CECS 225

Lab 3

Sotheanith Sok

1. **ALU\_Decoder Verilog module source**

`timescale 1ns / 1ps

module ALU\_Decoder( ALUOp, Funct, ALUControl);

input [1:0] ALUOp;

input [5:0] Funct;

output [2:0] ALUControl;

assign ALUControl[2]=ALUOp[0]|ALUOp[1]&Funct[1];

assign ALUControl[1]=~ALUOp[1]|ALUOp[1]&(~Funct[2]);

assign ALUControl[0]=ALUOp[1]&(Funct[3]|Funct[0]);

endmodule

1. **ALU\_Decoder Verilog Test Fixture**

`timescale 1ns / 1ps

module ALU\_Decoder\_Tester;

// Inputs

reg [1:0] ALUOp;

reg [5:0] Funct;

// Outputs

wire [2:0] ALUControl;

// Instantiate the Unit Under Test (UUT)

ALU\_Decoder uut (

.ALUOp(ALUOp),

.Funct(Funct),

.ALUControl(ALUControl)

);

initial begin

//Test Case 0

ALUOp = 2'b0;

Funct = 6'bX;

#10;

//Test Case 1

ALUOp = 2'b01;

Funct = 6'bX;

#10;

//Test Case 2

ALUOp = 2'b10;

Funct = 6'bXX0000;

#10;

//Test Case 3

ALUOp = 2'b10;

Funct = 6'bXX0010;

#10;

//Test Case 4

ALUOp = 2'b10;

Funct = 6'bXX0100;

#10;

//Test Case 5

ALUOp = 2'b10;

Funct = 6'bXX0101;

#10;

//Test Case 6

ALUOp = 2'b10;

Funct = 6'bXX1010;

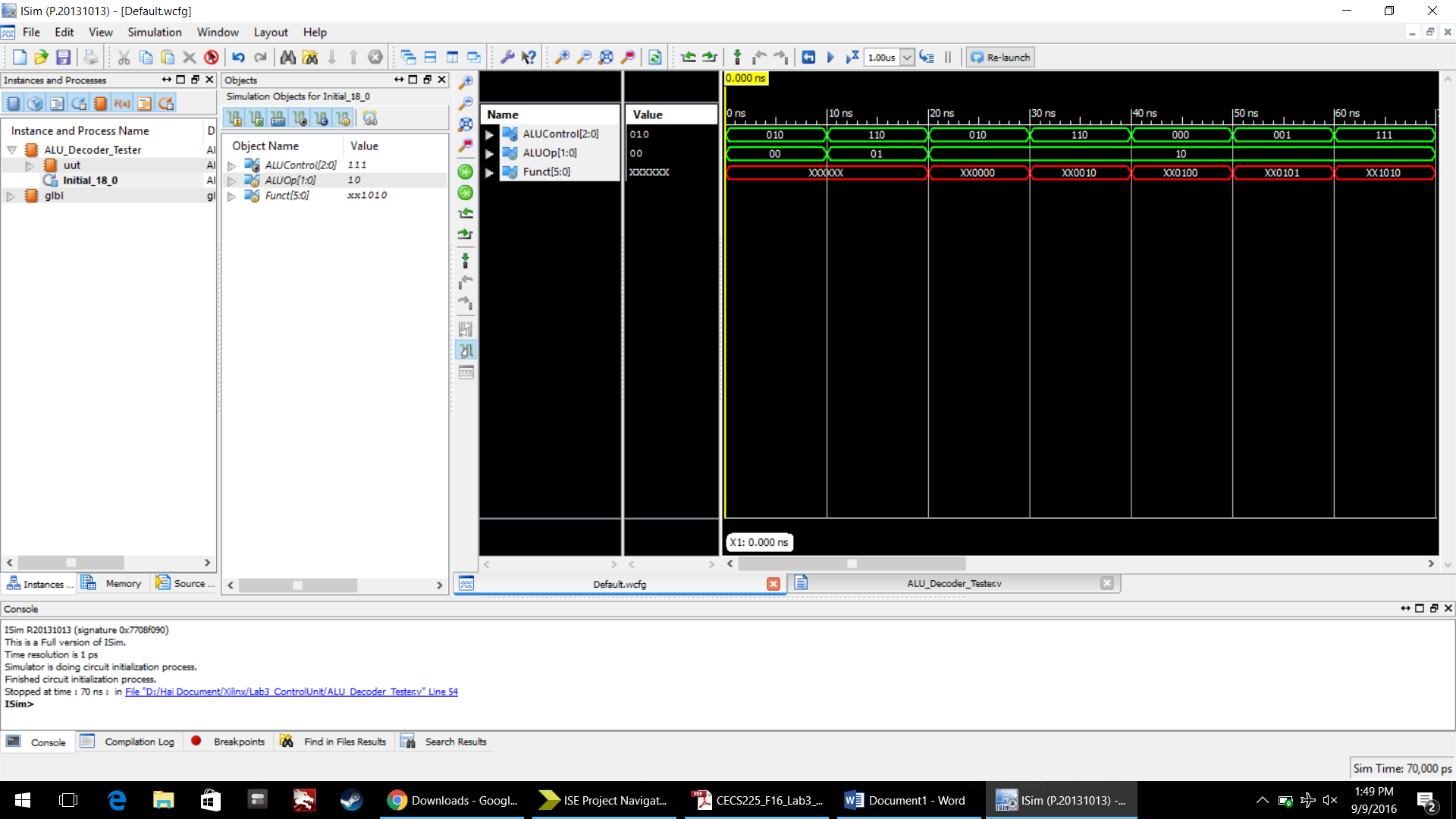
#10;

