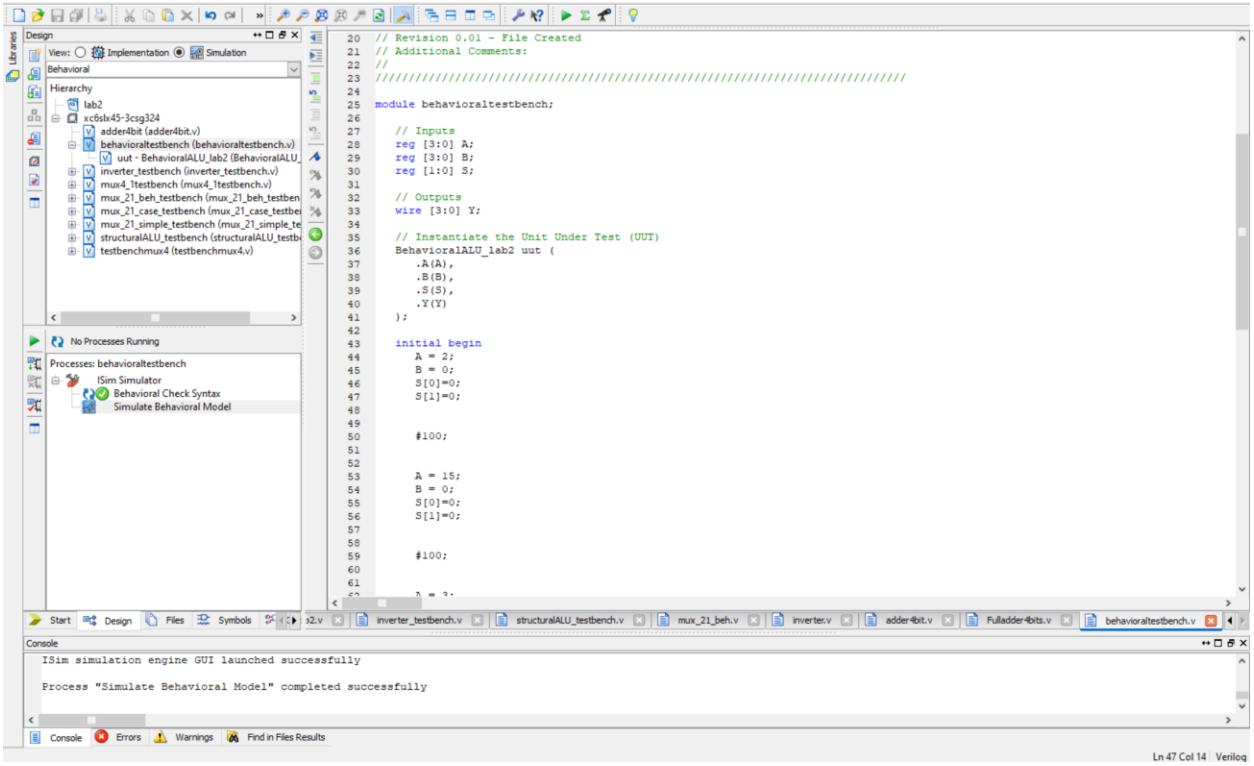

```
1  `timescale 1ns / 1ps
2  //////////////////////////////////////////////////////////////////////////////////
3  // Company:
4  // Engineer:
5  //
6  // Create Date:    16:27:30 02/03/2022
7  // Design Name:
8  // Module Name:    BehavioralALU_lab2
9  // Project Name:
10 // Target Devices:
11 // Tool versions:
12 // Description:
13 //
14 // Dependencies:
15 //
16 // Revision:
17 // Revision 0.01 - File Created
18 // Additional Comments:
19 //
20 //////////////////////////////////////////////////////////////////////////////////
21 module BehavioralALU_lab2(A,B,S,Y);
22
23 input [3:0] A;
24 input [3:0] B;
25 input [1:0] S;
26 output[3:0] Y;
27
28 reg [3:0] Y;
29
30 always @ (S or A or B) begin
31    case (S)
32       2'b00: Y=~A;
33       2'b01:Y=A+B;
34       2'b10:Y=A-B;
35       2'b11:Y=2*A;
36       default:Y=4'bzzzz;
37       endcase
38    end
39 endmodule
40
41
42
```

```
module BehavioralALU_lab2(A,B,S,Y);

input [3:0] A;
input [3:0] B;
input [1:0] S;
output[3:0] Y;

reg [3:0] Y;

always @ (S or A or B) begin
        case (S)
                2'b00: Y=~A;
                2'b01:Y=A+B;
                2'b10:Y=A-B;
                2'b11:Y=2*A;
                default:Y=4'bzzzz;
                endcase
        end
endmodule
```

