ECE3300L Lab 6

Dual BCD Up/Down Counters, ALU, and Control Display on 7-Segment

Group X
Czyrone Agbayani (BID: 014766336)
Caleb Jala-Guinto (BID: 015446587)

Objective:

The goal of this lab is to build a two-digit BCD up/down counter. It uses a 32-bit clock divider and a 32-to-1 multiplexer, driven by a 5-bit switch, to set how fast it counts. A top push-button selects the count direction, and a middle push-button resets the counter. LEDs above switches 5–12 light up to match the numbers shown on the 7-segment display.

Top Display Module:

```
22  module top_lab6(
23      input wire CLK,
24      input wire [4:0] SW, //Select
25      input wire [1:0] ALU,
26      input wire rst, // reset
27      input wire dir0, // dir0
28      input wire dir1, // dir1
29      input wire DP,
30      output wire [6:0] seg,
31      output wire [7:0] an,
32      output wire [7:0] LED
33  );
34
35      wire rst_n = ~rst;
36
37      wire [31:0] cnt;
38      wire clk_out;
39      wire [3:0] value0, value1;
40      wire [7:0] result;
41      wire [3:0] result_units, result_tens;
42
43      wire [3:0] ControlNibble;
44
45      assign DP = 0;
46      assign LED[3:0] = value0;
47      assign LED[7:4] = value1;
48
49     control_decoder DC(
50          .Sw4(ALU[0]),
