CECS 225

Lab 3

Ronnie Casasola

012737398

Section 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
```

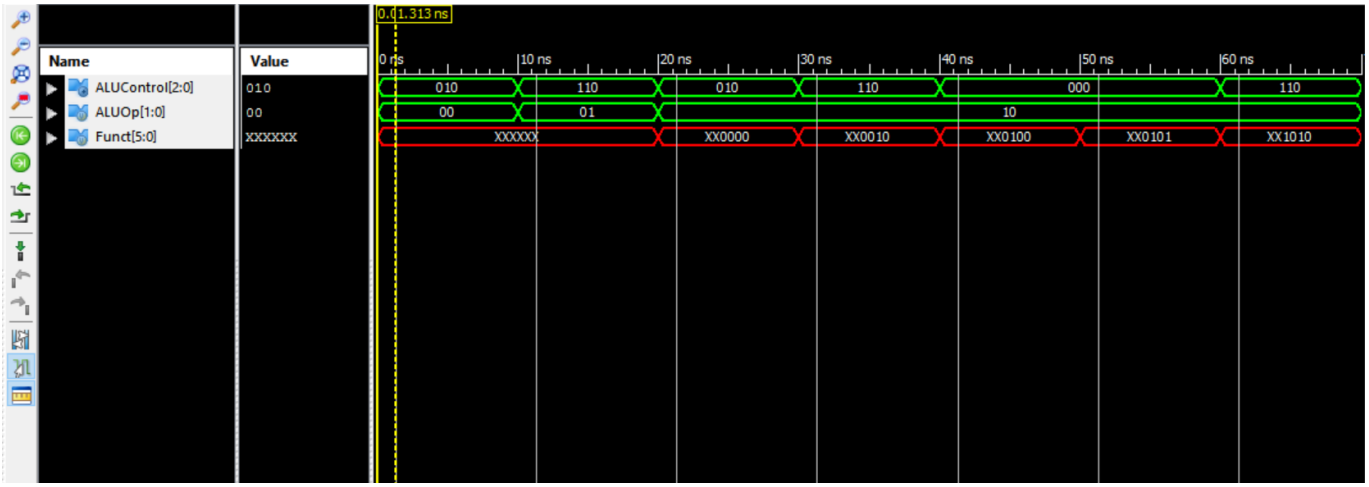
Section 2: ALU_Decoder Verilog Test Fixture

```
`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
```

Section 3: ALU_Decoder Simulation Wavelength screenshot



Section 4: Main_Decoder Verilog module source

```
`timescale 1ns / 1ps

module Main_Decoder(
    input [5:0] Opcode,
    output [1:0] ALUOp,
    output RegWrite,
    output RegDst,
    output ALUSrc,
    output Branch,
    output MemWrite,
    output MemtoReg
    );

        wire R_Format, lw, sw, beq;

        assign R_Format =
~Opcode[5]*~Opcode[4]*~Opcode[3]*~Opcode[2]*~Opcode[1]*~Opcode[0];
        assign lw = Opcode[5]*~Opcode[4]*~Opcode[3]*~Opcode[2]*Opcode[1]*Opcode[0];
        assign sw = Opcode[5]*~Opcode[4]*Opcode[3]*~Opcode[2]*Opcode[1]*Opcode[0];
        assign beq =
~Opcode[5]*~Opcode[4]*~Opcode[3]*Opcode[2]*~Opcode[1]*~Opcode[0];

