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

Mariah Montgomery

November 4, 2020

Summary

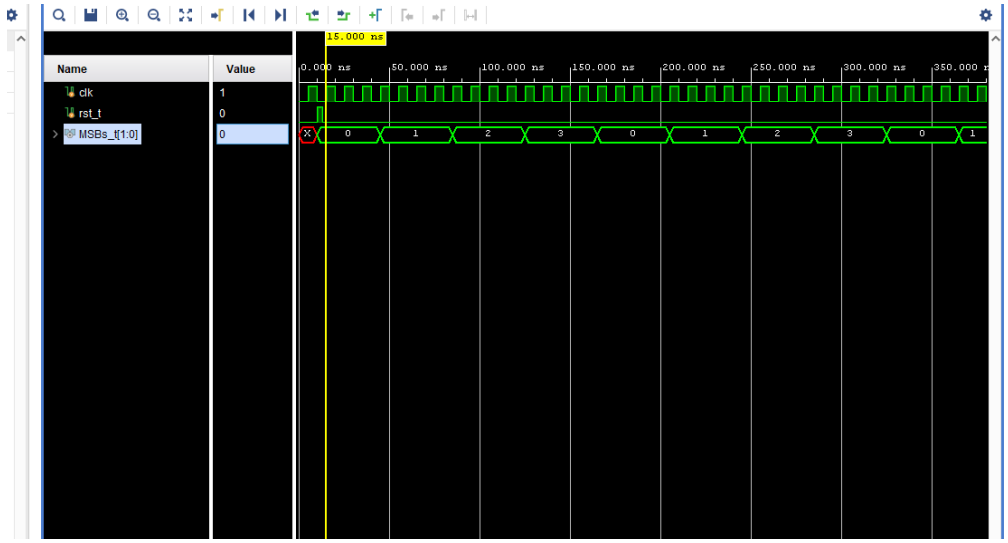
We expanded on lab nine in this lab and made a small calculator using the switches, buttons, LEDs, and seven-segment display on a Basys board. We first created a counter that counts to four so quickly that it lights each digit on the seven segment display without flickering. Next, we created a two's complement source that converted a number to two's complement so that our final calculator could perform simple mathematical calculations. Lastly, we combined modules from our past labs and the two we created for lab 10 to complete our calculator.

Q&A

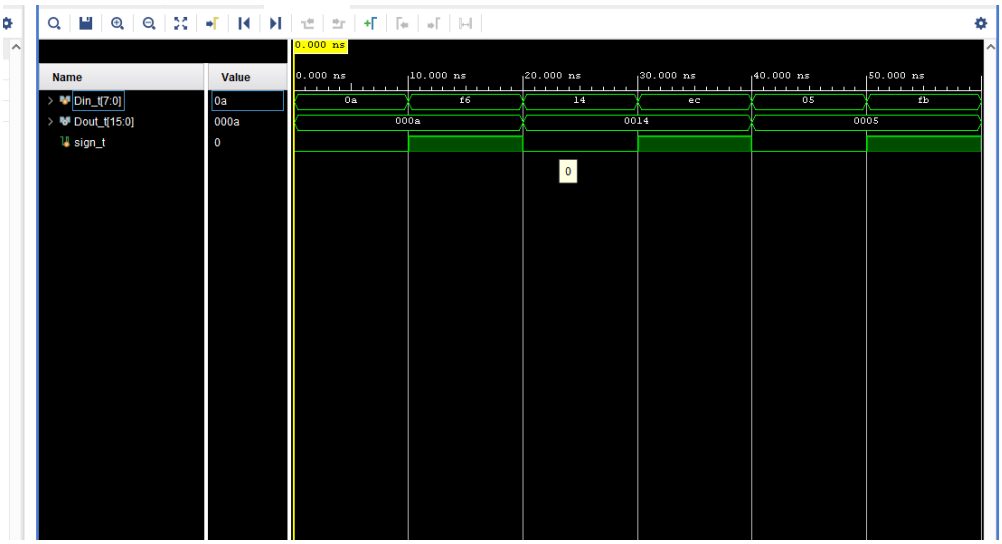
1. The N value that accomplished the full strength display was N=20.

