

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

Xingpeng Yi

April 14, 2020

## Summary

In this lab, students are required to design a calculator that operates by flipping switches and displays by the seven segment display on the basys 3 board. After finishing the lab, students are able to recognize synchronous design methodology for regular sequential circuits, develop a parameterized counter-timer module, and implement a clock driven, 4-digit display using multiple instances of your counter module. Students are required to design the sseg4\_TDM module. The module is similar to the previous module used in previous labs and needs to be slightly revised. The top level module uses the top level module from last lab and connects it with the mid-level module in this lab to make the display work on the board.

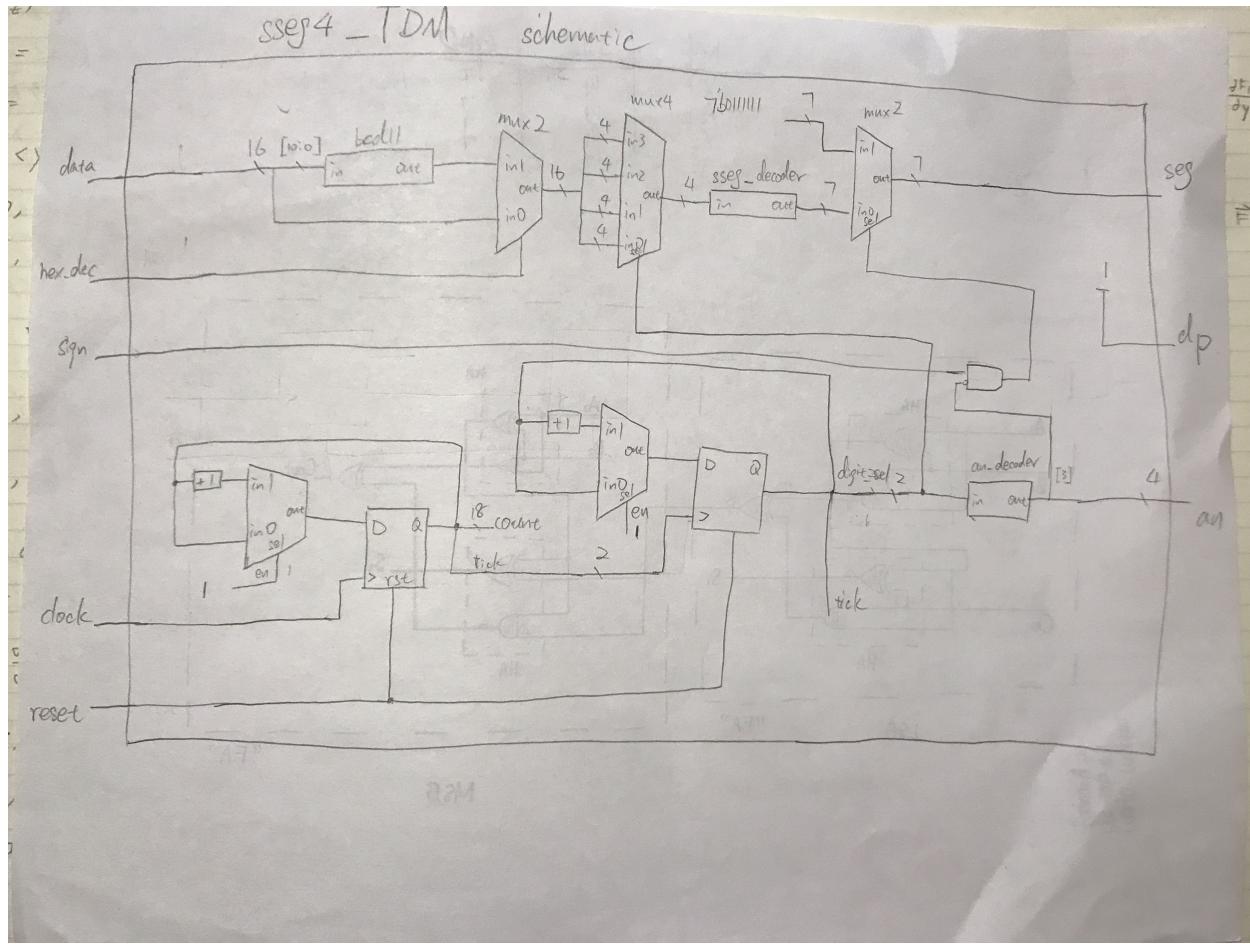


Figure 1: Sseg4\_TDM schematic.