        assign RegWrite =
(~Opcode[4]*~Opcode[3]*~Opcode[2])*(Opcode[5]+~Opcode[1])*(~Opcode[5]+Opcode[0])*(
Opcode[0]+~Opcode[1])*(Opcode[1]+Opcode[0]);
        assign RegDst = ~(Opcode[5] + Opcode[4] + Opcode[3]+ Opcode[2]+ Opcode[1] +
Opcode[0]);
        assign ALUSrc = (Opcode[5] * ~Opcode[4] * ~Opcode[2] * Opcode[1] * Opcode[0]);
```

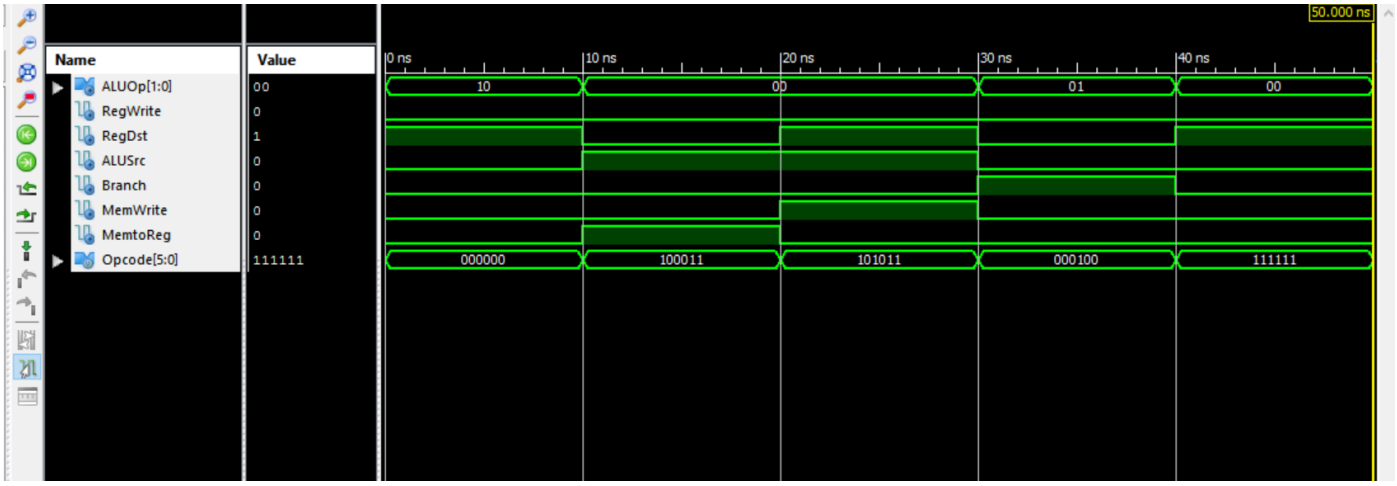
```
	assign Branch =
~Opcode[5]*~Opcode[4]*~Opcode[3]*Opcode[2]*~Opcode[1]*~Opcode[0];

	assign MemWrite =
Opcode[5]*~Opcode[4]*Opcode[3]*~Opcode[2]*Opcode[1]*Opcode[0];

	assign MemtoReg =
Opcode[5]*~Opcode[4]*~Opcode[3]*~Opcode[2]*Opcode[1]*Opcode[0];

	assign ALUOp[1] =
~Opcode[5]*~Opcode[4]*~Opcode[3]*~Opcode[2]*~Opcode[1]*~Opcode[0];

	assign ALUOp[0] =
~Opcode[5]*~Opcode[4]*~Opcode[3]*Opcode[2]*~Opcode[1]*~Opcode[0];



endmodule
```

Section 5: Main_Decoder Verilog Test fixture

```
`timescale 1ns / 1ps

module Main_Decoder_Tester;

	// Inputs
	reg [5:0] Opcode;

	// Outputs
	wire [1:0] ALUOp;
	wire RegWrite;
	wire RegDst;
	wire ALUSrc;
	wire Branch;
	wire MemWrite;
	wire MemtoReg;

	// Instantiate the Unit Under Test (UUT)
	Main_Decoder uut (
		.Opcode(Opcode),
		.ALUOp(ALUOp),
		.RegWrite(RegWrite),
		.RegDst(RegDst),
		.ALUSrc(ALUSrc),
		.Branch(Branch),
		.MemWrite(MemWrite),
		.MemtoReg(MemtoReg)
	);
```

```
    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
```

Section 6: Main_Decoder simulation waveform screenshot



Section 7: Control_Unit Verilog module source

```
`timescale 1ns / 1ps

module Control_Unit(
    input [5:0] Opcode,
    input [5:0] Funct,
    output MemtoReg,
    output MemWrite,
    output Branch,
    output ALUSrc,
    output RegDst,
    output RegWrite,
    output [2:0] ALUControl
    );

        wire [1:0] ALUOp_md_to_ad;

        ALU_Decoder ad(
                                        .ALUOp(         ),
                                        .Funct(         ),
                                        .ALUControl(    )
                                    );

        Main_Decoder md(
                                        .Opcode(        ),
                                        .RegWrite(      ),
                                        .RegDst(        ),
```

```
.ALUSrc(          ),
.Branch(          ),
.MemWrite(        ),
.MemtoReg(        ),
.ALUOp(           )
);

Endmodule
```

Section 8: 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'b000000;
        Funct = 6'b100000;
        #10;

        //Test case 1
        Opcode = 6'b000000;
        Funct = 6'b100010;
        #10;

        //Test case 2
        Opcode = 6'b000000;
        Funct = 6'b100100;
        #10;

        //Test case 3
        Opcode = 6'b000000;
        Funct = 6'b100101;
        #10;

        //Test case 4
```

```
        Opcode = 6'b000000;
        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
```

Section 9: Control_Unit simulation screenshot