Course Name: Digital System Design

Course Number and Section: **14:332:437:02**

Experiment: Lab 7

Lab Instructor: Zhenghao Li

Date Performed: 10/29/24

Date Submitted: 11/12/24

Submitted by: Nikhil Chandra Donepudi Venkata Sambasiva

Prelab

Data Memory

```
module data_memory(
    input logic clk, rst,
    input logic [31:0] A,    //address
    input logic [31:0] WD,   //input data
    input logic WE,          // write enable
    output logic [31:0] RD,
    output logic [31:0] prode  // check data in data_memory
    );

    logic [31:0] mem_array [255:0];
    assign RD = mem_array[A[7:0]];
    assign prode = RD;

    always_ff @(posedge clk or negedge rst)
    begin
       if (!rst) begin
          for (int i = 0; i < 256; i++) begin
             mem_array[i] <= i;
          end
       end
       else begin
          if (WE) begin
                mem_array[A[7:0]] <= WD;
             end
       end
    end

endmodule
```

MUX_ALUSrc

```
module MUX_ALUSrc(
    input logic ALUSrc,                    //select signal
    input logic[31:0] fromReg,          //from regfile
    input logic[31:0] fromImm,          //immediate value
    output logic[31:0] ALUSrc_out //output data
);

    assign ALUSrc_out = ALUSrc ? fromReg : fromImm;

endmodule
```

MUX_RegDst

```
module MUX_RegDst(
    input logic [4:0] RT,            // rt field of instruction
    input logic [4:0] RD,            // rd field of instruction
    input logic RegDes,              // Select signal
    output logic [4:0] RegDes_out    // Output data
);

    assign RegDes_out = RegDes ? RT : RD;

endmodule
```

MUX_MemtoReg

```
module MUX_MemtoReg(
    input logic MemtoReg,
    input logic[31:0] ALUResult,
    input logic[31:0] RD, //from data memory
    output logic[31:0] MemtoReg_out
);

    assign MemtoReg_out = MemtoReg ? RD : ALUResult;

endmodule
```

SignExtend

```
module SignExtend(
    input logic [15:0] imm,
    output logic [31:0] signImm
);
    assign signImm = {{16{imm[15]}}, imm};  // Sign extends
endmodule
```

