CECS 225

Lab 4

Sotheanith Sok

1. Adder\_32 Verilog Module source

`timescale 1ns / 1ps

module Adder\_32(augend, addend, sum);

input [31:0] augend, addend;

output [31:0] sum;

wire Carry0\_1, Carry1\_2, Carry2\_3;

RCA8 rca8\_0 ( .A\_8 ( augend[7:0] ),

.B\_8 ( addend[7:0] ),

.Cin ( 1'b0 ),

.Cout ( Carry0\_1 ),

.S\_8 ( sum[7:0] )

);

RCA8 rca8\_1 ( .A\_8 ( augend[15:8] ),

.B\_8 ( addend[15:8] ),

.Cin ( Carry0\_1 ),

.Cout ( Carry1\_2 ),

.S\_8 ( sum[15:8] )

);

RCA8 rca8\_2 ( .A\_8 ( augend[23:16] ),

.B\_8 ( addend[23:16] ),

.Cin ( Carry1\_2 ),

.Cout ( Carry2\_3 ),

.S\_8 ( sum[23:16] )

);

RCA8 rca8\_3 ( .A\_8 ( augend[31:24] ),

.B\_8 ( addend[31:24] ),

.Cin ( Carry2\_3 ),

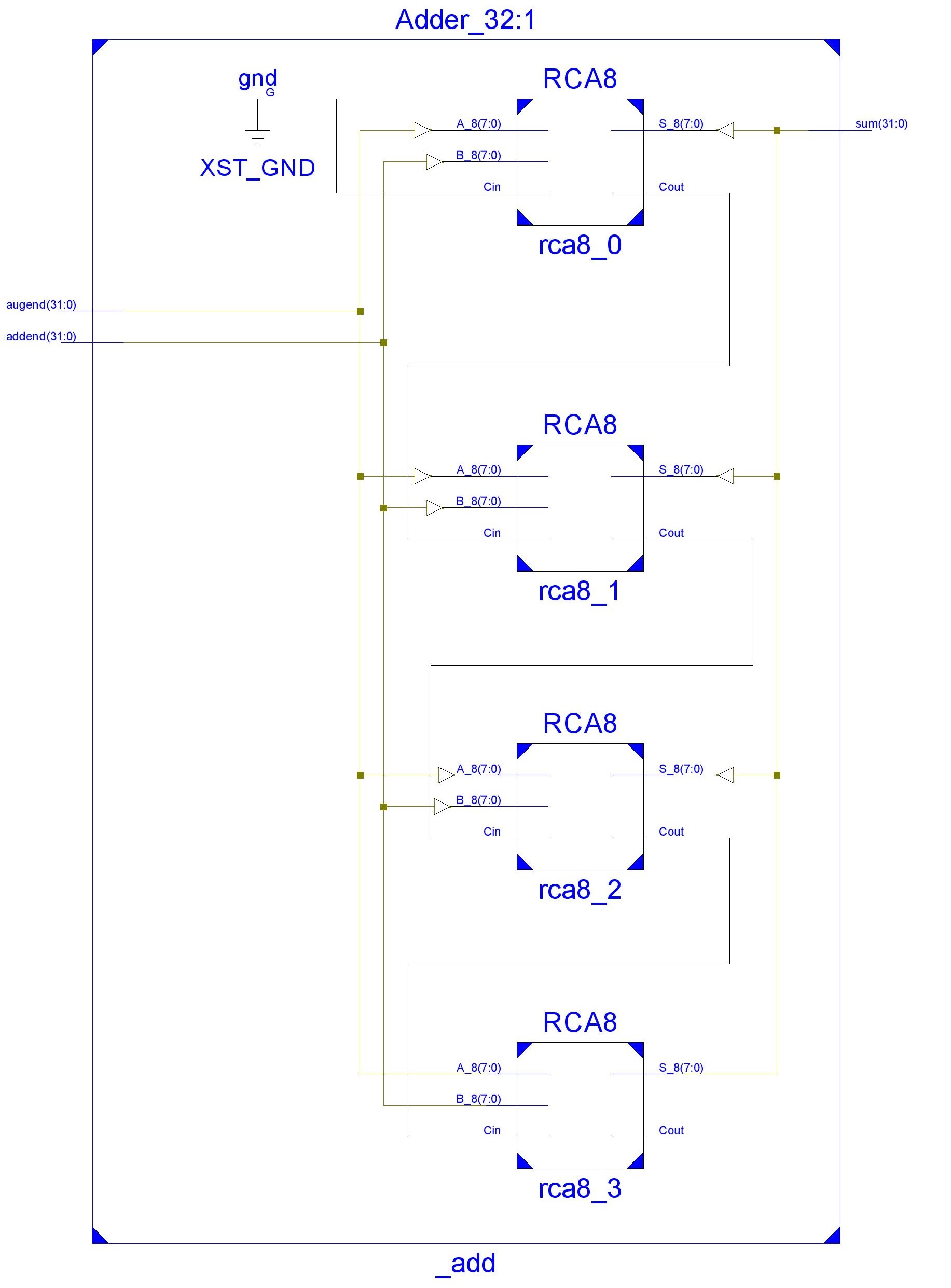
.Cout( ),

.S\_8 ( sum[31:24] )

);

endmodule

1. Adder\_32’s RTL Schematic



1. ALU Verilog Module Source

`timescale 1ns / 1ps

module ALU(ALUControl, A, B, Result, Zero\_Flag);

input [2:0] ALUControl;

input [31:0] A, B;

output [31:0] Result;

output Zero\_Flag;

wire [31:0] ANDout, ORout, SUMout, DIFFout, LTout;

AND\_32 \_and ( .argA (A),

.argB (B),

.AandB (ANDout)

);

OR\_32 \_or ( .argA (A),

.argB (B),

.AorB (ORout)

);

Adder\_32 \_add ( .augend (A),

.addend (B),

.sum (SUMout)

);

Subtracter\_32 \_sub ( .subtrahend (A),

.minuend (B),

.difference (DIFFout)

);