51          .Sw3(ALU[1]),
52          .Sw2(dir0),
53          .Sw1(dir1),
54          .ctrl_nibble(ControlNibble)
55      );
56
57      clock_divider u_clkdiv (
58          .clk(CLK),
59          .rst_n(rst_n),
60          .cnt(cnt)
61      );
```

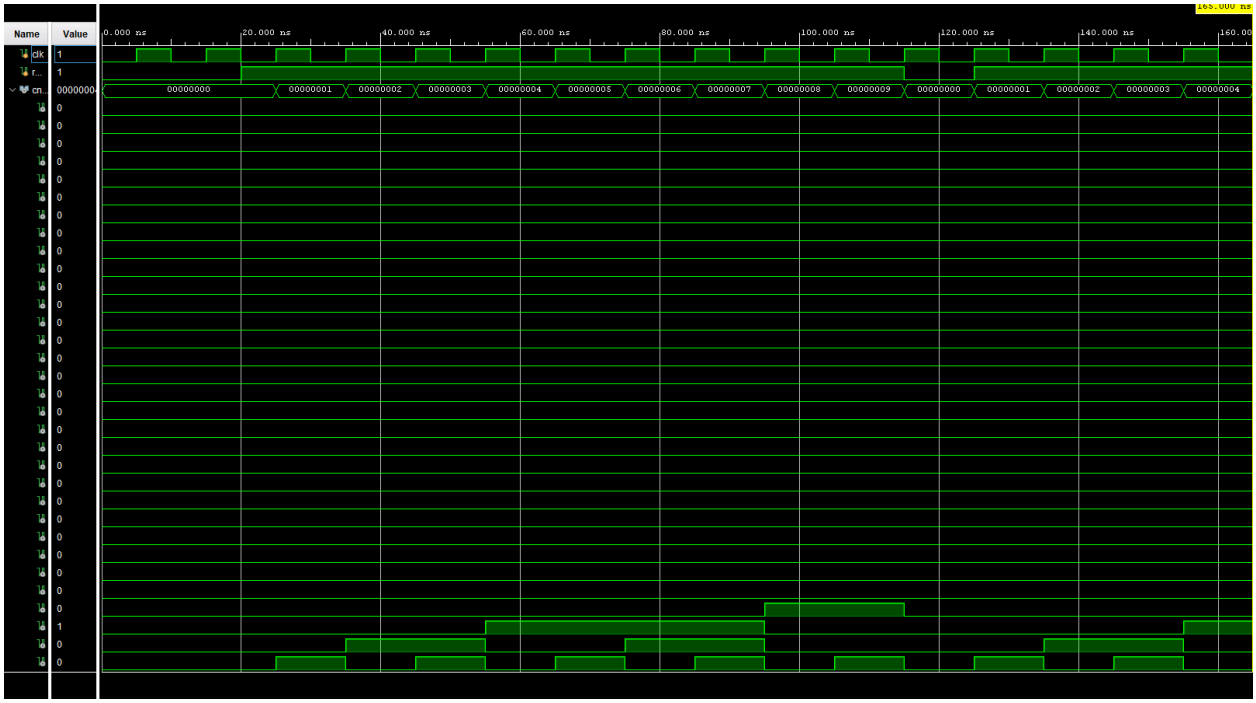
```
63      mux32x1 u_mux (
64          .cnt(cnt),
65          .sel(SW),
66          .clk_out(clk_out)
67      );
68
69      bcd_up_down_counter BCD0_COUNTER (
70          .clk_processed(clk_out),
71          .rst_n(rst_n),
72          .dir(dir0),
73          .digit(value0)
74      );
75
76      bcd_up_down_counter BCD1_COUNTER (
77          .clk_processed(clk_out),
78          .rst_n(rst_n),
79          .dir(dir1),
80          .digit(value1)
81      );
82
83      binary_to_bcd u_bcd (
84          .binary(result),
85          .units(result_units),
86          .tens(result_tens)
87      );
88
89      seg7_scan u_seg (
90          .clk(CLK),
91          .rst_n(rst_n),
92          .onesPlace(result_units),
93          .tensPlace(result_tens),
94          .digitCtrl(ControlNibble),
95          .SEG(seg),
96          .AN(an)
97      );
98
99      alu ADDSUB(
100         .A(value0),
101         .B(value1),
102         .ctrl(ALU),
103         .out(result)
104     );
105
106 endmodule
```