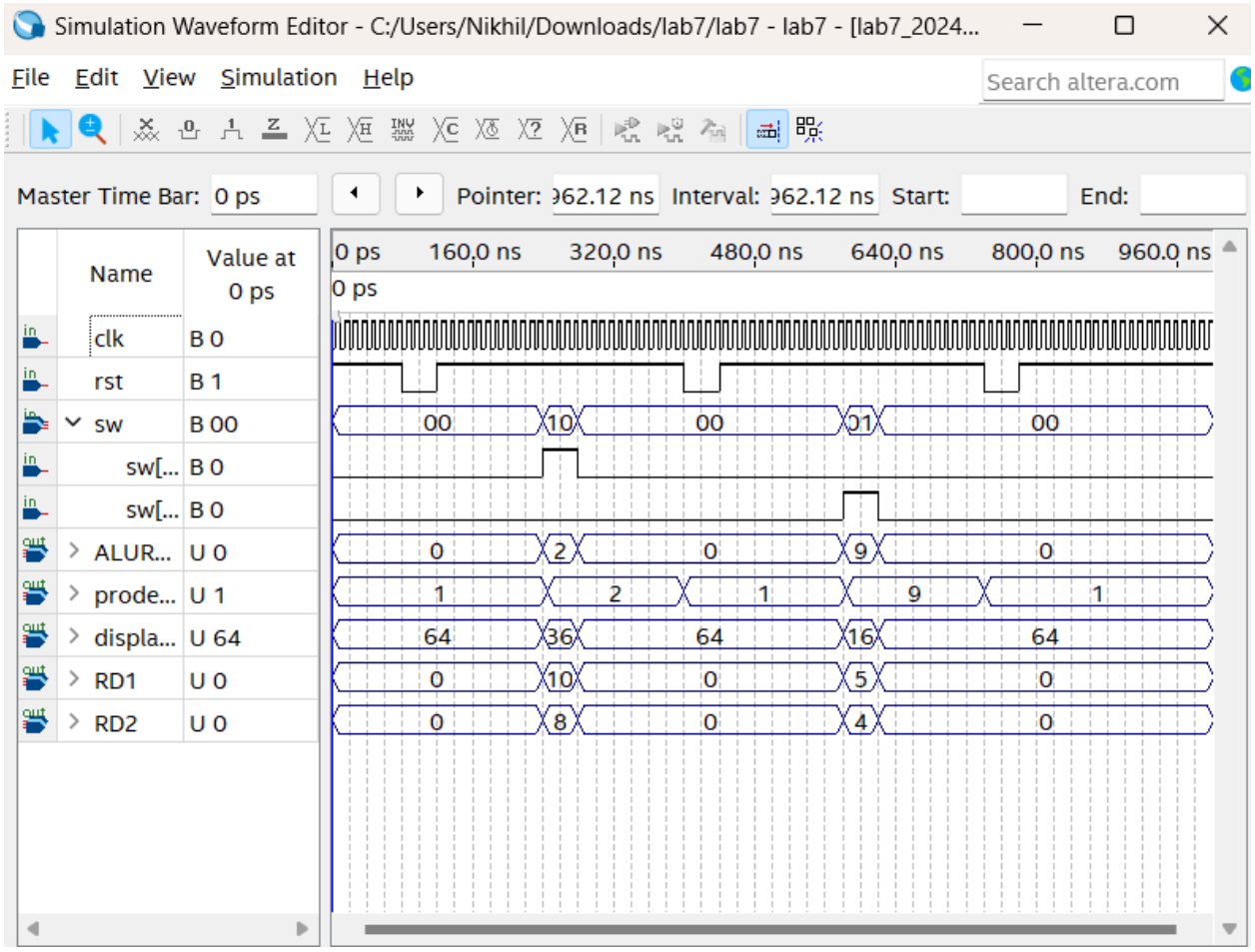
Top

```
module top(
    input logic clk, rst,
    input logic[1:0] sw, //address for instruction memory
    output logic[31:0] ALUResult, //output for pre-lab simulation
    output logic[31:0] RD1, RD2, //output for pre-lab simulation
    output logic[31:0] prode_register_file, //output for pre-lab simulation
    output logic[6:0] display_led //output for in-lab
);

 logic[31:0] inst_0 = 32'b0;
 logic[31:0] inst_1 = 32'b010101_00000_00001_0000_0000_0000_0101; // LW instruction
 logic[31:0] inst_2 = 32'b010100_00000_00110_0000_0000_0000_0010; // SW instruction
 logic[31:0] inst_ex;
 assign inst_ex = (sw==1)? inst_1:(sw==2)? inst_2: inst_0;

 // Intermediate signals for connections
 logic [4:0] regDst_out;
 logic [31:0] signImm, memData, ALU_in2, MemtoReg_out;
 logic MemtoReg, ALUSrc, WE;

 assign ALUSrc = 1;
 assign RegDst = 0;
 assign MemToReg = (inst_ex[31:26] == 6'b010101);
 assign WE3 = (inst_ex[31:26] ==  6'b010101);
 assign WE = (inst_ex[31:26] == 6'b010100); // store
 assign ALUControl = 3'b010; // sample control signal for ALU

register_file r_f(
    .clk(clk),
    .rst(rst),
    .A1(inst_ex[25:21]), // rs
    .A2(inst_ex[20:16]), // rt
    .A3(regDst_out),     // rd or rt based on RegDst
    .WD3(MemtoReg_out),  // Data to write back to register file
    .WE3(WE),
    .RD1(RD1),
    .RD2(RD2),
    .prode(prode_register_file)
);
```