SetLessThan\_32 \_slt ( .argA (A),

.argB (B),

.AltB(LTout)

);

Mux8to1\_32bit \_m81 ( .Sel (ALUControl),

.In0 (ANDout),

.In1 (ORout),

.In2 (SUMout),

.In3 (32'bX),

.In4 (32'bX),

.In5 (32'bX),

.In6 (DIFFout),

.In7 (LTout),

.Out (Result)

);

assign Zero\_Flag = ( Result ) ? 1'b0 : 1'b1;

endmodule

1. ALU Verilog Test Fixture

`timescale 1ns / 1ps

module ALU\_Tester;

// Inputs

reg [2:0] ALUControl;

reg [31:0] A;

reg [31:0] B;

// Outputs

wire [31:0] Result;

wire Zero\_Flag;

// Instantiate the Unit Under Test (UUT)

ALU uut (

.ALUControl(ALUControl),

.A(A),

.B(B),

.Result(Result),

.Zero\_Flag(Zero\_Flag)

);

initial begin

// Case 0:AND

ALUControl = 3'b0;

A = 32'h13256189;

B = 32'h13256189+1;

// Wait 100 ns for global reset to finish

#100;

// Case 1:OR

ALUControl = 3'b001;

A = 32'h13256189;

B = 32'h13256189+1;

// Wait 100 ns for global reset to finish

#100;

// Case 2:ADD

ALUControl = 3'b010;

A = 32'h13256189;

B = 32'h13256189+1;

// Wait 100 ns for global reset to finish

#100;

// Case 3:SUB

ALUControl = 3'b110;

A = 32'h13256189;

B = 32'h13256189+1;

// Wait 100 ns for global reset to finish

#100;

// Case 4:SLT

ALUControl = 3'b111;

A = 32'h13256189;

B = 32'h13256189+1;

// Wait 100 ns for global reset to finish

#100;

// Case 5:SLT

ALUControl = 3'b111;

A = 32'h13256189+1;

B = 32'h13256189;

