Julio Flores (ID# : 016326856)
Victor Perez (ID# : 016196050)

Group I

Session E02

Lab 5

BCD up/down counter on 7-segment display

Monday

JULY 21, 2025

ECE 3300L

Summer 2025

Objective:

The objective of this lab is to design and implement a two-digit BCD up/down counter using Verilog-HDL. The design incorporates a 32-bit clock divider and a 32×1 multiplexer to enable five selectable counting speeds via slide switches. The counter's direction is controlled by BTN1, while BTN0 is used for asynchronous reset. The output is displayed on a dual-digit 7-segment display using multiplexing techniques. The final design is simulated, synthesized, programmed, and tested on the Nexys A7 FPGA board.

Design(Code):

- This block of code assigns the counter and selectors as inputs and then the clk as the output

```
23  module mux32x1(
24      input [31:0] cnt,
25      input [4:0] sel,
26      output clk_out
27  );
28      assign clk_out = cnt[sel];
29  endmodule
30
```

- The seg7_scan module initialized the inputs and outputs.
- The case statement activates the first and second digit on the seven segment display
- The second case statement choose what number to display based on the input

```
module seg7_scan(
    input clk,
    input [3:0] digit1,
    input [3:0] digit0,
    output reg [7:0] AN,
    output reg [6:0] SEG
);

    reg [15:0] refresh_counter = 0;
    wire sel;
    reg [3:0] current_digit;

    always @(posedge clk) begin
        refresh_counter <= refresh_counter + 1;
    end

    assign sel = refresh_counter[15];



    always @(*) begin
        case (sel)
            1'b0: begin
                AN <= 8'b11111110; // Activate digit0
                SEG <= seg_decoder(digit0);
            end
            1'b1: begin
                AN <= 8'b11111101; // Activate digit1
                SEG <= seg_decoder(digit1);
            end
        endcase
    end

    function [6:0] seg_decoder;
        input [3:0] num;
        case (num)
            4'd0: seg_decoder = 7'b1000000;
            4'd1: seg_decoder = 7'b1111001;
            4'd2: seg_decoder = 7'b0100100;
            4'd3: seg_decoder = 7'b0110000;
            4'd4: seg_decoder = 7'b0011001;
            4'd5: seg_decoder = 7'b0010010;
            4'd6: seg_decoder = 7'b0000010;
            4'd7: seg_decoder = 7'b1111000;
            4'd8: seg_decoder = 7'b0000000;
            4'd9: seg_decoder = 7'b0010000;
            default: seg_decoder = 7'b1111111; // off
        endcase
    endfunction

endmodule

```

- initializes the clock as a 32 bit counter .

```
module clock_divider(
    input clk,
    output reg [31:0] cnt
);
    always @(posedge clk)
        cnt <= cnt + 1;
endmodule
```

- This block of code initializes the up and down counter for the display.
- Reset button is active low to count down and if pressed, digits count up.
- If-else loop utilized to count from 0-9(up) or 9-0(down)

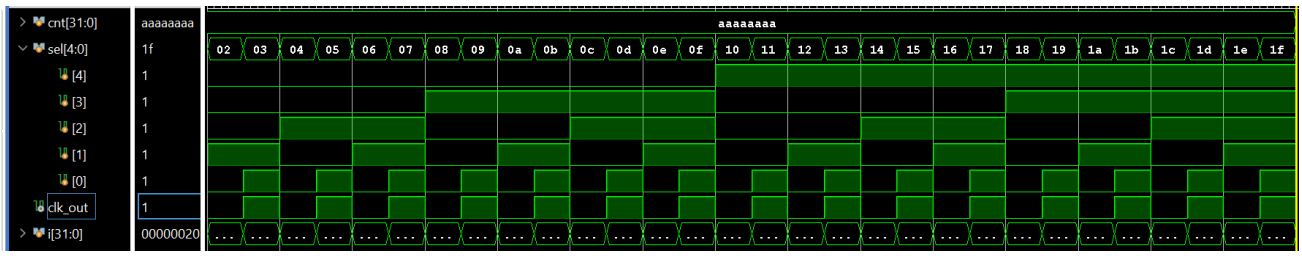
```
module bcd_up_down_counter(
    input clk_out,
    input rst_n,
    input dir, // 1 for up, 0 for down
    output reg [3:0] digit1, // tens
    output reg [3:0] digit0  // units
);
    always @(posedge clk_out or negedge rst_n) begin
        if (!rst_n) begin
            digit0 <= 0;
            digit1 <= 0;
        end else if (dir) begin
            // Count Up
            if (digit0 == 9) begin
                digit0 <= 0;
                if (digit1 == 9)
                    digit1 <= 0;
                else
                    digit1 <= digit1 + 1;
            end else begin
                digit0 <= digit0 + 1;
            end
        end else begin
            // Count Down
            if (digit0 == 0) begin
                digit0 <= 9;
                if (digit1 == 0)
                    digit1 <= 9;
                else
                    digit1 <= digit1 - 1;
            end else begin
                digit0 <= digit0 - 1;
            end
        end
    end
endmodule
```