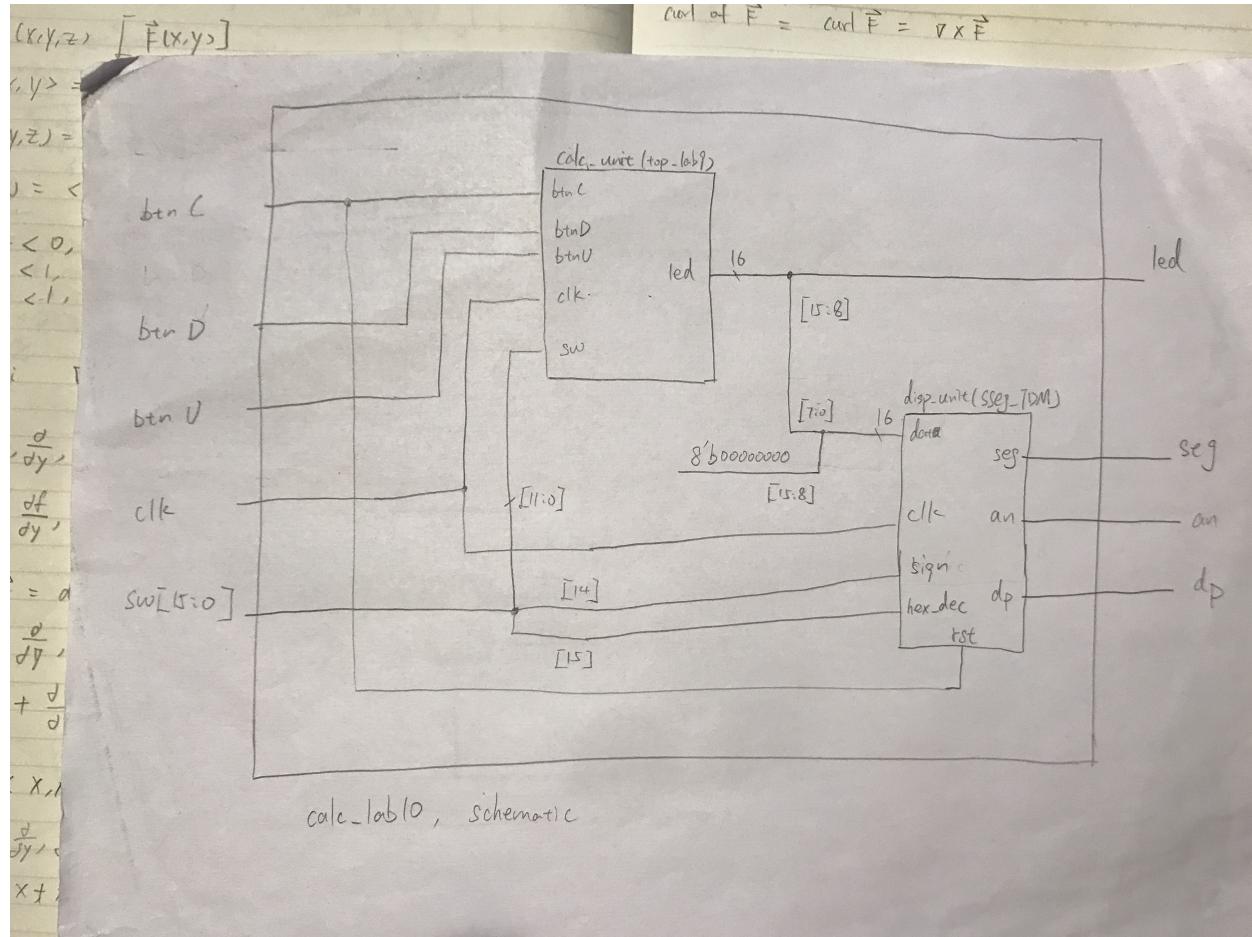


Figure 2: Calc\_lab10 schematic.