Problem 1: Verilog Test Bench:

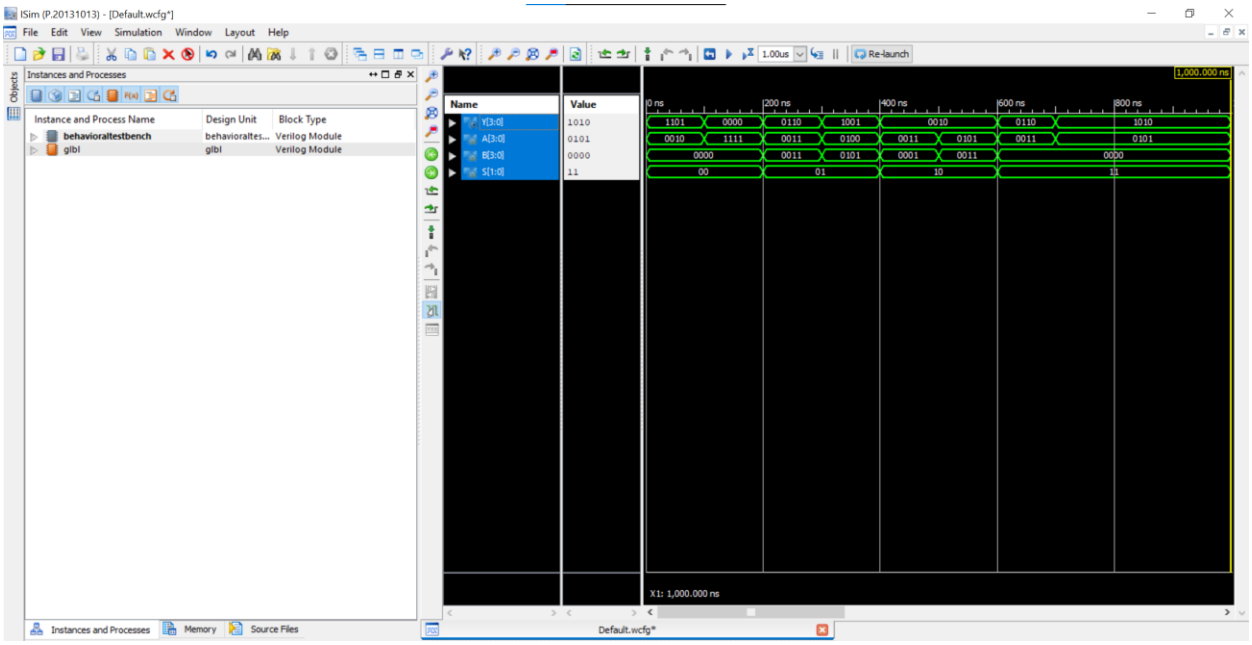
```
module behavioraltestbench;

	// Inputs
	reg [3:0] A;
	reg [3:0] B;
	reg [1:0] S;

	// Outputs
	wire [3:0] Y;

	// Instantiate the Unit Under Test (UUT)
	BehavioralALU_lab2 uut (
		.A(A),
		.B(B),
		.S(S),
		.Y(Y)
	);

	initial begin
		A = 2;
		B = 0;
		S[0]=0;
		S[1]=0;


		#100;


		A = 15;
		B = 0;
		S[0]=0;
		S[1]=0;


		#100;


		A = 3;
		B = 3;
		S[0]=1;
		S[1]=0;


		#100;
```

