Name: Laiba Fatima Khan

CA Lab10 Report

TASK 1:

```
module Control_Unit
(
	input[6:0] Opcode,
	output reg Branch, MemRead, MemtoReg, MemWrite, ALUSrc, RegWrite,
	output reg[1:0] ALUOp

);

	always@(Opcode)
	begin
	case(Opcode)
		7'b0110011 :  //R-Type
		begin
		ALUSrc =	0;
		MemtoReg =	0;
		RegWrite = 1;
		MemRead =	0;
		MemWrite = 0;
		Branch =	0;
		ALUOp =		2'b10;
		end

		7'b0000011 : //I-Type (ld)
		begin
		ALUSrc <=	1;
		MemtoReg <=	1;
		RegWrite <= 1;
		MemRead <=	1;
		MemWrite <= 0;
		Branch <=	0;
		ALUOp <=	2'b00;
		end

		7'b0100011  : //I-Type (sd)
		begin
		ALUSrc <=	1;
		MemtoReg <=	1'bx;
		RegWrite <= 0;
		MemRead <=	0;
		MemWrite <= 1;
```

```
                Branch <=       0;
                ALUOp <=        2'b00;
                end

                7'b1100011  : //SB-Type (beq)
                begin
                ALUSrc <=       0;
                MemtoReg <=     1'bx;
                RegWrite <= 0;
                MemRead <=      0;
                MemWrite <= 0;
                Branch <=       1;
                ALUOp <=        2'b01;
                end

        endcase
        end

endmodule
```

TASK 2:

```
module ALU_Control
(
        input[1:0] ALUOp,
        input[3:0] Funct,
        output reg[3:0] Operation

);
        always@(ALUOp or Funct)
        begin
        case(ALUOp)
                2'b00: Operation = 4'b0010;
                2'b01: Operation = 4'b0110;
                2'b10:
                begin
                        case(Funct)
                        4'b0000 : Operation = 4'b0010;
                        4'b1000 : Operation = 4'b0110;
                        4'b0111 : Operation = 4'b0000;

                        4'b0110 : Operation = 4'b0001;
                        endcase
                end
        endcase
        end
endmodule
```

TASK 3:

Using Control Unit and ALU Control modules from task1 and task2 resp lab8

```
module top_control
(
	input[6:0] Opcode,
	input[3:0] Funct,
	output Branch, MemRead, MemtoReg, MemWrite, ALUSrc, RegWrite,
	output[3:0] Operation
);
	wire[1:0] ALUOp_wire;

	Control_Unit con
	(
		.Opcode(Opcode),
		.Branch(Branch),
		.MemRead(MemRead),
		.MemtoReg(MemtoReg),
		.MemWrite(MemWrite),
		.ALUSrc(ALUSrc),
		.RegWrite(RegWrite),
		.ALUOp(ALUOp_wire)

	);
	ALU_Control alu_con
	(
		.ALUOp(ALUOp_wire),
		.Funct(Funct),
		.Operation(Operation)
	);
endmodule

module tb
(
);
	reg[6:0] Opcode;
	reg[3:0] Funct;
	wire Branch;
	wire MemRead;
	wire MemtoReg;
	wire MemWrite;
	wire ALUSrc;
	wire RegWrite;
	wire[3:0] Operation;
```

```
        top_control top
        (
                .Opcode(Opcode),
                .Funct(Funct),
                .Branch(Branch),
                .MemRead(MemRead),
                .MemtoReg(MemtoReg),
                .MemWrite(MemWrite),
                .ALUSrc(ALUSrc),
                .RegWrite(RegWrite),
                .Operation(Operation)
        );
        initial
        begin
         Funct = 4'b1000;
         Opcode = 7'b0110011;  //R-Type
        #50
         Opcode = 7'b1100011;      //SB-Type
        #50
         Opcode =  7'b0000011; //I-Type (ld)
        end
endmodule
```