Bharat Kathi
ECE 154A
11/22/23

Lab 4 Report

Number of hours spent on lab

I spent about 16 hours on this lab.

Table 1

Cycle	reset	PC	Instr	SrcA	SrcB	ALUResult	Zero	PCSrc	WriteData	MemWrite	Read Data	RegWrite	ImmSrc	ImmExt	PCTarget
1	1	00	`addi x2, x0, 5`	0	5	5	0	0	x	0	x	1	000	5	x
2	0	04	`addi x3, x0, 12`	0	c	c	0	0	x	0	x	1	000	c	x
3	0	08	`addi x7, x3, -9`	c	-9	3	0	0	x	0	x	1	000	-9	x
4	0	0C	`or x4, x7, x2`	3	5	7	0	0	x	0	x	1	x	x	x
5	0	10	`and x5, x3, x4`	c	7	4	0	0	x	0	x	1	x	x	x
6	0	14	`add x5, x5, x4`	4	7	b	0	0	x	0	x	1	x	x	x
7	0	18	`beq x5, x7, end`	b	3	8	0	0	x	0	x	0	010	0x30	0x48
8	0	1C	`slt x4, x3, x4`	c	7	0	1	0	x	0	x	1	x	x	x
9	0	20	`beq x4, x0, around`	0	0	0	1	1	x	0	x	0	010	8	0x28
10	0	28	`slt x4, x7, x2`	3	5	1	0	0	x	0	x	1	x	x	x
11	0	2C	`add x7, x4, x5`	1	b	c	0	0	x	0	x	1	x	x	x
12	0	30	`sub x7, x7, x2`	c	5	7	0	0	x	0	x	1	x	x	x
13	0	34	`sw x7, 84(x3)`	c	0x54	0x60	0	0	7	1	x	0	001	0x54	x
14	0	38	`lw x2, 96(x0)`	0	0x60	0x60	0	0	x	0	7	1	000	0x60	x
15	0	3C	`add x9, x2, x5`	7	b	0x12	0	0	x	0	x	1	x	x	x
16	0	40	`jal x3, end`	x	x	x	x	1	x	0	x	1	011	8	0x48
17	0	48	`add x2, x2, x9`	7	0x12	0x19	0	0	x	0	x	1	x	x	x
18	0	4C	`sw x2, 0x20(x3)`	0x44	0x20	0x64	0	0	0x19	1	x	0	001	0x20	x
19	0	50	`lui x2, 0x0BEEF`	x	x	x	x	0	x	0	x	1	100	0x0beef000	x
20	0	54	`sw x2, 0x24(x3)`	0x44	0x24	0x68	0	0	0x0beef000	1	x	0	001	0x24	x
21	0	58	`beq x2, x2, done`	0x0beef000	0x0beef000	0	1	0	x	0	x	0	010	0	0x58