## Q&A

- What are the three main “groups” of the RTL definition of sequential logic?

The set of registers in the system. The operations that are performed on the data stored in the registers. The control that supervises the sequence of operations in the system.

- Copy Figure 10.3b onto your own paper (or do it electronically) and draw three boxes around the components that belong to each group. Include your annotated figure in your report.

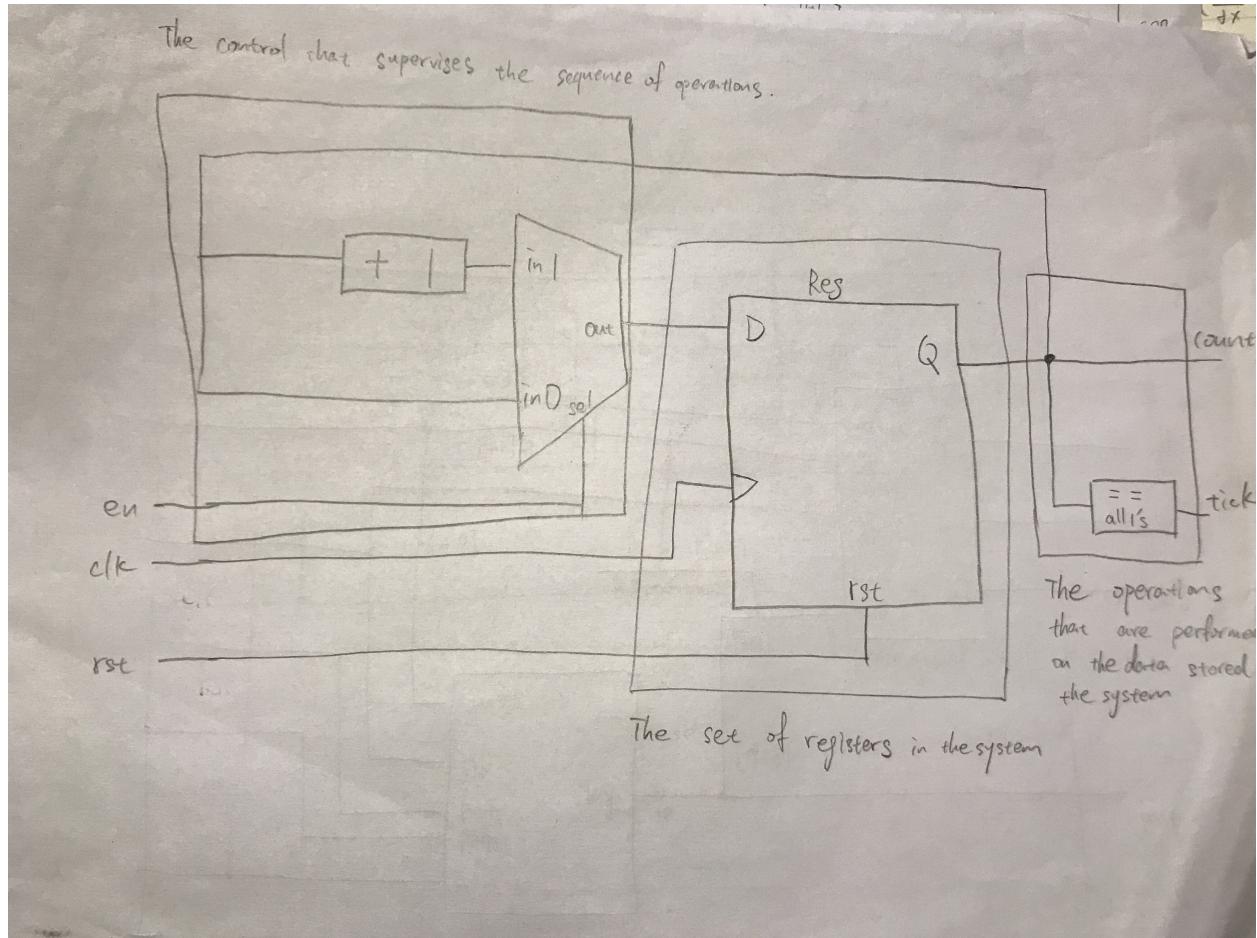


Figure 3: Three groups with the figure.