$stop;

end

endmodule

1. **ALU\_Decoder Simulation**



1. **Main\_Decoder Verilog module source**

`timescale 1ns / 1ps

module Main\_Decoder( Opcode, RegWrite, RegDst,

ALUSrc, Branch, MemWrite,

MemtoReg, ALUOp);

input [5:0] Opcode;

output RegWrite, RegDst, ALUSrc, Branch, MemWrite, MemtoReg;

output [1:0] ALUOp;

wire R\_Format, lw, sw, beg;

assign R\_Format=~(Opcode[5]|Opcode[4]|Opcode[3]|Opcode[2]|Opcode[1]|Opcode[0]);

assign lw=Opcode[5]&Opcode[1]&Opcode[0]&~(Opcode[4]|Opcode[3]|Opcode[2]);

assign sw=Opcode[5]&~Opcode[4]&Opcode[3]&~Opcode[2]&Opcode[1]&Opcode[0];

assign beq=Opcode[2]&~(Opcode[5]|Opcode[4]|Opcode[3]|Opcode[1]|Opcode[0]);

assign RegWrite=R\_Format|lw;

assign RegDst=R\_Format;

assign ALUSrc=lw|sw;

assign Branch=beq;

assign MemWrite=sw;

assign MemtoReg=lw;

assign ALUOp[1]=R\_Format;

assign ALUOp[0]=beq;

endmodule

1. **Main\_Decoder Verilog Test Fixture**

`timescale 1ns / 1ps

module Main\_Decoder\_Tester;

// Inputs

reg [5:0] Opcode;

// Outputs

wire RegWrite;

wire RegDst;

wire ALUSrc;

wire Branch;

wire MemWrite;

wire MemtoReg;

wire [1:0] ALUOp;

// Instantiate the Unit Under Test (UUT)

Main\_Decoder uut (

.Opcode(Opcode),

.RegWrite(RegWrite),

.RegDst(RegDst),

.ALUSrc(ALUSrc),

.Branch(Branch),

.MemWrite(MemWrite),

.MemtoReg(MemtoReg),

.ALUOp(ALUOp)

);

initial begin

//Test Case 0

Opcode = 6'b000000;

#10;

//Test Case 1

Opcode = 6'b100011;

#10;

//Test Case 2

Opcode = 6'b101011;

#10;

//Test Case 3

Opcode = 6'b000100;

#10;

//Test Case 4

Opcode = 6'b111111;