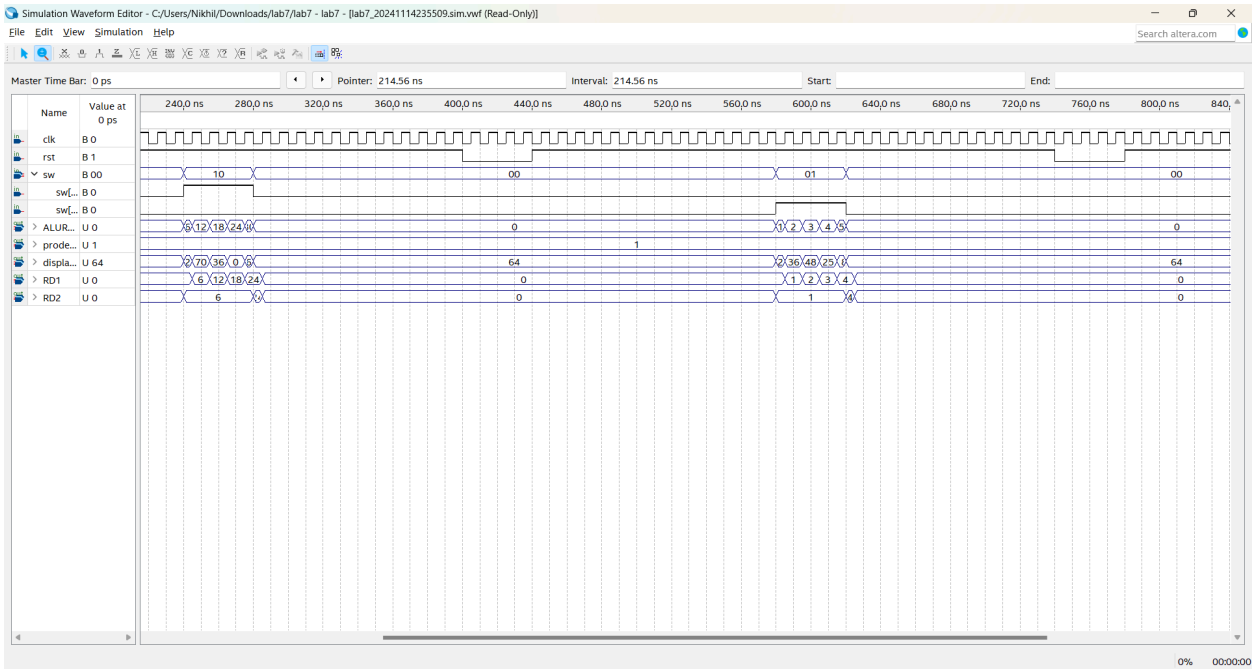
```
41    // Data Memory Instantiation
42  data_memory dm(
43        .clk(clk),
44        .rst(rst),
45        .A(ALUResult),
46        .WD(RD2),
47        .WE(WE),
48        .RD(memData),
49        .prode()
50  );
51
52  ALU t1(
53        .SrcA(RD1),
54        .SrcB(ALU_in2),
55        .ALUControl(inst_ex[29:27]),
56        .ALUResult(ALUResult)
57  );
58
59    // MUX for MemtoReg
60  MUX_MemtoReg mux_memtoreg(
61        .MemtoReg(inst_ex[31]),
62        .ALUResult(ALUResult),
63        .RD(memData),
64        .MemtoReg_out(MemtoReg_out)
65  );
66
67    // MUX for ALUSrc
68  MUX_ALUSrc mux_alusrc(
69        .fromReg(RD2),
70        .fromImm(signImm),
71        .ALUSrc(inst_ex[28]),
72        .ALUSrc_out(ALU_in2)
73  );
74
75    // MUX for RegDst
76  MUX_RegDst mux_regdst(
77        .RT(inst_ex[20:16]),
78        .RD(inst_ex[15:11]),
79        .RegDes(inst_ex[26]),
80        .RegDes_out(regDst_out)
81  );
82
```

```
83    // Sign Extend Module
84    SignExtend sign_ext(
85        .imm(inst_ex[15:0]),
86        .signImm(signImm)
87    );
88
89    display t2(
90        .data_in(ALUResult),
91        .segments(display_led)
92    );
93
94    endmodule
```