XDC File:

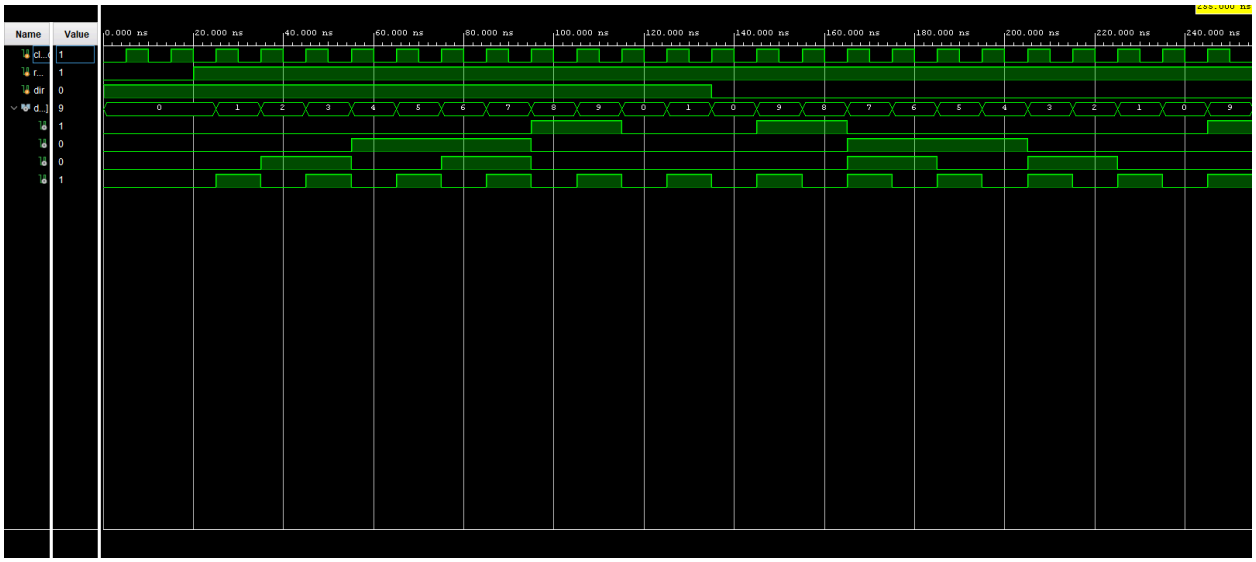
```
# Clock signal
set_property -dict { PACKAGE_PIN E3    IOSTANDARD LVCMOS33 } [get_ports { CLK }]; #IO_L12P_T1_MRCC_35 Sch=clk100mhz
create_clock -add -name sys_clk_pin -period 10.00 -waveform {0 5} [get_ports {CLK}];


##Switches

set_property -dict { PACKAGE_PIN J15   IOSTANDARD LVCMOS33 } [get_ports { SW[0] }]; #IO_L24N_T3_RS0_15 Sch=sw[0]
set_property -dict { PACKAGE_PIN L16   IOSTANDARD LVCMOS33 } [get_ports { SW[1] }]; #IO_L3N_T0_DQS_EMCCLK_14 Sch=sw[1]
set_property -dict { PACKAGE_PIN M13   IOSTANDARD LVCMOS33 } [get_ports { SW[2] }]; #IO_L6N_T0_D08_VREF_14 Sch=sw[2]
set_property -dict { PACKAGE_PIN R15   IOSTANDARD LVCMOS33 } [get_ports { SW[3] }]; #IO_L13N_T2_MRCC_14 Sch=sw[3]
set_property -dict { PACKAGE_PIN R17   IOSTANDARD LVCMOS33 } [get_ports { SW[4] }]; #IO_L12N_T1_MRCC_14 Sch=sw[4]
set_property -dict { PACKAGE_PIN T18   IOSTANDARD LVCMOS33 } [get_ports { ALU[0] }]; #IO_L7N_T1_D10_14 Sch=sw[5]
set_property -dict { PACKAGE_PIN U18   IOSTANDARD LVCMOS33 } [get_ports { ALU[1] }]; #IO_L17N_T2_A13_D29_14 Sch=sw[6]
set_property -dict { PACKAGE_PIN R13   IOSTANDARD LVCMOS33 } [get_ports { dir0 }]; #IO_L5N_T0_D07_14 Sch=sw[7]
set_property -dict { PACKAGE_PIN T8    IOSTANDARD LVCMOS18 } [get_ports { dir1 }]; #IO_L24N_T3_34 Sch=sw[8]

## LEDs

set_property -dict { PACKAGE_PIN H17   IOSTANDARD LVCMOS33 } [get_ports { LED[0] }]; #IO_L18P_T2_A24_15 Sch=led[0]
set_property -dict { PACKAGE_PIN K15   IOSTANDARD LVCMOS33 } [get_ports { LED[1] }]; #IO_L24P_T3_RS1_15 Sch=led[1]
set_property -dict { PACKAGE_PIN J13   IOSTANDARD LVCMOS33 } [get_ports { LED[2] }]; #IO_L17N_T2_A25_15 Sch=led[2]
set_property -dict { PACKAGE_PIN N14   IOSTANDARD LVCMOS33 } [get_ports { LED[3] }]; #IO_L8P_T1_D11_14 Sch=led[3]
set_property -dict { PACKAGE_PIN R18   IOSTANDARD LVCMOS33 } [get_ports { LED[4] }]; #IO_L7P_T1_D09_14 Sch=led[4]
set_property -dict { PACKAGE_PIN V17   IOSTANDARD LVCMOS33 } [get_ports { LED[5] }]; #IO_L18N_T2_A11_D27_14 Sch=led[5]
set_property -dict { PACKAGE_PIN U17   IOSTANDARD LVCMOS33 } [get_ports { LED[6] }]; #IO_L17P_T2_A14_D30_14 Sch=led[6]
set_property -dict { PACKAGE_PIN U16   IOSTANDARD LVCMOS33 } [get_ports { LED[7] }]; #IO_L18P_T2_A12_D28_14 Sch=led[7]