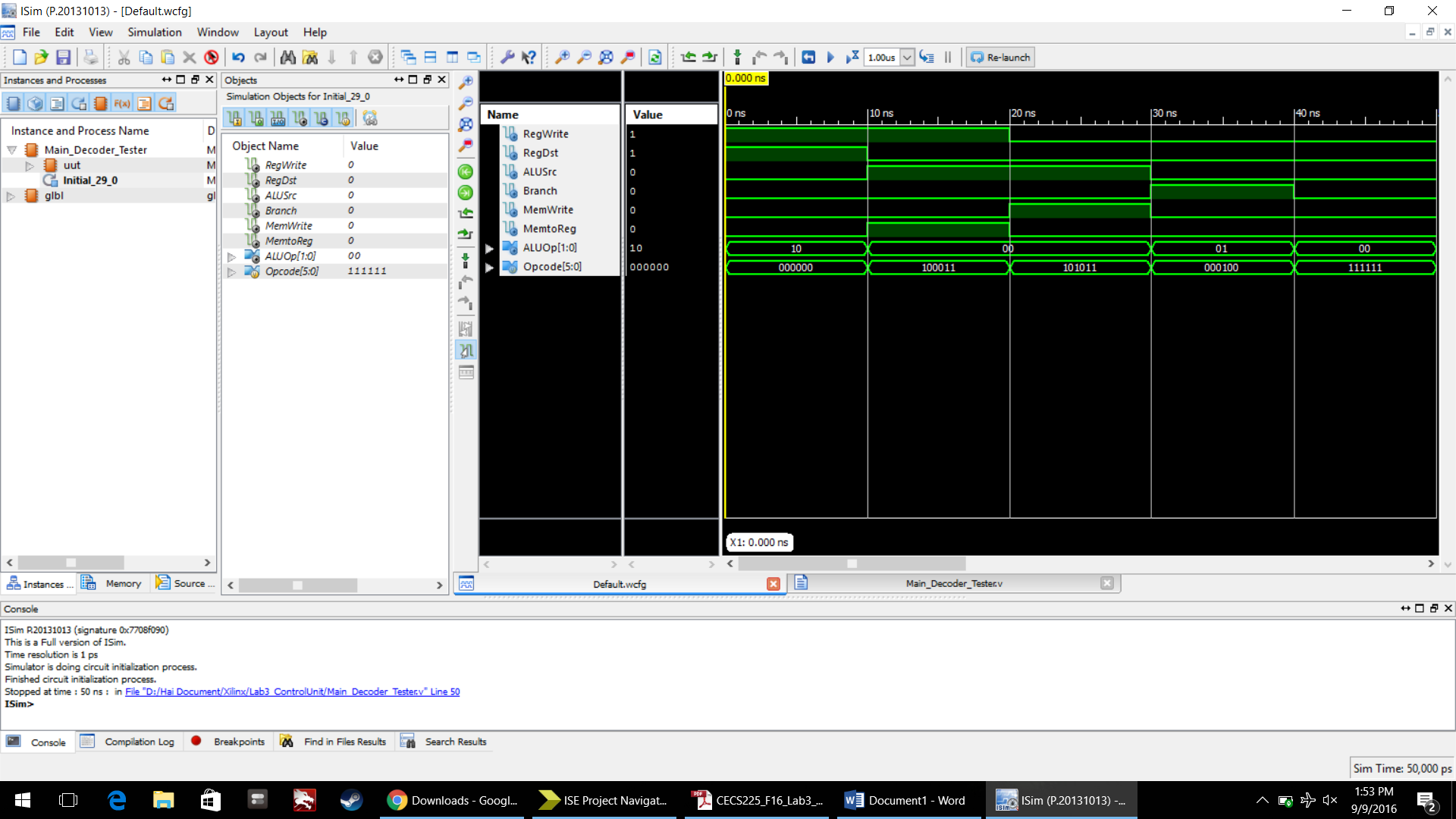
#10;

$stop;

end

endmodule

1. **Main\_Decoder Simulation**



1. **Control\_Unit Verilog module source**

`timescale 1ns / 1ps

module Control\_Unit( Opcode, Funct, MemtoReg, MemWrite,

Branch, ALUSrc, RegDst, RegWrite,

ALUControl);

input [5:0] Opcode,Funct;

output MemtoReg, MemWrite, Branch,ALUSrc, RegDst, RegWrite;

output [2:0] ALUControl;

wire [1:0] ALUOp\_md\_to\_ad;

ALU\_Decoder ad( .ALUOp(ALUOp\_md\_to\_ad),

.Funct(Funct),

.ALUControl(ALUControl)

);

Main\_Decoder md( .Opcode(Opcode),

.RegWrite(RegWrite),

.RegDst(RegDst),

.ALUSrc(ALUSrc),

.Branch(Branch),

.MemWrite(MemWrite),

.MemtoReg(MemtoReg),

.ALUOp(ALUOp\_md\_to\_ad)

);

Endmodule

1. **Control\_Unit Verilog Test Fixture**

`timescale 1ns / 1ps

module Control\_Unit\_Tester;

// Inputs

reg [5:0] Opcode;

reg [5:0] Funct;

// Outputs

wire MemtoReg;

wire MemWrite;

wire Branch;

wire ALUSrc;

wire RegDst;

wire RegWrite;

wire [2:0] ALUControl;

// Instantiate the Unit Under Test (UUT)

Control\_Unit uut (

.Opcode(Opcode),

.Funct(Funct),

.MemtoReg(MemtoReg),

.MemWrite(MemWrite),

.Branch(Branch),

.ALUSrc(ALUSrc),

.RegDst(RegDst),

.RegWrite(RegWrite),

.ALUControl(ALUControl)

);

initial begin

// Test Case 0

Opcode = 6'b0;

Funct = 6'b100000;

#10;

// Test Case 1

Opcode = 6'b0;

Funct = 6'b100010;

#10;

// Test Case 2

Opcode = 6'b0;

Funct = 6'b100100;

#10;

// Test Case 3

Opcode = 6'b0;

Funct = 6'b100101;

#10;

// Test Case 4

Opcode = 6'b0;

Funct = 6'b101010;

#10;

// Test Case 5

Opcode = 6'b100011;

Funct = 6'bX;

#10;

// Test Case 6

Opcode = 6'b101011;

Funct = 6'bX;

#10;

// Test Case 7

Opcode = 6'b000100;

Funct = 6'bX;

#10;

// Test Case 8

Opcode = 6'b111111;

Funct = 6'bX;

#10;

$stop;

end

endmodule

1. **Control\_Unit Simulation**