// Wait 100 ns for global reset to finish

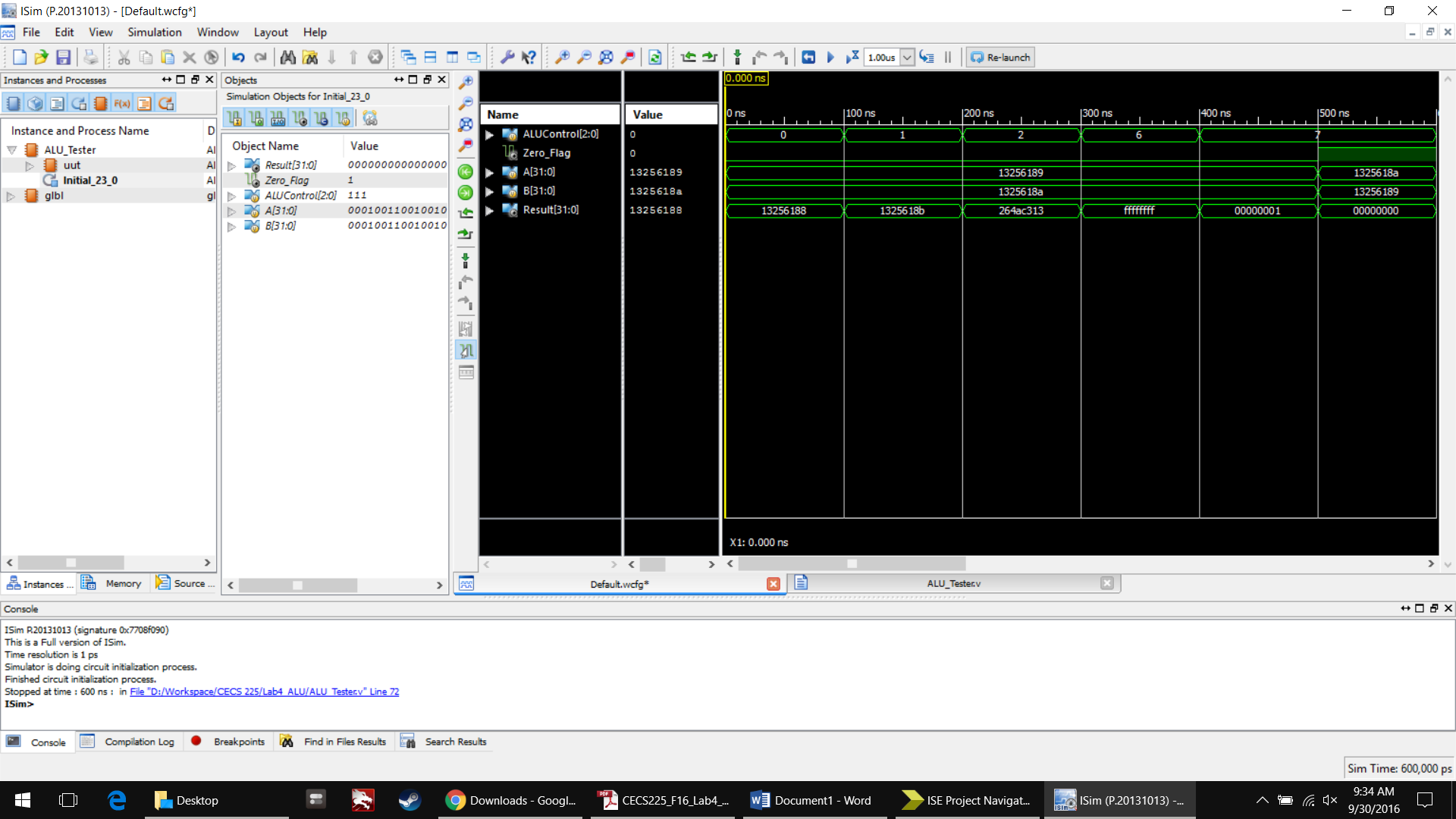
#100;