Results

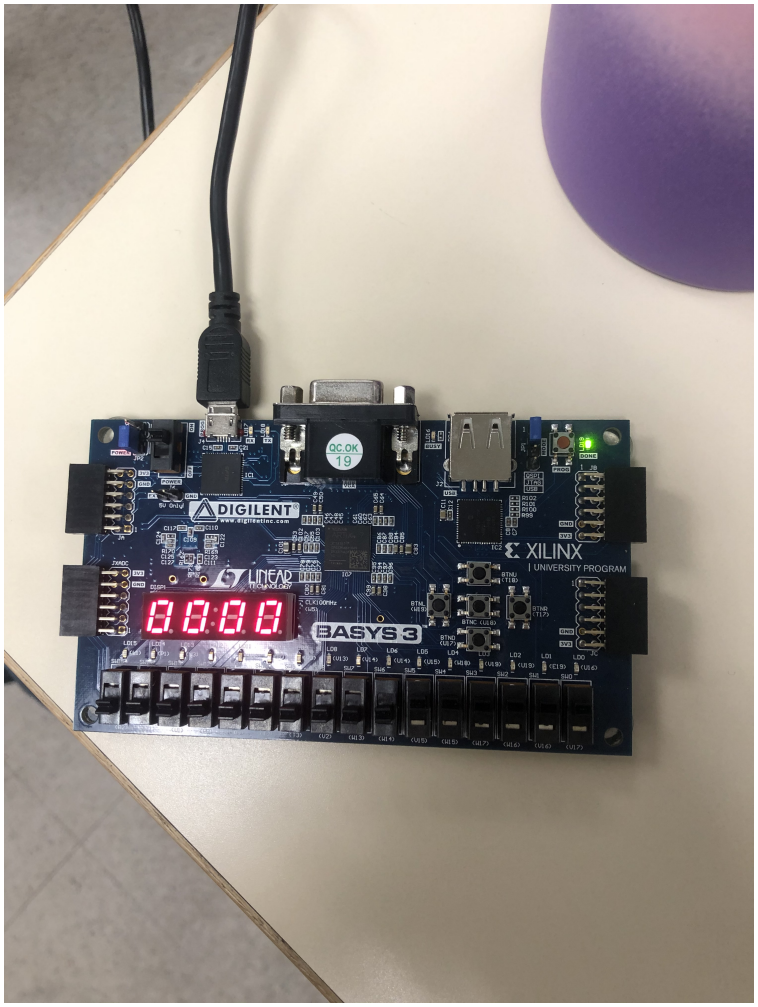
1. Counter Test Simulation



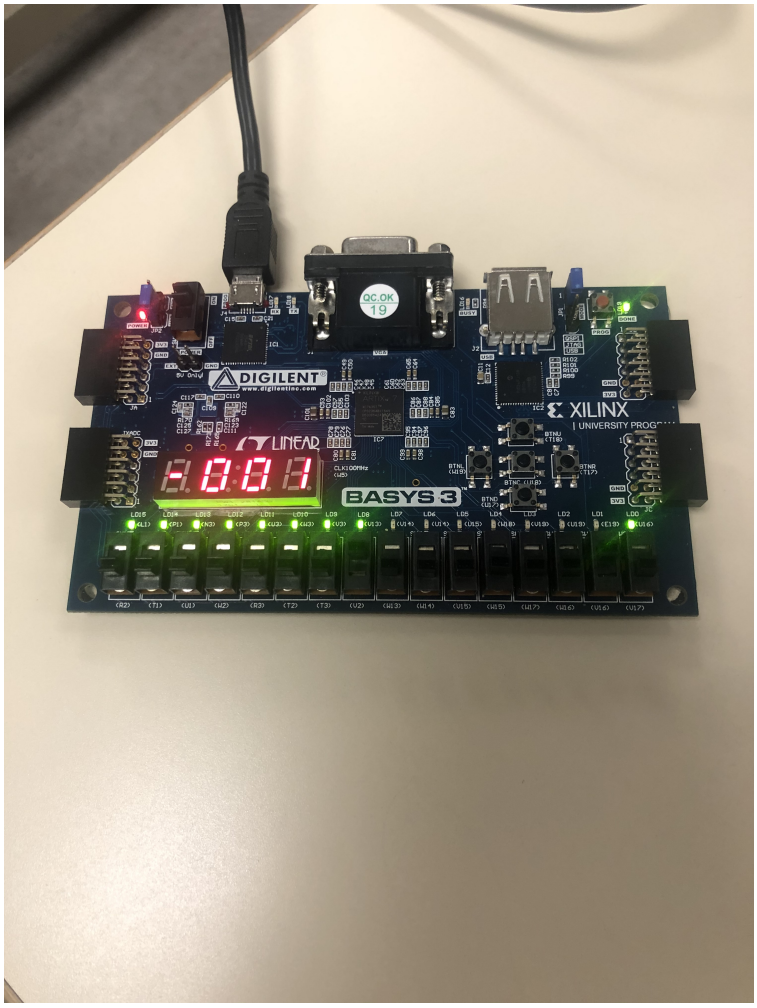
2. Show 2's Complement Test Simulation



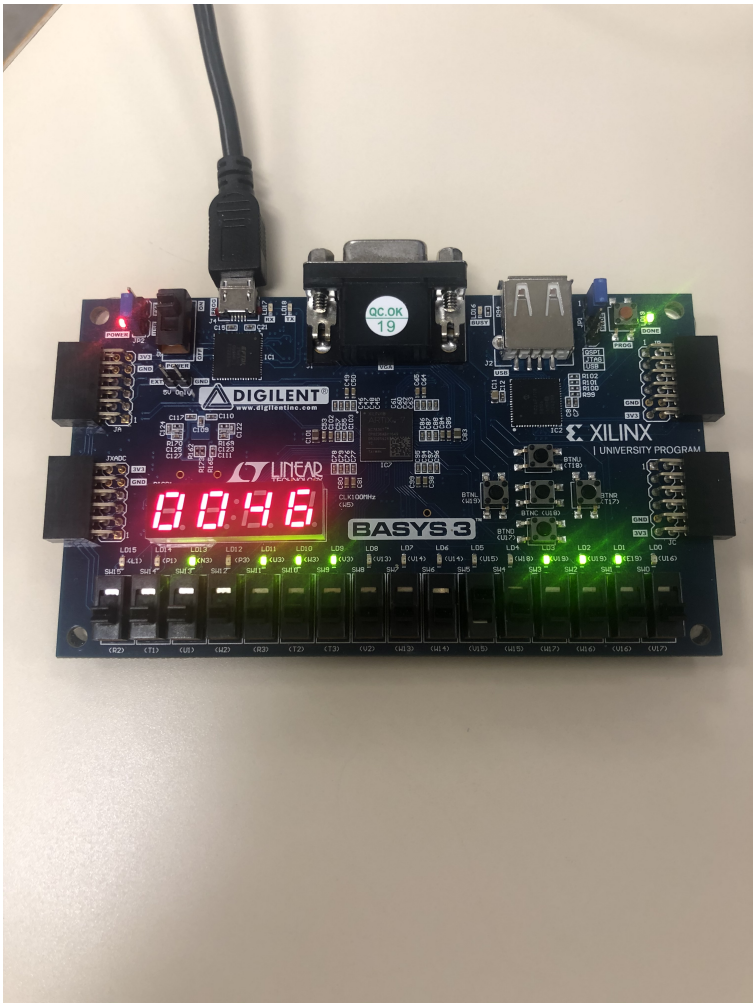
3. Full Strength Four Digits



4. Negative Number Display



5. Positive Number Display



Code

Listing 1: Counter

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

   wire [N-1:0] Qnext;
   wire [N-1:0] Qreg;
   assign Qnext = Qreg +1;

   register #(.N(N)) my_reg_in_count1(
      .D(Qnext),
      .clk(clk),
      .en(1),
      .rst(rst),
      .Q(Qreg)
   );
```

```
   assign MSBs = Qreg[N-1:N-2];
endmodule
```

Listing 2: Counter Test

```
module counter_test();
   reg clk;
   reg rst_t;
   wire [1:0] MSBs_t;

   counter #(.N(4)) dut(
      .clk(clk),
      .rst(rst_t),
      .MSBs(MSBs_t)
   );

   always begin
      clk = ~clk; #5;
   end

   initial begin
      clk = 0;
      rst_t = 0; #10;
      rst_t = 1; #3;
      rst_t = 0; #10;
      $finish;
   end

endmodule
```

Listing 3: Wrapper

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

   wire [1:0] MSBs;

   counter #(.N(20)) my_counter(
      .clk(clk),
      .rst(btnC),
      .MSBs(MSBs)
   );


   sseg4 my_sseg4_lab10(
      .data(16'b0),
      .hex_dec(0),
      .sign(0),
      .digit_sel(MSBs),
```

```
      .an(an),
      .dp(dp),
      .seg(seg)
   );
endmodule
```

Listing 4: Show 2's Complement

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

