ELC 2137 Lab 09: ALU with Input Register

Yiting Wang

October 28, 2020

Summary

In this lab, we will take the next step by creating an Arithmetic Logic Unit (ALU) capable of a few operations. In order to do mathematical operations with the ALU, we need two numbers. One of these will come from the switches. For the other, we will use a register to store a number.

Q&A

There is no question in the lab 09 assignment.

Results

Firgure 1 is the simulation waveform and ERT of the register.

Time (ns):	0-5	5-10	10-15	15-20	20-25	25-30	30-35	35-40	40-45	45-50	50-55
D (hex)	0	0	A	A	3	3	0	0	0→6	6	6
clk	0	1	0	1	0	1	0	1	0	1	0
en	0	0	1	1	1→0	0→1	1→0	0	0→1	1	1
rst	0	0→1	0	0	0	0	0	0	0	0	0
Q (hex)	X	X→0	0	a	a	a	a	a	a	6	6

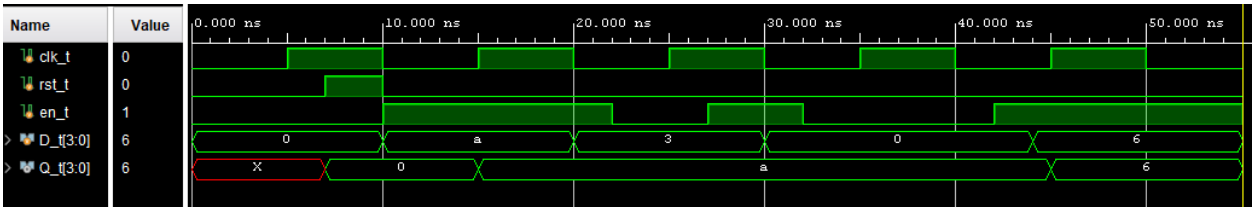


Figure 1: the simulation waveform and ERT of the register

Firgure 2 is the simulation waveform and ERT of the ALU.

This is the picture of a copy of the relevant TCL Console output for Top-Level Simulation.

Time (ns):	0-10	10-20	20-30	30-40	40-50	50-60
in0	1101	1101	1101	1101	1101	1101
in1	1010	1010	1010	1010	1010	1010
op	0000	0001	0010	0011	0100	0101
out	0111	0011	1000	1111	0111	1101

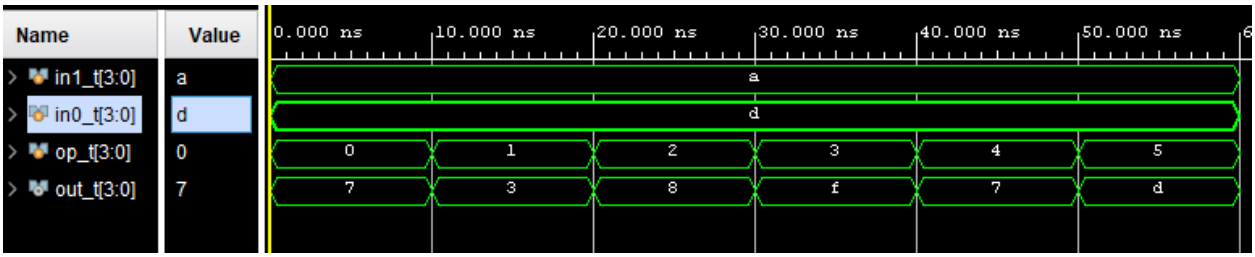
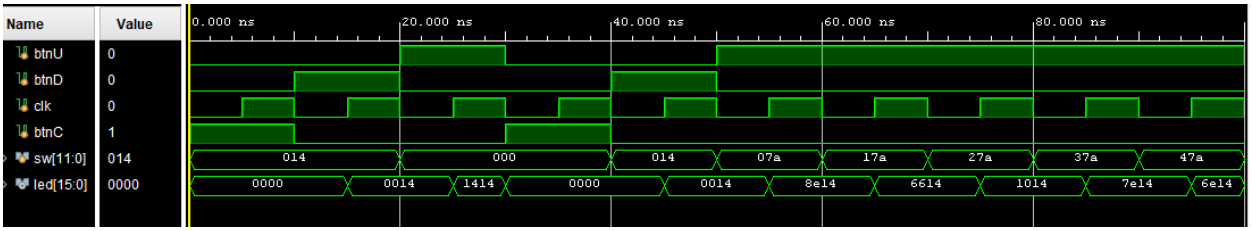


