ELC 2137 Lab 10: 7-segment Display with Time-Division Multiplexing

Maddie Vorhies

November 4, 2020

Summary

The goal of this lab was to use lab 8 and lab 9 and put them together to create a 7-segment display with time- division multiplexing. In order to do this, I needed the top9 module from lab 9 and I needed the sseg4 from lab 8. In addition to it, I also created a counter and a show2c. The counter has a timer that is used to get a short pulse when the counter reaches its maximum value. The show2c is used to output the 2's compliment of the incoming number. To check the counter and the show2c, I created a testbench. Once the testbench was successfully executed, I was able to combine the top9 from lab 9, the sseg4 from lab 8, and the counter and show2c from lab 10 to create a mini calculator. The calculator is able to add and subtract. If needed, it will display a negative number on the board. I tested my board to make sure that it worked and I was successfully able to output both a positive and a negative number.

Results

Figure 1: Simulation Waveform for counter (N = 4)

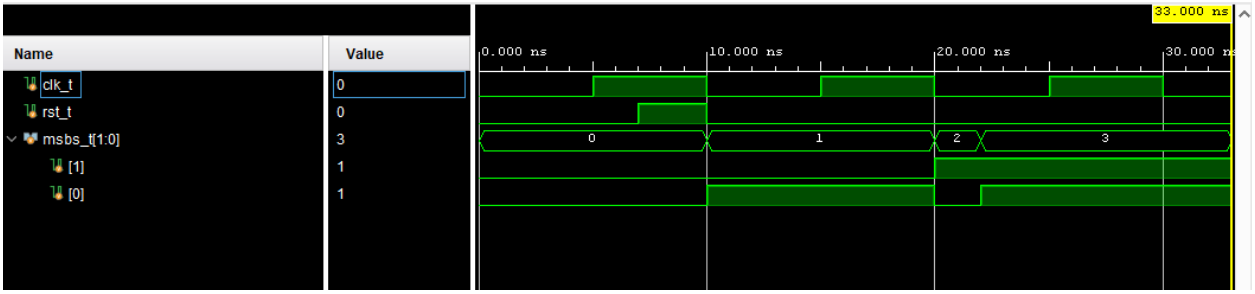


Figure 2: Simulation Waveform for show2c

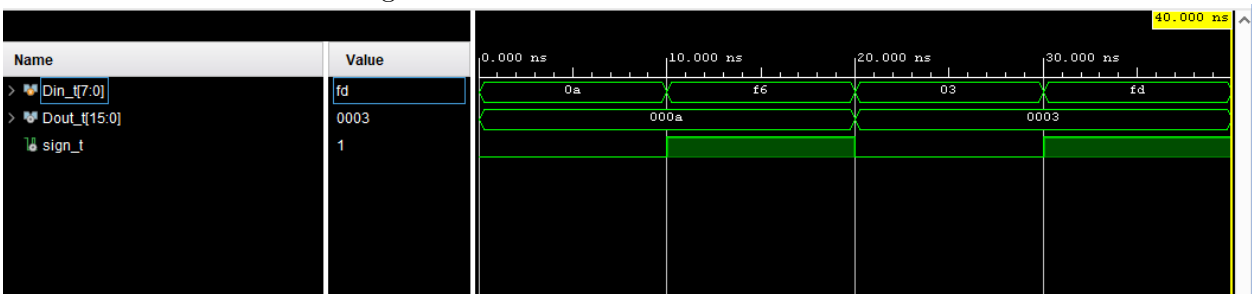
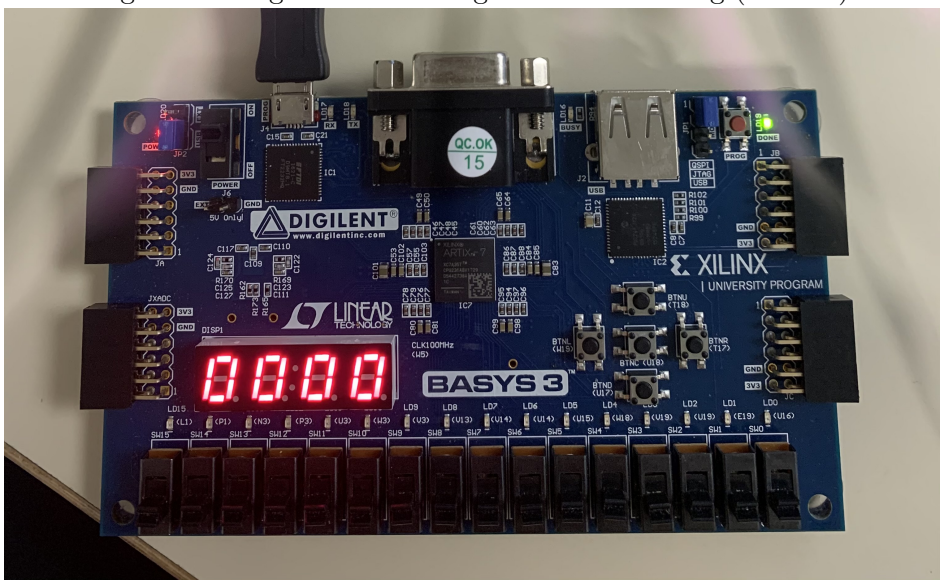
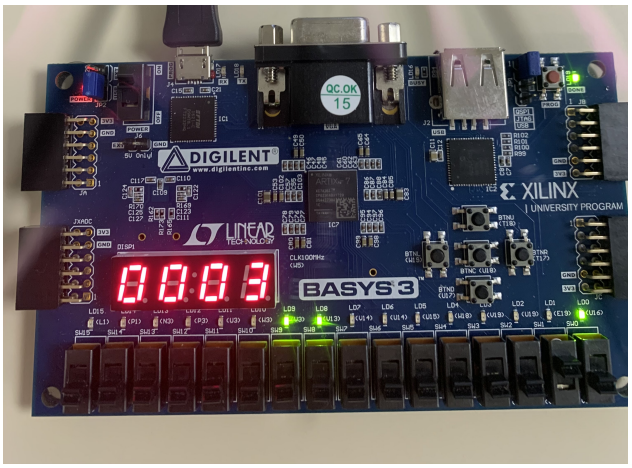


Figure 3: 4-digits at full strength and no blinking (N = 20)



Positive number



Negative Number

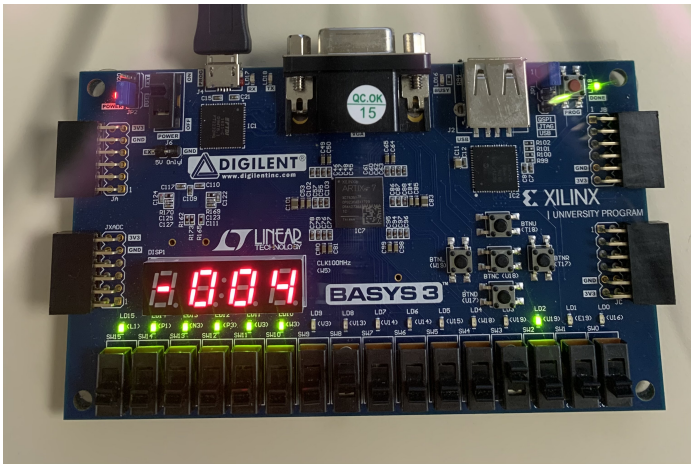


Table 1: Board Pictures

Code

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

// Company:
// Engineer:
//
// Create Date: 10/29/2020 11:58:01 AM
// Design Name:
// Module Name: counter
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//
    //////////////////////////////////////////////////////////////////////////////////



module counter #(parameter N = 1)
    (
    input clk, rst,
    output [1:0] msbs
    );

    wire [N-1:0] Q_reg, Q_next;
    assign Q_next = Q_reg + 1;

    register #(.N(N)) reg_in_count (
       .D(Q_next),
       .en(1),
       .clk(clk),
       .rst(rst),
       .Q(Q_reg)
    );

    assign msbs = Q_reg[N-1:N-2];