   wire [7:0] mux_out;
   wire [7:0] converted_input;
   assign converted_input = ~Din[7:0] + 1;

   mux2 #(.BITS(8)) my_mux2_lab10(
      .in0(Din),
      .in1(converted_input),
      .sel(Din[7]),
      .out(mux_out)
   );


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

endmodule
```

Listing 5: Show 2's Complement Test

```
module show_2c_test();

   reg [7:0] Din_t;
   wire [15:0] Dout_t;
   reg 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 = 20; #10;
      Din_t = -20; #10;
      Din_t = 5; #10;
      Din_t = -5; #10;
      $finish;
```

```
   end
endmodule
```

Listing 6: Top Lab Module

```
module top_lab_10(
   input btnU,
   input btnD,
   input [10:0] sw,
   input clk,
   input btnC,
   output [6:0] seg,
   output dp,
   output [3:0] an,
   output [15:0] led
   );

   wire [15:0] lab9Out;
   wire TwocSignOut;
   wire [15:0] TwocMagOut;
   wire [1:0] MSBs;

   top_lab9 my_top_lab9(
      .btnU(btnU),
      .btnD(btnD),
      .btnC(btnC),
      .sw(sw),
      .clk(clk),
      .led(lab9Out)
   );

   show_2c my_show_2c(
      .Din(lab9Out[15:8]),
      .Dout(TwocMagOut),
      .sign(TwocSignOut)
   );

   counter #(.N(20)) my_counter(
      .clk(clk),
      .rst(btnC),
      .MSBs(MSBs)
   );


   sseg4 my_sseg4_lab10(
      .data(TwocMagOut),
      .hex_dec(sw[15]),
      .sign(TwocSignOut),
      .digit_sel(MSBs),
      .an(an),
      .dp(dp),
      .seg(seg)
   );
```

```
   assign led = lab9Out;

endmodule
```