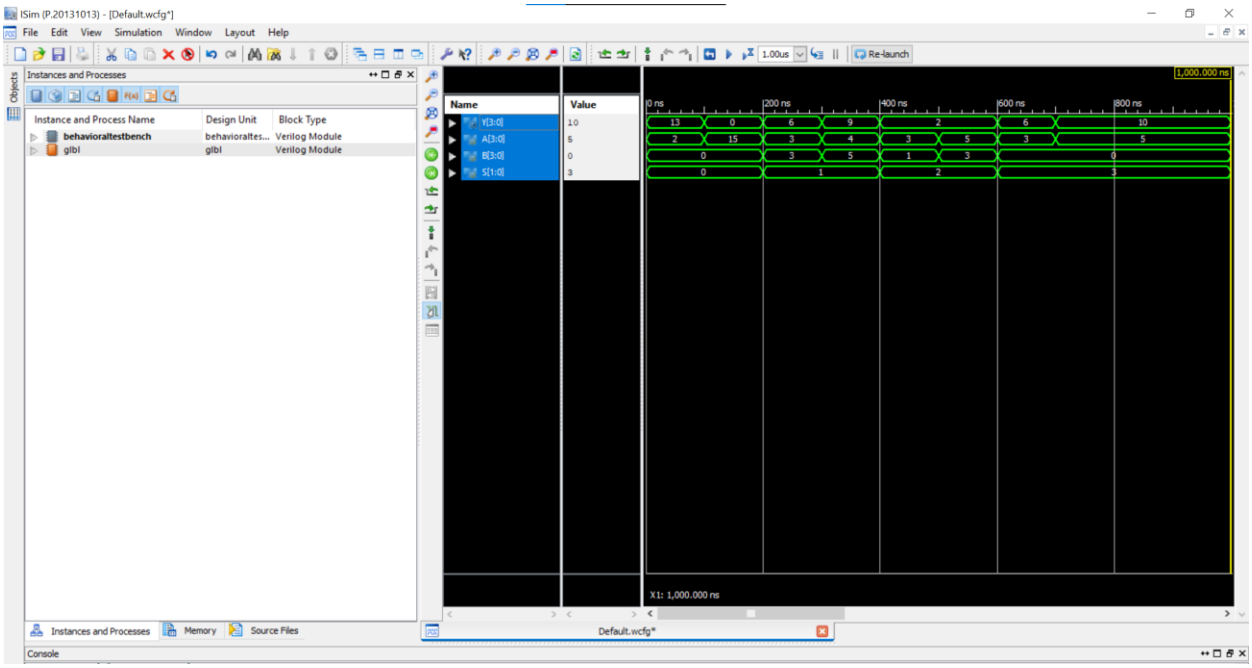
```
        A = 4;
        B = 5;
        S[0]=1;
        S[1]=0;

        #100;
        A = 3;
        B = 1;
        S[0] = 0;
        S[1]=1;

        #100;
        A = 5;
        B = 3;
        S[0] = 0;
        S[1]=1;



        #100;

        A = 3;
        B = 0;
        S[0] = 1;
        S[1]=1;

        #100;

        A = 5;
        B = 0;
        S[0] = 1;
        S[1]=1;



        // Add stimulus here


    end

endmodule
```


Design
View: Implementation Simulation
Behavioral
Hierarchy
lab2
xc6slx45-3csg324
adder4bit (adder4bit.v)
behavioraltestbench (behavioraltestbench.v)
uut - BehavioralALU_lab2 (BehavioralALU_
inverter_testbench (inverter_testbench.v)
mux4_1testbench (mux4_1testbench.v)
mux_21_beh_testbench (mux_21_beh_testben
mux_21_case_testbench (mux_21_case_testbe
mux_21_simple_testbench (mux_21_simple_te
structuralALU_testbench (structuralALU_testb
testbenchmux4 (testbenchmux4.v)
No Processes Running
Processes: behavioraltestbench
ISim Simulator
Behavioral Check Syntax
Simulate Behavioral Model
// Revision 0.01 - File Created
// Additional Comments:
module behavioraltestbench;
// Inputs
reg [3:0] A;
reg [3:0] B;
reg [1:0] S;
// Outputs
wire [3:0] Y;
// Instantiate the Unit Under Test (UUT)
BehavioralALU_lab2 uut (
.A(A),
.B(B),
.S(S),
.Y(Y)
);
initial begin
A = 2;
B = 0;
S[0]=0;
S[1]=0;
#100;
A = 15;
B = 0;
S[0]=0;
S[1]=0;
#100;
Start
Design
Files
Symbols
inverter_testbench.v
structuralALU_testbench.v
mux_21_beh.v
inverter.v
adder4bit.v
Fulladder4bits.v
behavioraltestbench.v
Console
ISim simulation engine GUI launched successfully
Process "Simulate Behavioral Model" completed successfully
Console
Errors
Warnings
Find in Files Results
Ln 47 Col 14
Verilog

Problem 1: Simulation Waveforms (add as many pages as you need).

Include *atleast* two test vectors per function. For example, demonstrate through the waveforms, that the ALU performs inversion correctly on two inputs, say, 0010 and 1111.