set property -dict { PACKAGE_PIN V16   IOSTANDARD LVCMOS33 } [get ports { LED[8] }]; #IO L16N T2 A15 D31 14 Sch=led[8]

#7 segment display

set_property -dict { PACKAGE_PIN T10   IOSTANDARD LVCMOS33 } [get_ports {  seg[0] }]; #IO_L24N_T3_A00_D16_14 Sch=ca
set_property -dict { PACKAGE_PIN R10   IOSTANDARD LVCMOS33 } [get_ports {  seg[1] }]; #IO_25_14 Sch=cb
set_property -dict { PACKAGE_PIN K16   IOSTANDARD LVCMOS33 } [get_ports {  seg[2] }]; #IO_25_15 Sch=cc
set_property -dict { PACKAGE_PIN K13   IOSTANDARD LVCMOS33 } [get_ports {  seg[3] }]; #IO_L17P_T2_A26_15 Sch=cd
set_property -dict { PACKAGE_PIN P15   IOSTANDARD LVCMOS33 } [get_ports {  seg[4] }]; #IO_L13P_T2_MRCC_14 Sch=ce
set_property -dict { PACKAGE_PIN T11   IOSTANDARD LVCMOS33 } [get_ports {  seg[5] }]; #IO_L19P_T3_A10_D26_14 Sch=cf
set_property -dict { PACKAGE_PIN L18   IOSTANDARD LVCMOS33 } [get_ports {  seg[6] }]; #IO_L4P_T0_D04_14 Sch=cg

#set_property -dict { PACKAGE_PIN H15   IOSTANDARD LVCMOS33 } [get_ports { dp }]; #IO_L19N_T3_A21_VREF_15 Sch=dp

set_property -dict { PACKAGE_PIN J17   IOSTANDARD LVCMOS33 } [get_ports { an[0] }]; #IO_L23P_T3_FOE_B_15 Sch=an[0]
set_property -dict { PACKAGE_PIN J18   IOSTANDARD LVCMOS33 } [get_ports { an[1] }]; #IO_L23N_T3_FWE_B_15 Sch=an[1]
set_property -dict { PACKAGE_PIN T9    IOSTANDARD LVCMOS33 } [get_ports { an[2] }]; #IO_L24P_T3_A01_D17_14 Sch=an[2]
set_property -dict { PACKAGE_PIN J14   IOSTANDARD LVCMOS33 } [get_ports { an[3] }]; #IO_L19P_T3_A22_15 Sch=an[3]
set_property -dict { PACKAGE_PIN P14   IOSTANDARD LVCMOS33 } [get_ports { an[4] }]; #IO_L8N_T1_D12_14 Sch=an[4]
set_property -dict { PACKAGE_PIN T14   IOSTANDARD LVCMOS33 } [get_ports { an[5] }]; #IO_L14P_T2_SRCC_14 Sch=an[5]
set_property -dict { PACKAGE_PIN K2    IOSTANDARD LVCMOS33 } [get_ports { an[6] }]; #IO_L23P_T3_35 Sch=an[6]
set_property -dict { PACKAGE_PIN U13   IOSTANDARD LVCMOS33 } [get_ports { an[7] }]; #IO_L23N_T3_A02_D18_14 Sch=an[7]

##Buttons

set_property -dict { PACKAGE_PIN C12   IOSTANDARD LVCMOS33 } [get_ports { CPU_RESETN }]; #IO_L3P_T0_DQS_AD1P_15 Sch=cpu_resetn

set property -dict { PACKAGE PIN N17   IOSTANDARD LVCMOS33 } [get ports { rst }]; #IO L9P T1 DQS 14 Sch=btnc
```