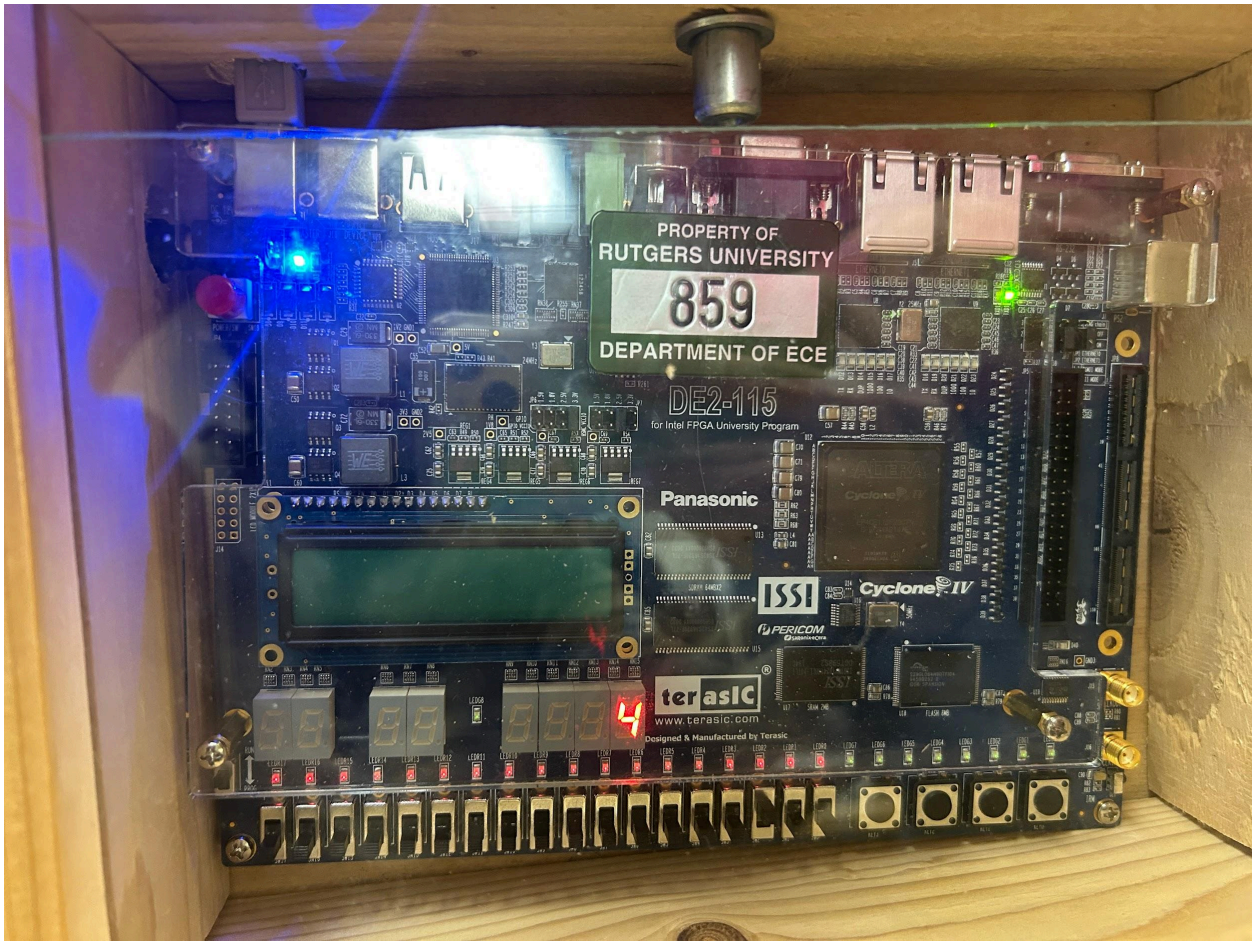
Waveform



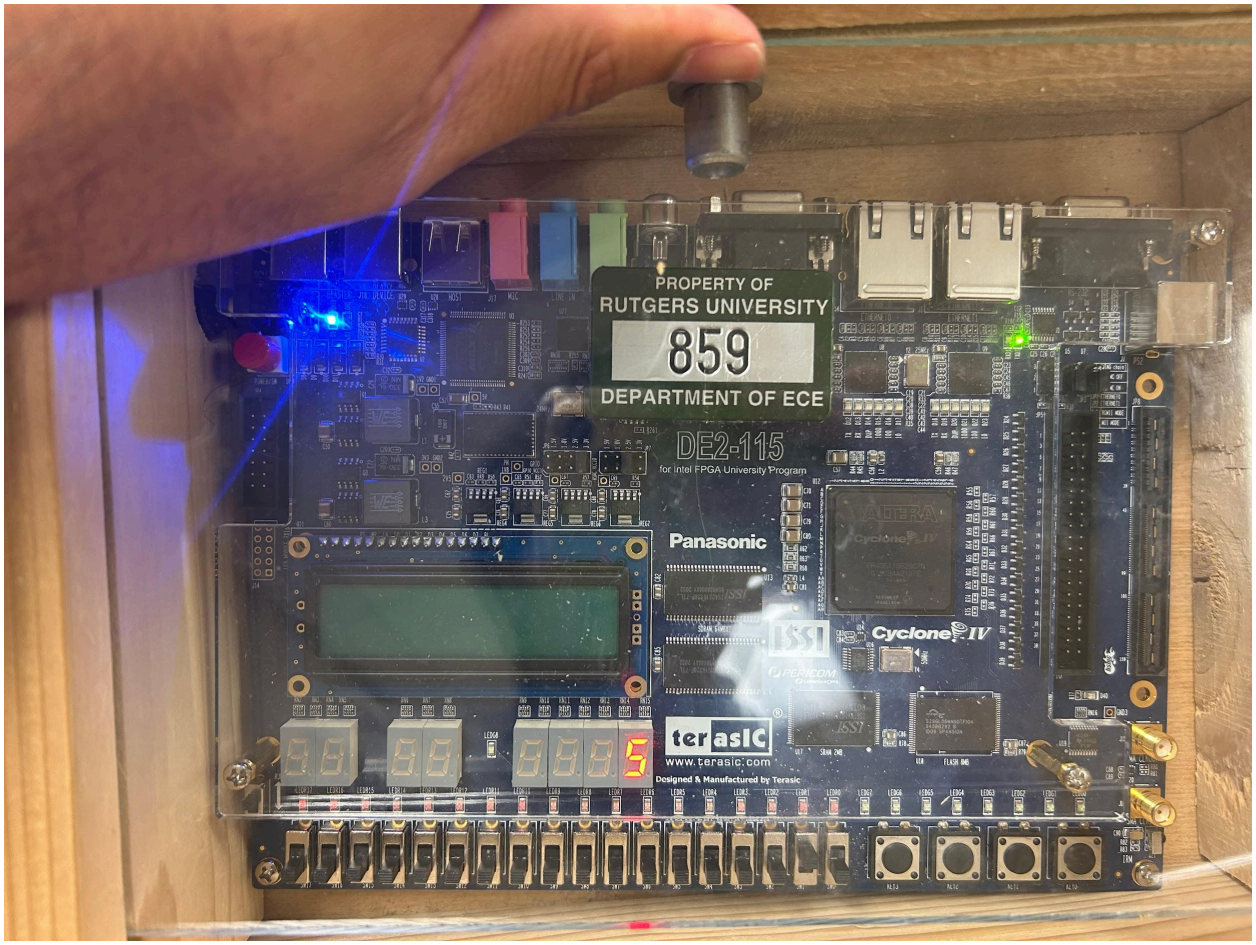
PreLab values



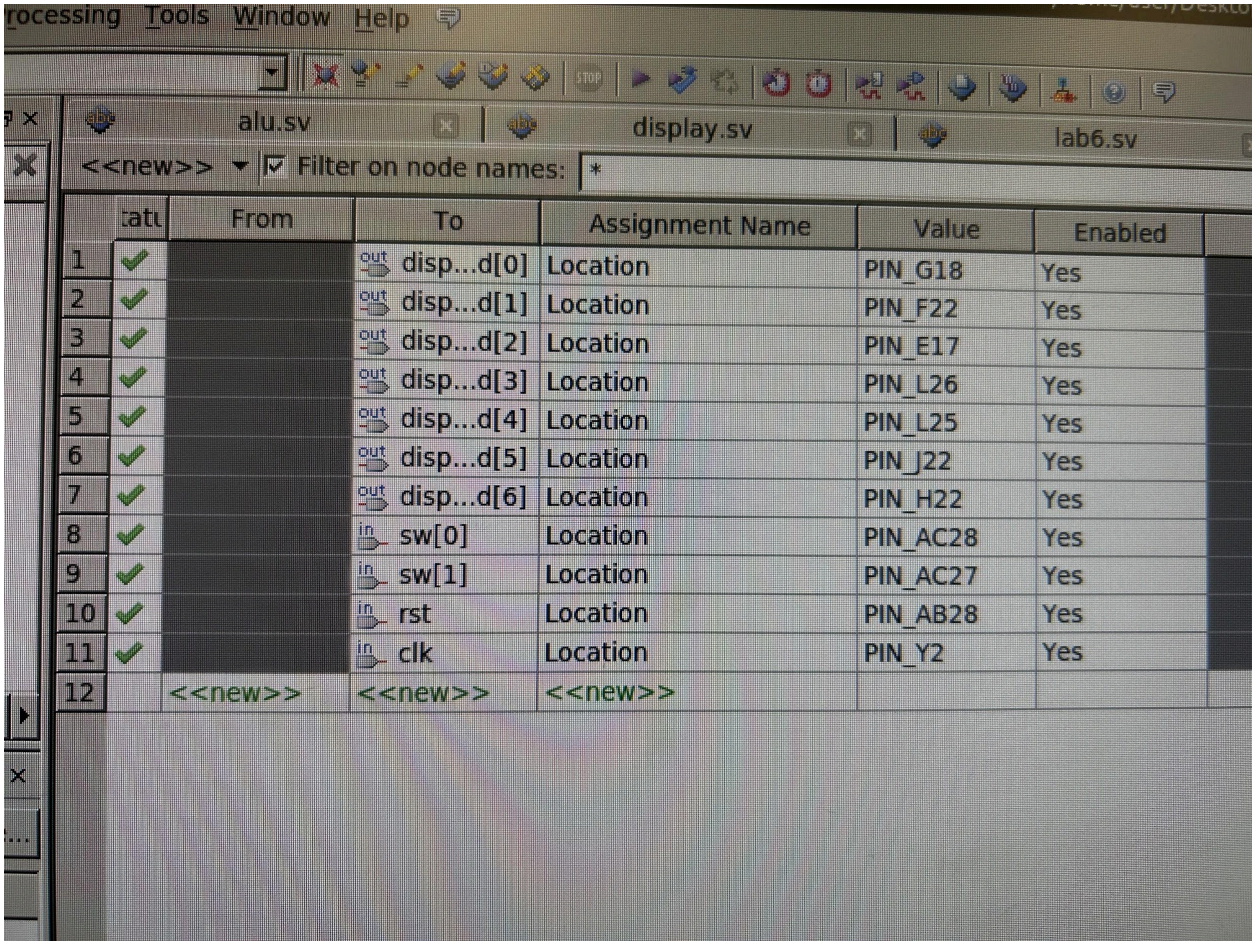
Board number: 514
SW - register file[4] to data memory[2]



LW - register_file[5] → data_memory[1]



Pin assignment:



	tatu	From	To	Assignment Name	Value	Enabled
1	✓		out disp...d[0]	Location	PIN_G18	Yes
2	✓		out disp...d[1]	Location	PIN_F22	Yes
3	✓		out disp...d[2]	Location	PIN_E17	Yes
4	✓		out disp...d[3]	Location	PIN_L26	Yes
5	✓		out disp...d[4]	Location	PIN_L25	Yes
6	✓		out disp...d[5]	Location	PIN_J22	Yes
7	✓		out disp...d[6]	Location	PIN_H22	Yes
8	✓		in sw[0]	Location	PIN_AC28	Yes
9	✓		in sw[1]	Location	PIN_AC27	Yes
10	✓		in rst	Location	PIN_AB28	Yes
11	✓		in clk	Location	PIN_Y2	Yes
12		<<new>>	<<new>>	<<new>>		