- If instead of a counter, you wanted to make a shift register that moved the input bits from right to left (low to high). What would you put on the line  $Q \text{ next} = /*????*/$ ?

$$Q_{\text{next}} = Q_{\text{reg}} - 1'b1;$$

## Results

Time (ns):	0	5	10	15	20	25	30	35	40	45	50	55	60	65	70	75
clk	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
en	0	1	1	1	1	0	0	1	1	0	0	1	1	0	0	0
rst	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
count	X	0	0	1	1	2	2	2	3	3	3	3	4	4	4	4
s	X	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0



Figure 4: counter simulation waveform and ERT

Time (ns):	0-20	20 - 60	60 - 140	140 - 300	300-460	460 -780	780 -1000
data	2	1	2	2	3	3	4
hex_dec	0	1	0	1	0	1	0
sign	0	0	0	0	0	0	0
count	23	79	24	24	30	30	19
an	X	X	X	X	e	e	e
dp	1	1	1	1	1	1	1

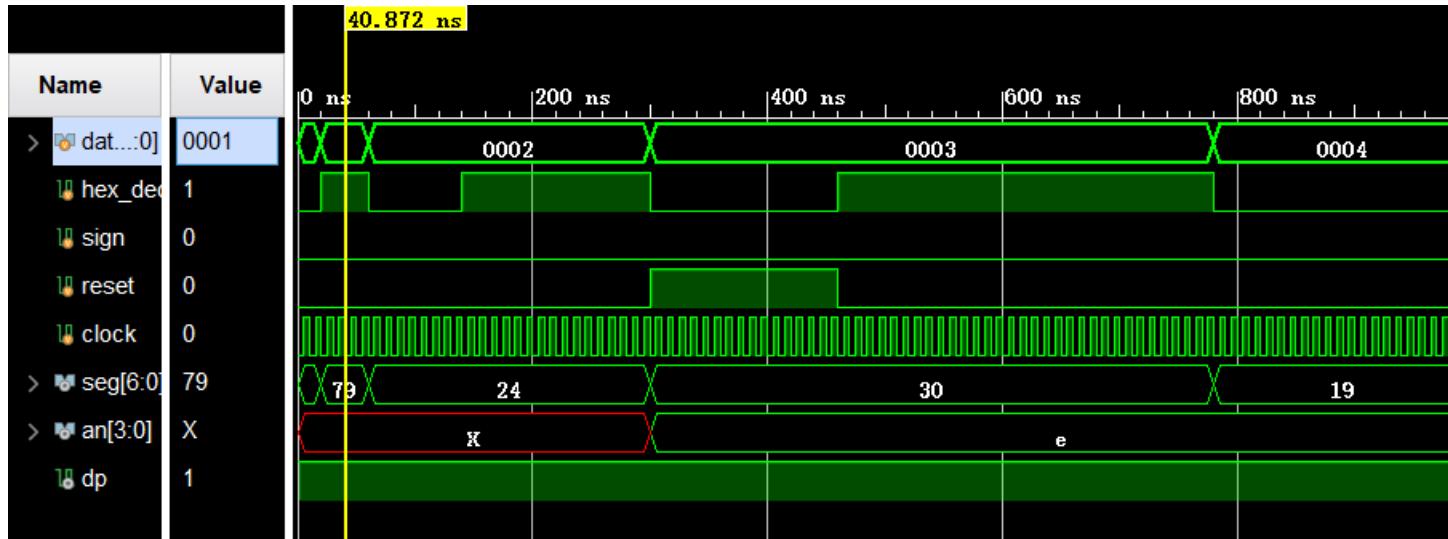


Figure 5: sseg4\_TDM simulation waveform and ERT

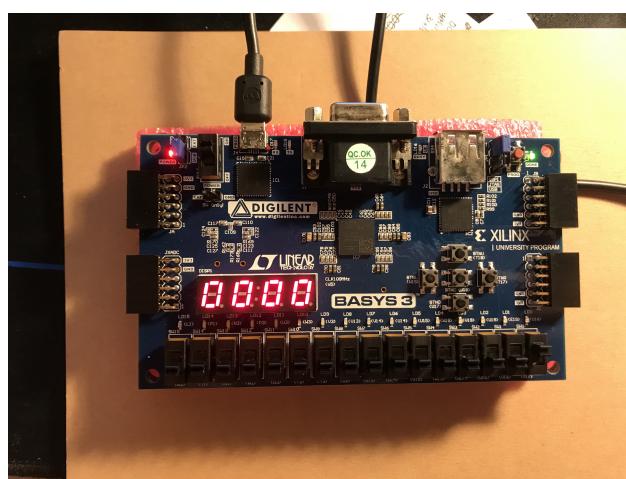


Figure 6: Board display 1.

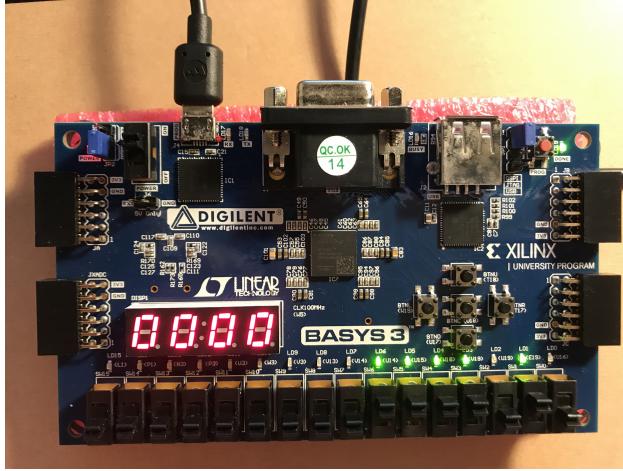


Figure 7: Board display.

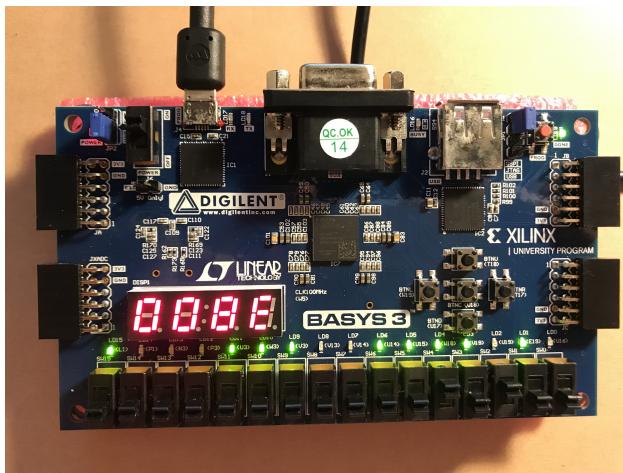


Figure 8: Board display 3.

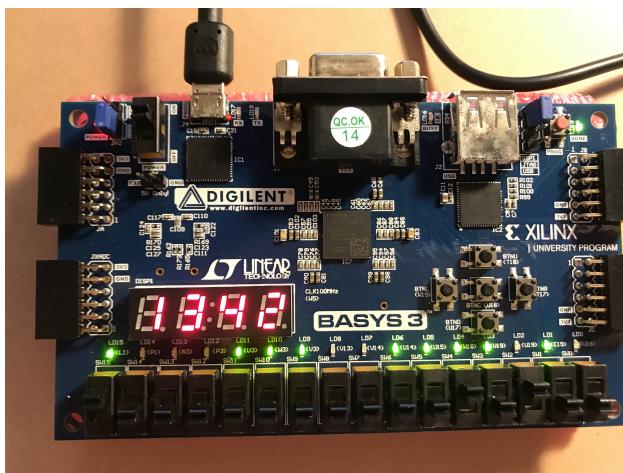


Figure 9: Board display 4.