endmodule
```

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

// Company:
```

```
// Engineer:
//
// Create Date: 10/29/2020 12:16:13 PM
// Design Name:
// Module Name: counter_test
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//
   //////////////////////////////////////////////////////////////////////////////////



module counter_test();

    reg clk_t, rst_t;
    reg [1:0] msbs_t;

    counter #(.N(4)) dut (
       .clk(clk_t),
       .rst(rst_t),
       .msbs(msbs_t)
    );

    always begin
       clk_t = ~clk_t; #5;
    end

    initial begin
        clk_t = 0; msbs_t = 2'b00; rst_t = 0; #7;
        rst_t = 1; #3;
        msbs_t = 2'b01; rst_t = 0; #10;
        msbs_t = 2'b10;            #2;
        msbs_t = 2'b11;            #11;
        $finish;
    end



endmodule
```

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

// Company:
```

```
// Engineer:
//
// Create Date: 10/29/2020 12:49:23 PM
// Design Name:
// Module Name: show_2c
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//
   //////////////////////////////////////////////////////////////////////////////////



module show_2c(
    input [7:0] Din,
    output [15:0] Dout,
    output sign
    );

    wire [7:0] mux_out;
    wire [7:0] in1;

    assign in1 = ~Din[7:0] + 1;

    mux2 #(.BITS(8)) mu2_3 (
       .in0(Din[7:0]),
       .in1(in1),
       .sel(Din[7]),
       .out(mux_out)
    );

    assign Dout[15:0] = {8'b0, mux_out};
    assign sign = Din[7];


endmodule
```

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

// Company:
// Engineer:
//
// Create Date: 11/04/2020 06:29:34 PM
// Design Name:
```

```
// Module Name: show_2c_test
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//
   //////////////////////////////////////////////////////////////////////////////////



module show_2c_test();

    reg [7:0] Din_t;
    wire [15:0] Dout_t;
    wire sign_t;

    show_2c dut (
       .Din(Din_t),
       .Dout(Dout_t),
       .sign(sign_t)
    );

    initial begin
       Din_t = 10; #10;
       Din_t = -10; #10;
       Din_t = 3; #10;
       Din_t = -3; #10;
       $finish;
    end

endmodule
```

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

// Company:
// Engineer:
//
// Create Date: 10/29/2020 12:36:36 PM
// Design Name:
// Module Name: wrapper
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
```

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



module wrapper(
    input clk,
    input btnC,
    output [6:0] seg,
    output dp,
    output [3:0] an
    );

    wire [1:0] msbs_out;

    counter #(.N(20)) counter0 (
       .clk(clk),
       .rst(btnC),
       .msbs(msbs_out)
    );

    sseg4 sseg4_0 (
       .data(16'b0),
       .sign(0),
       .hex_dec(0),
       .digit_sel(msbs_out),
       .seg(seg),
       .dp(dp),
       .an(an)
    );

endmodule
```

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

// Company:
// Engineer:
//
// Create Date: 11/04/2020 06:44:25 PM
// Design Name:
// Module Name: top_lab10
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
```

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



module top_lab10(
    input btnU,
    input btnD,
    input [15:0] sw,
    input clk,
    input btnC,
    output [6:0] seg,
    output dp,
    output [3:0] an,
    output [15:0] led
    );

    wire [15:0] show_out;
    wire sign_out;
    wire [1:0] counter_out;
    wire [15:0] calc_out;

    top_lab9 top0 (
        .btnU(btnU),
        .btnD(btnD),
        .btnC(btnC),
        .sw(sw),
        .clk(clk),
        .led(calc_out)
     );

     show_2c show0 (
        .Din(calc_out[15:8]),
        .Dout(show_out[15:0]),
        .sign(sign_out)
     );

     counter #(.N(20)) counter0 (
        .clk(clk),
        .rst(btnC),
        .msbs(counter_out[1:0])
     );

     sseg4 sseg4_0 (
        .data(show_out[15:0]),
        .sign(sign_out),
        .hex_dec(sw[15]),
        .digit_sel(counter_out[1:0]),
```

```
        .seg(seg[6:0]),
        .dp(dp),
        .an(an[3:0])
    );

    assign led = calc_out;


endmodule
```