- Assigns every input and output needed for entire project
- Main module to call all the functions.

```
module top_lab5(
    input CLK,
    input [4:0] SW,
    input [1:0] BTN,
    output [7:0] AN,
    output [6:0] SEG,
    output [12:0] LED
);

    wire [31:0] cnt;
    wire clk_out;
    wire [3:0] digit0, digit1;

    assign rst_n = ~BTN[0];

    // Instantiate clock divider
    clock_divider cd (
        .clk(CLK),
        .cnt(cnt)
    );

    // Instantiate Mux
    mux32x1 mux (
        .cnt(cnt),
        .sel(SW),
        .clk_out(clk_out)
    );

    // Instantiate BCD counter
    bcd_up_down_counter bcd (
        .clk_out(clk_out),
        .rst_n(rst_n),
        .dir(BTN[1]),
        .digit0(digit0),
        .digit1(digit1)
    );

    // Instantiate 7-segment scan
    seg7_scan scan (
        .clk(CLK),
        .digit0(digit0),
        .digit1(digit1),
        .AN(AN),
        .SEG(SEG)
    );

    assign LED[8:5] = digit0;
    assign LED[12:9] = digit1;
    assign LED[4:0] = SW;

endmodule
```

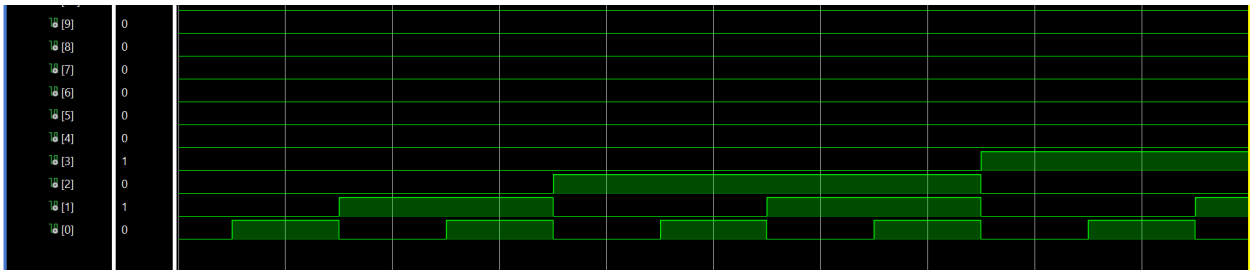
Testbench and Waveform:

Mux32x1_tb



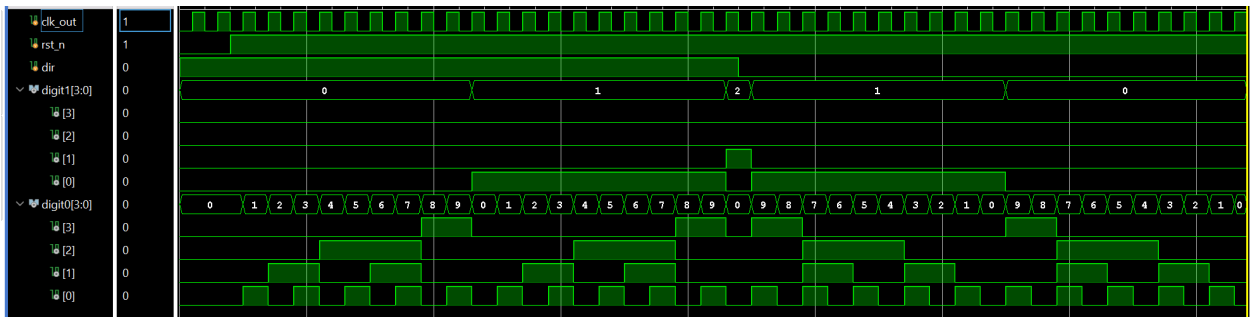
- Shows the clk_out equivalent to the counter

Clock_divider_tb



- Shows an increment in binary form up to 32 bits.