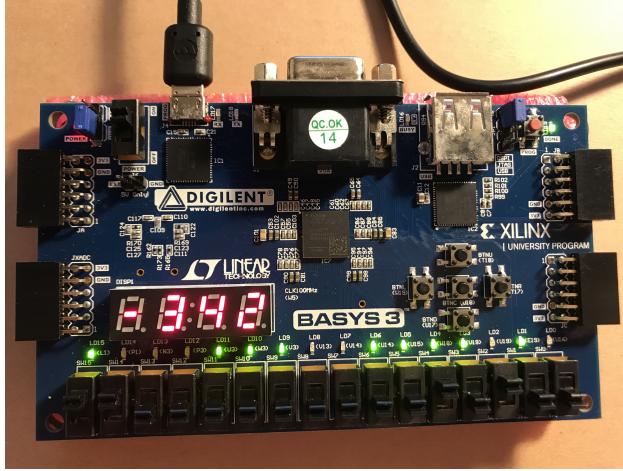


Figure 10: Board display 5.

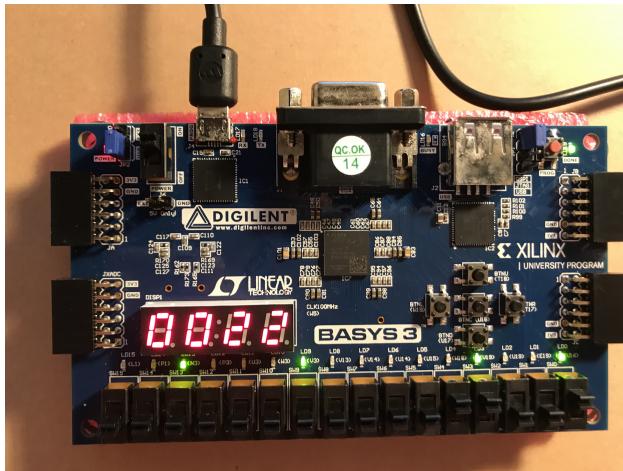


Figure 11: Board display 6.

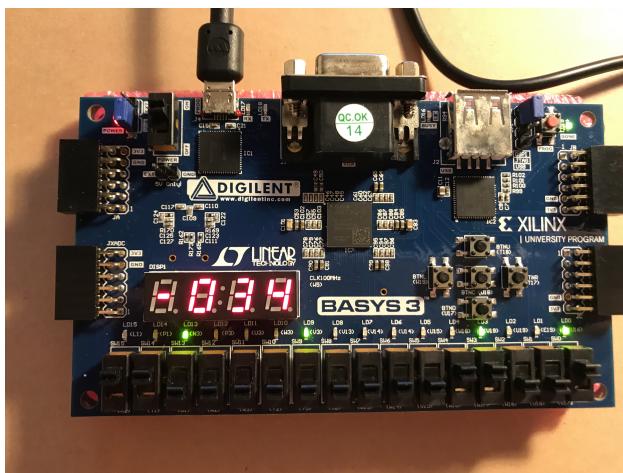


Figure 12: Board display 7.

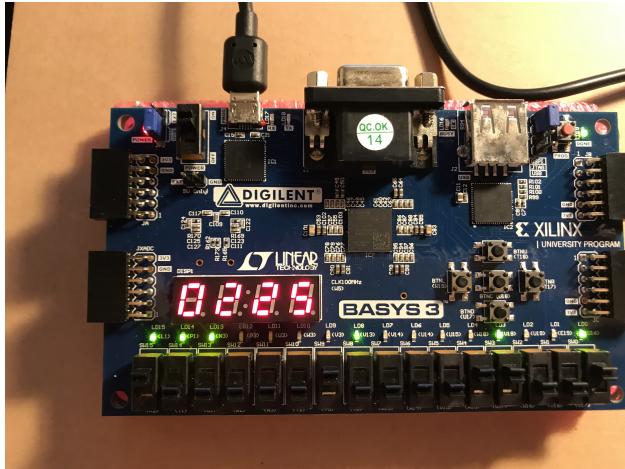


Figure 13: Board display 8.