Simulation Waveforms

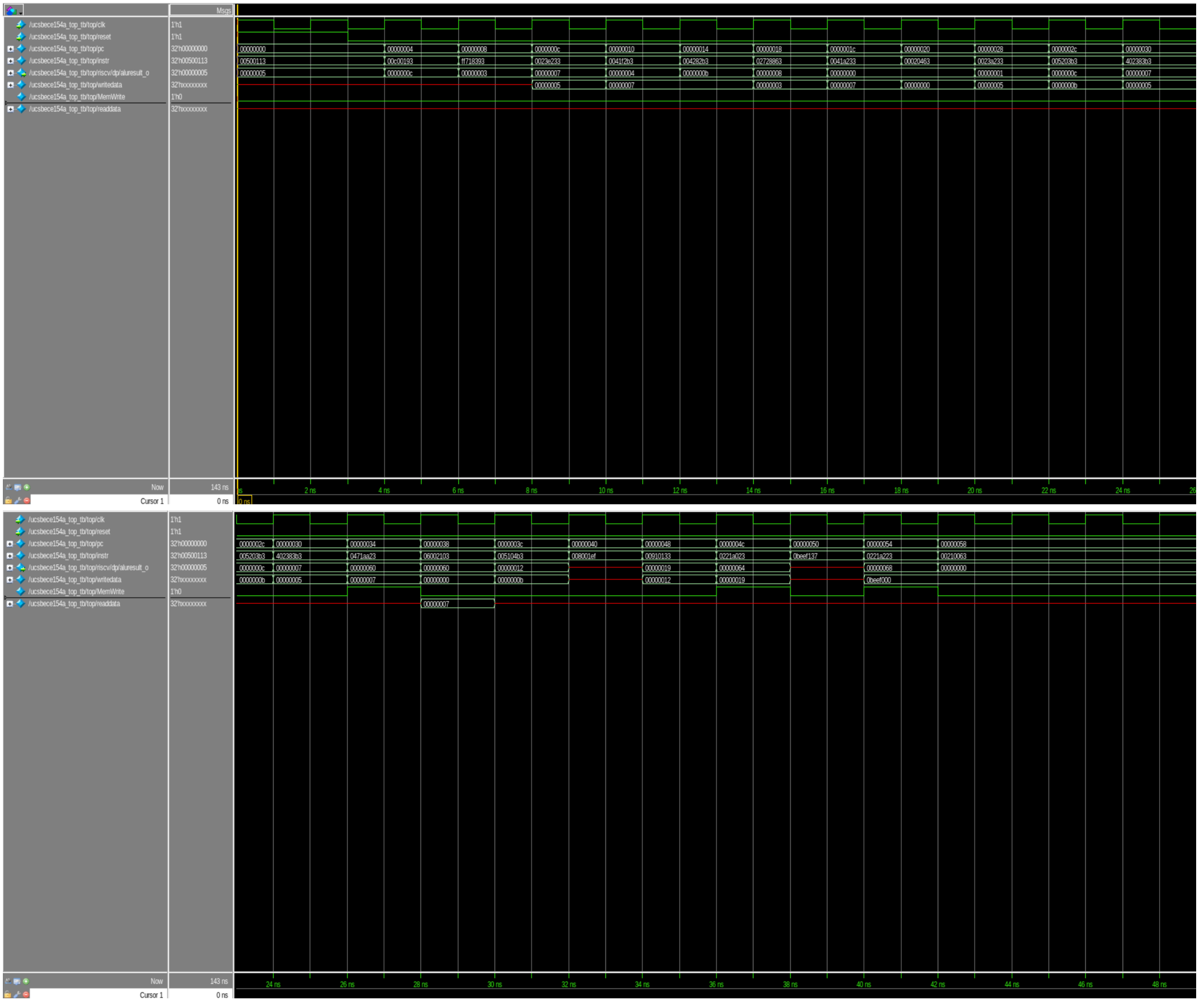
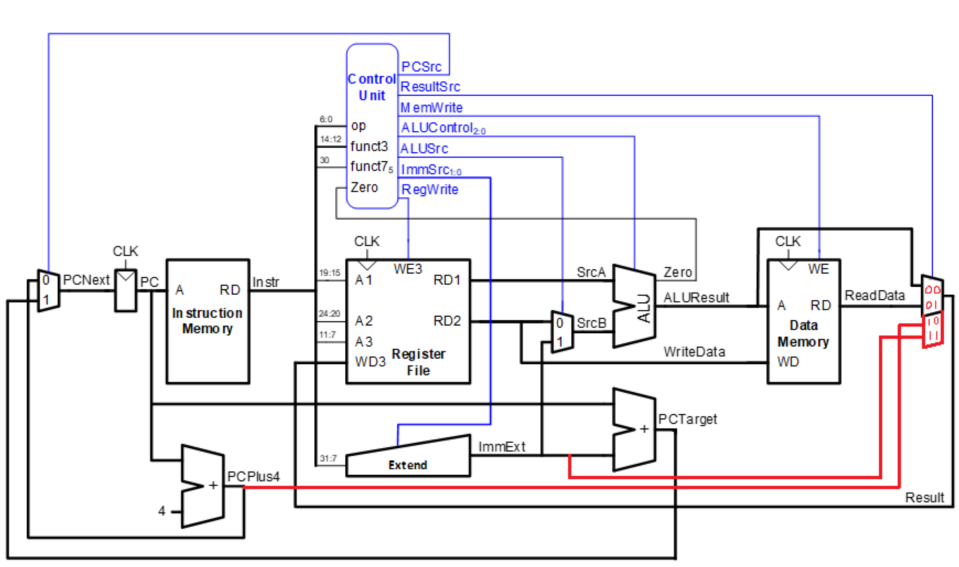


Table 2

op	Instruct.	RegWrite	ImmSrc	ALUSrc	MemWrite	ResultSrc	Branch	ALUOp	Jump
3	lw	1	000	1	0	01	0	00	0
35	sw	0	001	1	1	XX	0	00	0
51	R-type	1	XXX	0	0	00	0	10	0
99	beq	0	010	0	0	XX	1	01	0
19	I-type	1	000	1	0	00	0	10	0
111	jal	1	011	X	0	10	0	XX	1
55	lui	1	100	X	0	11	0	XX	0