Test Benches:

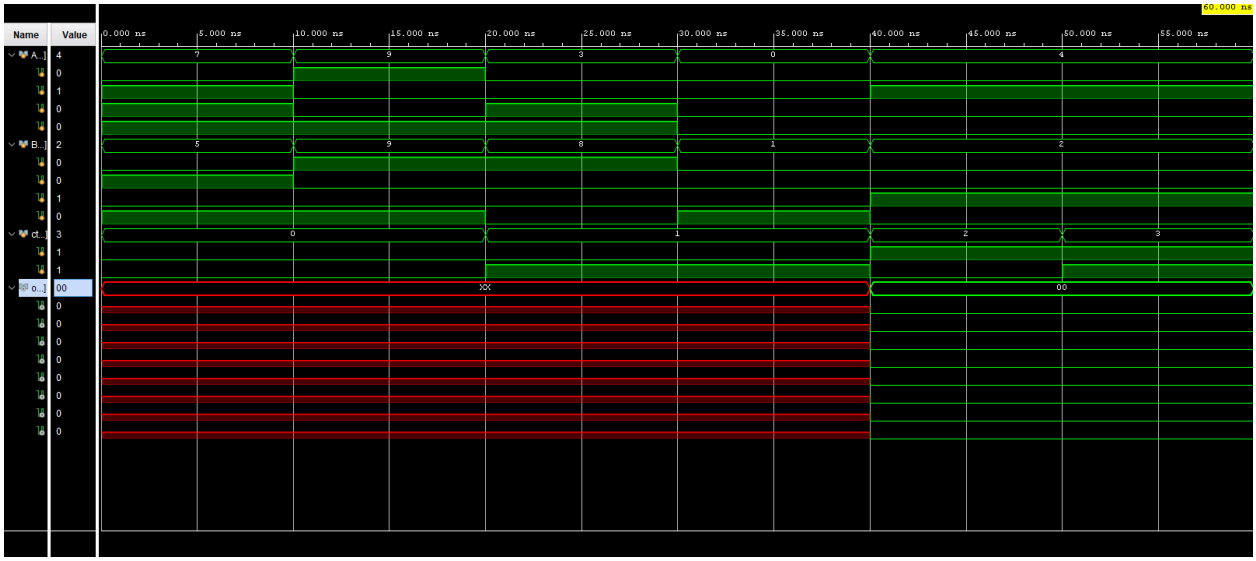
clock_divider _tb



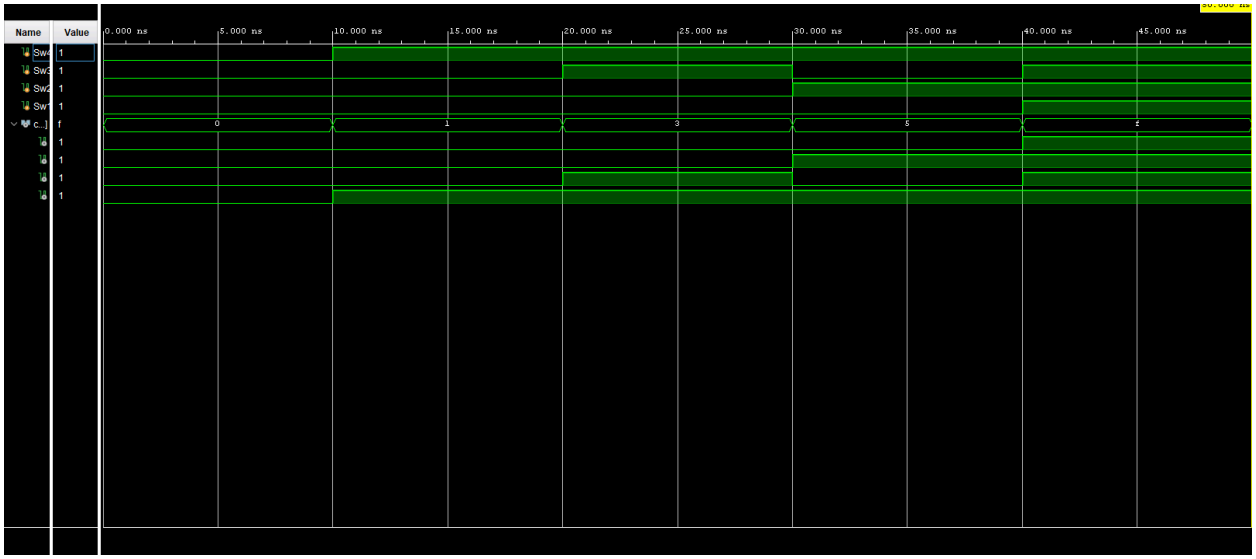
bcd_counter_tb



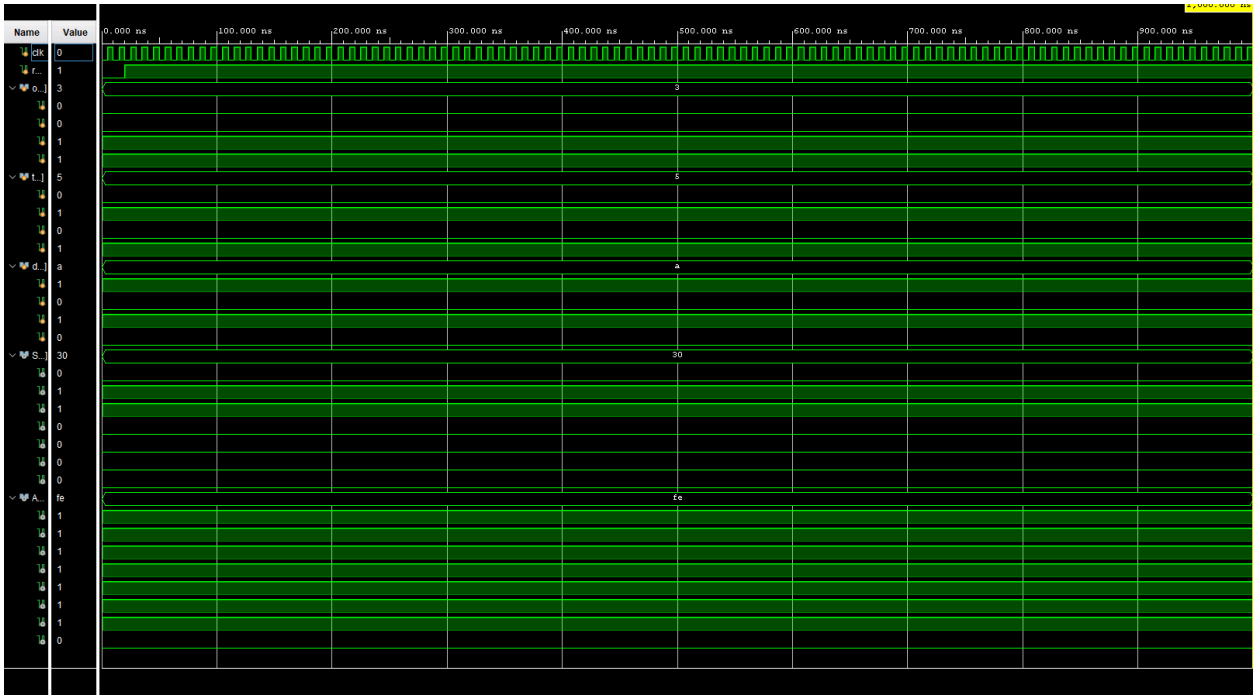
alu_tb



control_decoder_tb



seg7_scan_tb



Video Link:
https://youtu.be/l0HdzIq_M1M

Contributions:
Czyrone (50%) - Verilog code, Physical demo
Caleb (50%) - Verilog code, testbenches
Both worked on the lab report together and troubleshooted code

Reflections:
This lab required more troubleshooting than I expected. The reverse-count only worked while I held down the direction button, so I had to tweak how the input was sampled. The 7-segment display was also ghosting, showing faint traces of other digits, until I fixed the multiplexing logic and corrected some pin assignments in the constraint file. Overall, these fixes highlighted how important proper pin mapping and timing control are in FPGA projects.