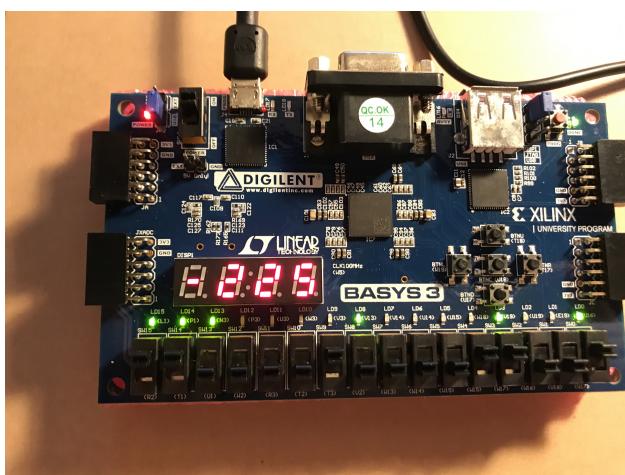


Figure 14: Board display 9.

## Code

Listing 1: counter module source file

```
module counter #(parameter N=1)
(
    input clk, rst, en,
    output [N-1:0] count,
    output tick
);

reg [N-1:0] Q_reg, Q_next;

always @ (posedge clk, posedge rst)
begin
    if(rst)
        Q_reg <= 0;
    else
        Q_reg <= Q_next;
end

always @*
begin
    if(en)
        Q_next = Q_reg + 1'b1;
    else
        Q_next = Q_reg;
end

assign count = Q_reg;
assign tick = (Q_reg == {N{1'b1}}) ? 1'b1 : 1'b0;

endmodule
```

Listing 2: sseg4 TDM module file

```
module sseg4_TDM(
    input [15:0] data,
    input hex_dec,
    input sign,
    input reset,
    input clock,
    output [6:0] seg,
    output [3:0] an,
    output dp
);
    wire [15:0] bcd_o, mux2_1;
    wire [3:0] mux4_o;
    wire [6:0] sseg_1;
    wire [1:0] digit_sel;
    reg [6:0] mux2_in1;
    wire [1:0] tick;
    wire and1;
```

```

assign dp = 1;
assign mux2_in1 = 7'b0111111;
counter #(N(18)) timer(.clk(clock), .en(1), .rst(reset), .tick(tick),
    .count());
counter #(N(2)) counter2(.clk(clock), .en(tick), .rst(reset), .tick()
    , .count(digit_sel));
bcd11 bcd(.B11(data[10:0]), .011(bcd_o));
mux2 #(N(16)) mux1(.hex_dec(hex_dec), .in1(bcd_o), .in0(data), .out(
    mux2_1));
mux4 #(N(4)) mux4_1(.in3(mux2_1[15:12]), .in2(mux2_1[11:8]), .in1(
    mux2_1[7:4]), .in0(mux2_1[3:0]), .digit_sel(digit_sel), .out(mux4_o
));
sseg_decoder Sd (.num(mux4_o), .ssegs(sseg_1));
AnodeDec Ad(.in(digit_sel), .out(an));
assign and1 = sign && (~an[3]);
mux2 #(N(7)) mux2_2(.hex_dec(and1), .in1(mux2_in1), .in0(sseg_1), .out(
    seg));
endmodule

```

---

Listing 3: calc lab10 module file

```

module calc_lab10(
    input [15:0] sw,
    input clk, btnC, btnD, btnU,
    output reg [15:0] led,
    output [6:0] seg,
    output [3:0] an,
    output dp
);
reg [15:0] data;

top_lab9 calc_unit(.sw(sw[11:0]), .clk(clk), .btnC(btnC), .btnD(btnD),
    .btnU(btnU), .led(led));

assign data[7:0] = led[15:8];
assign data[15:8] = 8'b00000000;

sseg4_TDM disp_unit(.data(data), .hex_dec(sw[15]), .sign(sw[14]), .
    reset(btnC), .clock(clk), .seg(seg), .an(an), .dp(dp));
endmodule

```

---