Figure 2: the simulation waveform and ERT of the ALU



Code

File Inclusion

Listing 1: mux4 Verilog code

```
‘timescale 1ns / 1ps
//
   //////////////////////////////////////////////////////////////////////////////////

// Company:
// Engineer:
//
// Create Date: 10/22/2020 11:26:33 AM
// Design Name:
// Module Name: register
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
```

```
//
   //////////////////////////////////////////////////////////////////////////////////



module register #(parameter N=1)(
    input clk,
    input rst,
    input en,
    input [N-1:0] D,
    output reg [N-1:0] Q
    );

    always @(posedge clk, posedge rst)
    begin
        if (rst == 1)
            Q <= 0;
        else if (en == 1)
            Q <= D;
    end

endmodule
```

File Inclusion

Listing 2: mux4 Test Benches Verilog code

```
‘timescale 1ns / 1ps
//
   //////////////////////////////////////////////////////////////////////////////////

// Company:
// Engineer:
//
// Create Date: 10/22/2020 11:32:06 AM
// Design Name:
// Module Name: register_test
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//
   //////////////////////////////////////////////////////////////////////////////////



module register_test();
```

```
    reg clk_t, rst_t, en_t;
    reg [3:0] D_t;
    wire [3:0] Q_t;

    register #(.N(4)) r(
        .clk(clk_t), .rst(rst_t), .en(en_t),
        .D(D_t), .Q(Q_t)
    );

    always begin
        clk_t = ~clk_t; #5;
    end

    initial begin
        clk_t = 0; en_t = 0; rst_t = 0; D_t = 4'h0; #7;
        rst_t = 1; #3;
        D_t = 4'ha; en_t = 1; rst_t = 0; #10;
        D_t = 4'h3; #2;
        en_t = 0; #5;
        en_t = 1; #3;
        D_t = 4'h0; #2;
        en_t = 0; #10;
        en_t = 1; #2;
        D_t = 4'h6; #11;
        $finish;
    end

endmodule
```

File Inclusion

Listing 3: anode decoder Verilog code

```
`timescale 1ns / 1ps
//
   //////////////////////////////////////////////////////////////////////////////////

// Company:
// Engineer:
//
// Create Date: 10/22/2020 11:55:09 AM
// Design Name:
// Module Name: alu
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
```

```
//
//
    //////////////////////////////////////////////////////////////////////////////////



module alu #(parameter N=8)(
    input [N-1:0] in1,
    input [N-1:0] in0,
    input [3:0] op,
    output reg [N-1:0] out
    );

    parameter ADD = 0;
    parameter SUB = 1;
    parameter AND = 2;
    parameter OR = 3;
    parameter XOR = 4;

    always @*
    begin
        case(op)
            ADD: out = in0 + in1;
            SUB: out = in0 - in1;
            AND: out = in0 & in1;
            OR: out = in0 | in1;
            XOR: out = in0 ^ in1;
            default: out = in0;
        endcase
    end

endmodule
```

File Inclusion

Listing 4: anode decoder Test Benches Verilog code

```
`timescale 1ns / 1ps
//
    //////////////////////////////////////////////////////////////////////////////////

// Company:
// Engineer:
//
// Create Date: 10/22/2020 12:13:02 PM
// Design Name:
// Module Name: alu_test
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
```

```
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//
    //////////////////////////////////////////////////////////////////////////////////



module alu_test();

    reg [3:0] in1_t, in0_t, op_t;
    wire [3:0] out_t;

    alu #(.N(4)) a (
        .in1(in1_t), .in0(in0_t), .op(op_t),
        .out(out_t)
    );

    initial begin
        in1_t = 4'ha; in0_t = 4'hd; op_t = 4'h0; #10;
        op_t = 4'h1; #10;
        op_t = 4'h2; #10;
        op_t = 4'h3; #10;
        op_t = 4'h4; #10;
        op_t = 4'h5; #10;
        $finish;
    end

endmodule
```

File Inclusion

Listing 5: sseg4 Verilog code

```
`timescale 1ns / 1ps
//
    //////////////////////////////////////////////////////////////////////////////////

// Company:
// Engineer:
//
// Create Date: 10/22/2020 12:40:57 PM
// Design Name:
// Module Name: top_lab9
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
```

```
// Additional Comments:
//
//
    //////////////////////////////////////////////////////////////////////////////////



module top_lab9(
    input btnU,
    input btnD,
    input [11:0] sw,
    input clk,
    input btnC,
    output [15:0] led
    );

    wire [7:0] Q_reg1, out_alu;

    register #(.N(8)) reg1(
        .D(sw[7:0]), .en(btnD), .clk(clk), .rst(btnC),
        .Q(Q_reg1)
    );

    alu #(.N(8)) alu0 (
        .in1(Q_reg1), .in0(sw[7:0]), .op(sw[11:8]),
        .out(out_alu)
    );

    register #(.N(8)) reg2(
        .D(out_alu), .en(btnU), .clk(clk), .rst(btnC),
        .Q(led[15:8])
    );

    assign led[7:0] = Q_reg1;

endmodule
```