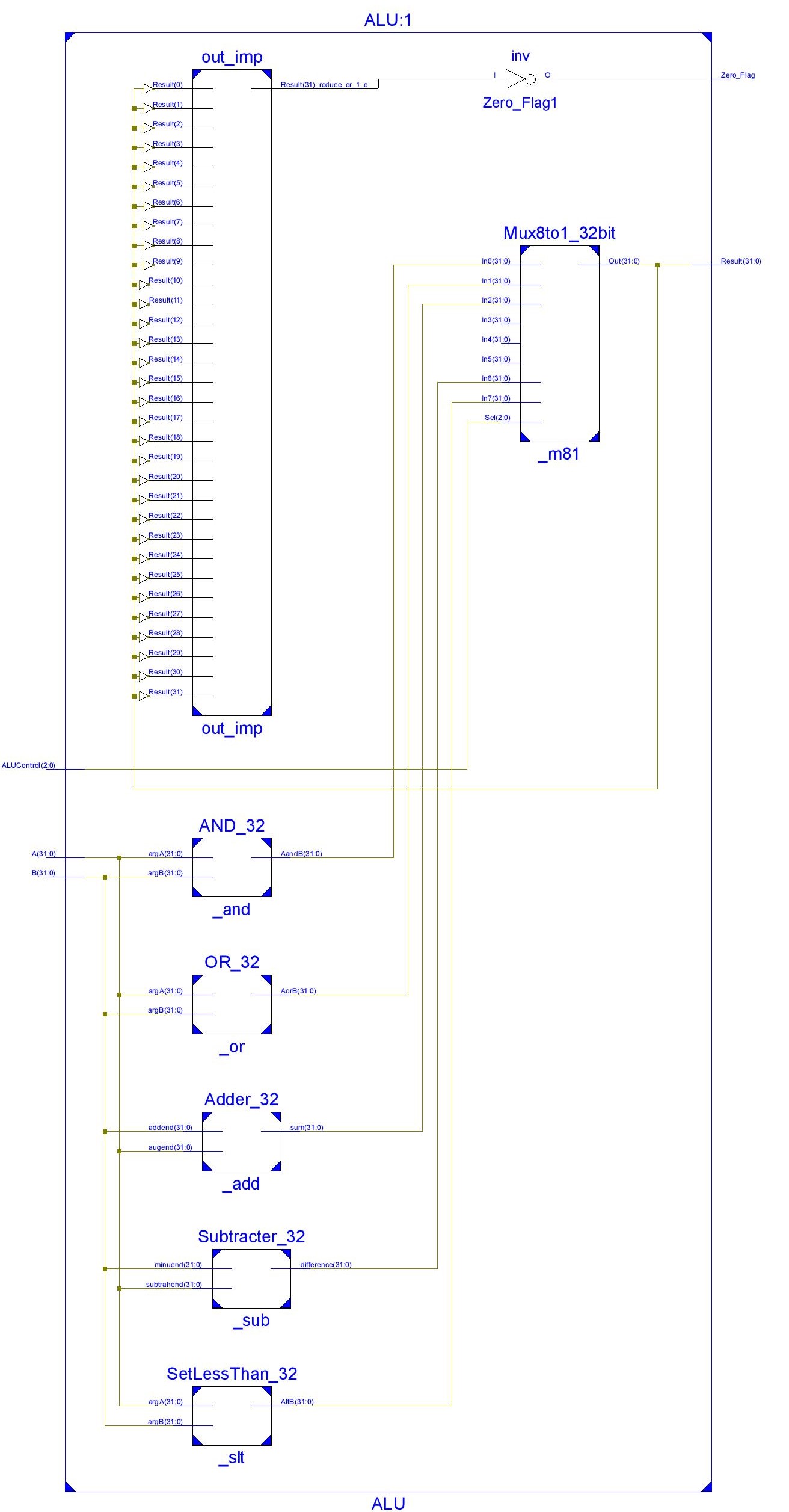
$stop;

end

endmodule

1. ALU Simulation Result



1. ALU RTL Schematic