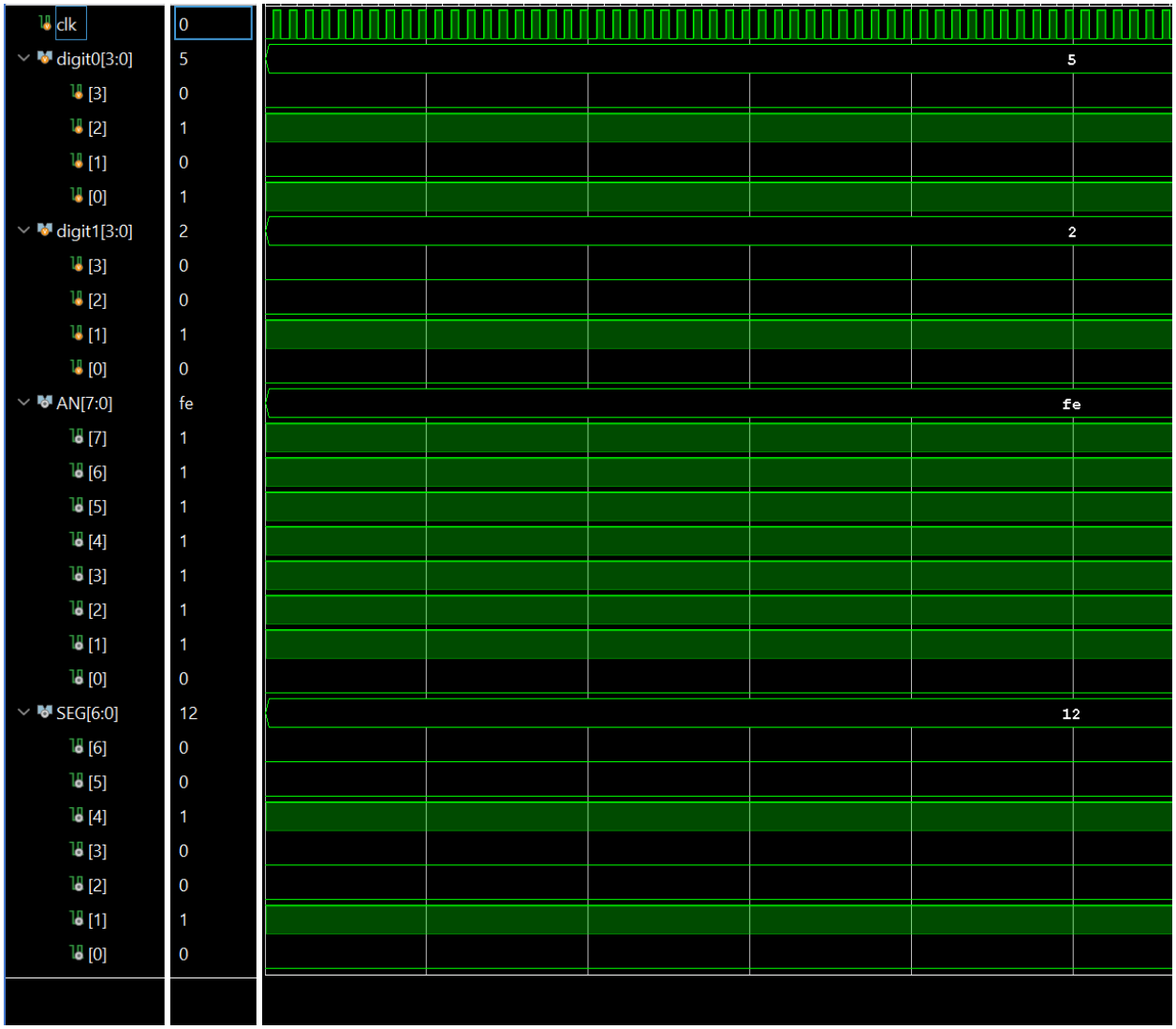
BCD_up_down_counter_tb



- Digit 0 counts up to 9, once at 9, digit 1 is active and digit 0 resets to 0 to create the number 10.

- At 20, digit 0 counts down from 9-0, at 0 digit 1 is decreased to 0 again to create single value numbers.

Seg7_scan_tb



- Digit 0 and 1 are set as 2 and 5
- AN connects the 7 seg display to either digit 0 or 1, in this case it is connected to digit 0
- To connect to digit 1, AN must be 0b10111111
- The 7 seg display is showing the number 5 from digit 0.

Implementation:

1. Slice Logic

Site Type	Used	Fixed	Prohibited	Available	Util%
Slice LUTs*	28	0	0	63400	0.04
LUT as Logic	28	0	0	63400	0.04
LUT as Memory	0	0	0	19000	0.00
Slice Registers	56	0	0	126800	0.04
Register as Flip Flop	56	0	0	126800	0.04
Register as Latch	0	0	0	126800	0.00
F7 Muxes	4	0	0	31700	0.01
F8 Muxes	0	0	0	15850	0.00

Contributions:

Julio Flores: 50% - Verilog Code (seg7_scan, top_lab5, testbench for these modules) and report

Victor Perez : 50% - Verilog Code (mux, clk divider, and bcd counter modules, testbench for these modules) and report

Reflection:

This lab provided valuable hands-on experience in digital design using Verilog-HDL, reinforcing concepts such as clock division, control logic, and display multiplexing. Implementing the up/down counter with selectable speed and directional control shows the fundamentals of synchronous circuit design and hardware implementation. Finally implementing the project on the physical board and simulating it in the testbench shows the functionality of this design.