ucsbece154a_controller.v

```
// ucsbece154a_controller.v
// All Rights Reserved
// Copyright (c) 2023 UCSB ECE
// Distribution Prohibited


module ucsbece154a_controller (
    input          [6:0] op_i,
    input          [2:0] funct3_i,
```

```
    input               funct7b5_i,
    input           zero_i,
    output wire          RegWrite_o,
    output wire          ALUSrc_o,
    output wire          MemWrite_o,
    output wire    [1:0] ResultSrc_o,
    output reg     [2:0] ALUControl_o,
    output wire          PCSrc_o,
    output wire    [2:0] ImmSrc_o
);


 `include "ucsbece154a_defines.vh"

// TO DO: Generate properly PCSrc by replacing all `z` values with the correct values

wire branch, jump;
assign PCSrc_o = (zero_i & branch) | jump;


//  TO DO: Implement main decoder
//  • Replace all `z` values with the correct values
//  • Extend the code to implement jal and lui

 reg [11:0] controls;
wire [1:0] ALUOp;


assign {RegWrite_o,
   ImmSrc_o,
       ALUSrc_o,
       MemWrite_o,
       ResultSrc_o,
   branch,
   ALUOp,
   jump} = controls;

always @ * begin
  case (op_i)
   instr_lw_op:        controls = 12'b100010010000;
   instr_sw_op:        controls = 12'b000111000000;
   instr_Rtype_op:     controls = 12'b100000000100;
   instr_beq_op:       controls = 12'b001000001010;
   instr_ItypeALU_op:  controls = 12'b100010000100;
   instr_jal_op:       controls = 12'b101100100001;
   instr_lui_op:       controls = 12'b110000110000;

   default: begin
                          controls = 12'b000000000000;
           `ifdef SIM
               $warning("Unsupported op given: %h", op_i);
           `else
           ;
           `endif

       end
  endcase
 end

//  TO DO: Implement ALU decoder by replacing all `z` values with the correct values

 wire RtypeSub;

 assign RtypeSub = funct7b5_i & op_i[5];

 always @ * begin
 case(ALUOp)
   ALUop_mem:                 ALUControl_o = 3'b000;
   ALUop_beq:                 ALUControl_o = 3'b001;
   ALUop_other:
       case(funct3_i)
           instr_addsub_funct3:
                 if(RtypeSub) ALUControl_o = 3'b001;
                 else         ALUControl_o = 3'b000;
           instr_slt_funct3:  ALUControl_o = 3'b101;
           instr_or_funct3:   ALUControl_o = 3'b011;
```

```
            instr_and_funct3:  ALUControl_o = 3'b010;
            default: begin
                               ALUControl_o = 3'b000;
                `ifdef SIM
                    $warning("Unsupported funct3 given: %h", funct3_i);
                `else
                   ;
                `endif
            end
        endcase
    default:
       `ifdef SIM
           $warning("Unsupported ALUop given: %h", ALUOp);
       `else
           ;
       `endif
   endcase
  end

endmodule
```

ucsbece154a_datapath.v

```
// ucsbece154a_datapath.v
// All Rights Reserved
// Copyright (c) 2023 UCSB ECE
// Distribution Prohibited


module ucsbece154a_datapath (
    input               clk, reset,
    input               RegWrite_i,
    input         [2:0] ImmSrc_i,
    input               ALUSrc_i,
    input               PCSrc_i,
    input         [1:0] ResultSrc_i,
    input         [2:0] ALUControl_i,
    output              zero_o,
    output reg   [31:0] pc_o,
    input        [31:0] instr_i,
    output wire  [31:0] aluresult_o, writedata_o,
    input        [31:0] readdata_i
);

`include "ucsbece154a_defines.vh"

/// Your code here

wire [31:0] PCNext, PCPlus4, PCTarget;
reg  [31:0] ImmExt;
wire [31:0] SrcA, SrcB;
reg [31:0] Result;
wire dummy1;
wire dummy2;
assign PCNext = PCSrc_i ? PCTarget : PCPlus4;
assign SrcB = ALUSrc_i ? ImmExt : writedata_o;

always @ * begin
     case (ResultSrc_i)
        2'b00: Result = aluresult_o;
        2'b01: Result = readdata_i;2'b10: Result = PCPlus4;
        2'b11: Result = ImmExt;
        default: Result = 32'b0;
    endcase
end

always @ * begin
     case (ImmSrc_i)
        3'b000: ImmExt = {{20{instr_i[31]}}, instr_i[31:20]};
        3'b001: ImmExt = {{20{instr_i[31]}}, instr_i[31:25], instr_i[11:7]};
        3'b010: ImmExt = {{20{instr_i[31]}}, instr_i[7], instr_i[30:25], instr_i[11:8], 1'b0};
        3'b011: ImmExt = {{12{instr_i[31]}},instr_i[19:12],instr_i[20],instr_i[30:21],1'b0};
        3'b100: ImmExt = {instr_i[31:12],12'b0};
        default: ImmExt = 32'b0;
```

```
    endcase
end

always @(posedge clk or posedge reset) begin
    if (reset) begin
        pc_o <= 32'b0;
    end
    else begin
         pc_o <= PCNext;
     end
end

assign PCPlus4 = pc_o + 32'b100;
assign PCTarget = pc_o + ImmExt;

ucsbece154a_alu ALU (
    .a_i(SrcA),
    .b_i(SrcB),
    .alucontrol_i(ALUControl_i),
    .result_o(aluresult_o),
    .zero_o(zero_o)
);

ucsbece154a_rf rf(
    .clk(clk),
    .a1_i(instr_i[19:15]),
    .a2_i(instr_i[24:20]),
    .a3_i(instr_i[11:7]),
    .rd1_o(SrcA),
    .rd2_o(writedata_o),
    .we3_i(RegWrite_i),
    .wd3_i(Result)
);

endmodule
```