Problem 2: Behavioral Components (insert 1 page per component):

2 bit input- 1 bit select- 1 bit output MUX:

```
module mux_21_case(in0, in1, sel, out);
input sel;
input in0;
input in1;
output out;

assign out= (sel)?in1:in0;

endmodule
```

Inverter:

```
module inverter(inp,op);
   input[3:0] inp;
   output[3:0] op;

assign op=~inp;


endmodule
```

Adder:

```
module Fulladder4bits( A, B, Y);

input [3:0] A;
input [3:0] B;
output [3:0] Y;

assign Y= A+B;


endmodule
```

Structural 4:1 MUX made from 2:1 behavioral mux from above: (1 bit input, 4 select lines, 2 bit select bits, 1 bit output)

```
module mux4_1(in0,in1,in2,in3,sel0, sel1, out);
input in0,in1,in2,in3,sel0,sel1;

output out;


wire t,u ;



mux_21_simple mux1 (
.in0(in0),
.in1(in1),
.sel(sel0),
.out(t));



mux_21_simple mux2 (
.in0(in2),
.in1(in3),
.sel(sel0),
.out(u));



mux_21_simple mux3 (
.in0(t),
.in1(u),
.sel(sel1),
.out(out));




endmodule
```

Structural 4-bit input mux 2 select line bits 4 bit output mux, made from the structural 4:1 mux from above:

```
module mux4bitinput_4bitoutput( A,B,C,D,S0,S1,Y);

input [3:0] A;
input [3:0] B;
input [3:0] C;
input [3:0] D;
input S0;
input S1;

output [3:0] Y;



mux4_1 mux41(
.in0(A[0]),
.in1(B[0]),
.in2(C[0]),
.in3(D[0]),
.out(Y[0]),
.sel0(S0),
.sel1(S1));

mux4_1 mux42(
.in0(A[1]),
.in1(B[1]),
.in2(C[1]),
.in3(D[1]),
.sel0(S0),
.sel1(S1),
.out(Y[1]));

mux4_1 mux43(
.in0(A[2]),
.in1(B[2]),
.in2(C[2]),
.in3(D[2]),
.sel0(S0),
.sel1(S1),
.out(Y[2]));
```

```
mux4_1 mux44(
.in0(A[3]),
.in1(B[3]),
.in2(C[3]),
.in3(D[3]),
.sel0(S0),
.sel1(S1),
.out(Y[3]));



endmodule
```

Problem 2: ALU Structural Verilog (Use as many pages as needed.)

```
module StructuraALU_lab2(A,B,S0,S1,C,Y);

        input [3:0] A;
        input [3:0] B;
        input S0;
        input S1;
        input [3:0] C;

        output[3:0] Y;


        wire[3:0]Ainverted;
        wire [3:0] Binverted;
        wire [3:0] Bcompliment;
        wire [3:0] selectline1;
        wire [3:0] selectline2;
        wire [3:0] selectline3;

        inverter inverterA(
        .inp(A),
        .op(Ainverted));

        inverter inverterB(
        .inp(B),
        .op(Binverted));

        Fulladder4bits complementer(
        .A(Binverted),
        .B(C),
        .Y(Bcompliment));

        Fulladder4bits AplusB(
        .A(A),
        .B(B),
        .Y(selectline2));

        Fulladder4bits AminusB(
        .A(A),
        .B(Bcompliment),
        .Y(selectline1));
```

```
        Fulladder4bits doubleA(
        .A(A),
        .B(A),
        .Y(selectline3));

        mux4bitinput_4bitoutput bigone(
        .A(Ainverted),
        .B(selectline1),
        .C(selectline2),
        .D(selectline3),
        .Y(Y),
        .S1(S1),
        .S0(S0));



endmodule
```

Problem 2: Verilog Test Bench :

```
module structuralALU_testbench;

	// Inputs
	reg [3:0] A;
	reg [3:0] B;
	reg S0;
	reg S1;
	reg [3:0] C;

	// Outputs
	wire [3:0] Y;

	// Instantiate the Unit Under Test (UUT)
	StructuraALU_lab2 uut (
		.A(A),
		.B(B),
		.S0(S0),
		.S1(S1),
		.C(C),
		.Y(Y)
	);

	initial begin
		// Initialize Inputs
		A = 15;
		B = 0;
		C = 1;
		S0=0;
		S1=0;


		#100;
		A = 2;
		B = 0;
		C = 1;
		S0=0;
		S1=0;


		#100;
		A = 3;
		B = 3;
		C = 1;
		S0=0;
```

```
S1=1;

#100;
A = 5;
B = 2;
S0 =0;
S1=1;
C = 1;


#100;
A = 10;
B = 7;
S0 =1;
S1=0;
C = 1;

#100;
A = 9;
B = 5;
S0 =1;
S1=0;
C=1;


#100;

A = 7;
B = 1;
S0 = 1;
S1=1;
C = 1;


#100;

A = 6;
B = 1;
S0 = 1;
S1=1;
C = 1;
```

```
        end

endmodule
```

Problem 2: Simulation Results

Include *atleast* two test vectors per function. For example, demonstrate through the waveforms, that the ALU performs inversion correctly on two inputs, say, 0010 and 1111.