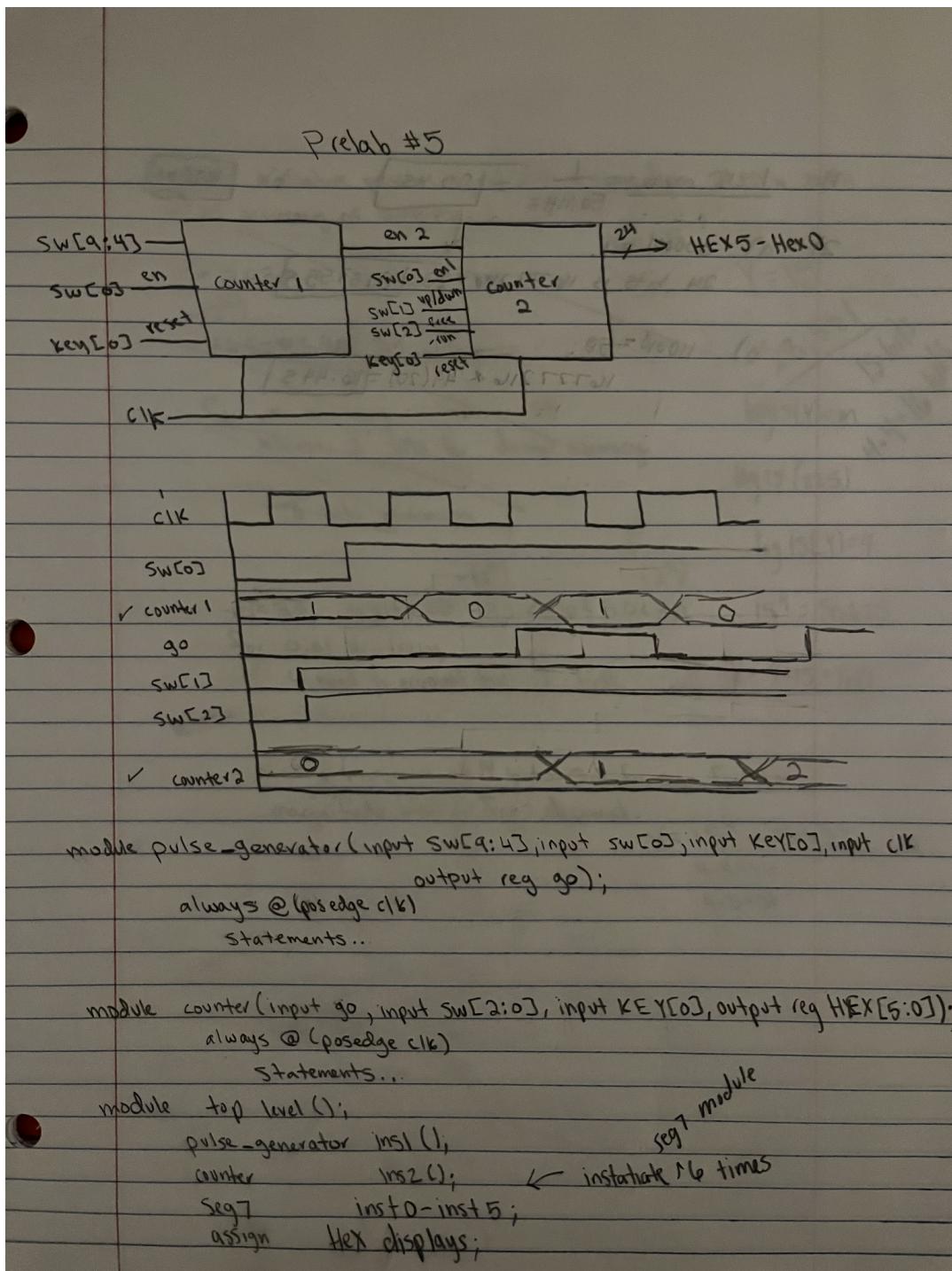


Alejandro Torres  
915207383

### Lab5: Counter Design

Pre-lab:



$$1 \text{ clk cycle} = \frac{1}{50 \text{ MHz}} = 20 \text{ ns}$$

b)  $1000 \text{ ns}$

$$2) \text{ a) } 000001 = 1$$

$$24 \text{ bits is } 10777216 * 20 = 0.3355 \text{ s}$$

$$\text{b) } 110010 = 50$$

$$10777216 * 49(20) = 10.44 \text{ s}$$

Top.v:

```
1  module top(
2    input      [7:0]  MAX10_CLK1_50,
3    output     [7:0]  HEX0,
4    output     [7:0]  HEX1,
5    output     [7:0]  HEX2,
6    output     [7:0]  HEX3,
7    output     [7:0]  HEX4,
8    output     [7:0]  HEX5,
9    input      [1:0]  KEY,
10   output     [9:0]  LEDR,
11   input      [9:0]  SW
12 );
13
14  wire pulse_out;
15  wire [23:0] counter_out;
16
17 // Instantiate counters
18 pulse_gen count1(
19   .divideby(SW[9:4]),
20   .en(SW[0]),
21   .reset(~KEY[0]),
22   .clk(MAX10_CLK1_50),
23   .go(pulse_out),
24   .led(LEDR[9:0]));
25
26 main_counter count2(
27   .en1(SW[0]),
28   .en2(pulse_out),
29   .up_down(SW[1]),
30   .free_run(SW[2]),
31   .reset(~KEY[0]),
32   .clk(MAX10_CLK1_50),
33   .count(counter_out));
34
35 // Instantiate 6 7-seg displays
36 seg7 hex0(
37   .in(counter_out[3:0]),
38   .out(HEX0));
39
40 seg7 hex1(
41   .in(counter_out[7:4]),
42   .out(HEX1));
43
44 seg7 hex2(
45   .in(counter_out[11:8]),
46   .out(HEX2));
47
48 seg7 hex3(
49   .in(counter_out[15:12]),
50   .out(HEX3));
51
52 seg7 hex4(
53   .in(counter_out[19:16]),
54   .out(HEX4));
55
56 seg7 hex5(
57   .in(counter_out[23:20]),
58   .out(HEX5));
59
60 endmodule
61
```

Main\_counter.v:

```
1  module main_counter(
2    input en1,
3    input en2,
4    input up_down,
5    input free_run,
6    input reset,
7    input clk,
8    output reg [23:0] count = 0);
9
10   always @(posedge clk) begin
11     if (reset == 1'b0) begin
12       if (en1 == 1'b1 && en2 == 1'b1) begin
13
14         // Increment indefinitely
15         if (free_run == 1'b1 && up_down == 1'b1) begin
16           count <= (count == 24'hFFFFF) ? 24'd0: count + 1;
17         end
18
19         // decrement indefinitely
20         else if (free_run == 1'b1 && up_down == 1'b0) begin
21           count <= (count == 24'd0) ? 24'hFFFFF: count - 1;
22         end
23
24         else if (free_run == 1'b0) begin
25           // Halt at half of max value
26           if (count == 24'd8388607) begin
27             count <= count;
28           end
29           else if (up_down == 1'b1) begin
30             count <= (count == 24'hFFFFF) ? 24'd0: count + 1;
31           end
32           else begin
33             count <= (count == 24'd0) ? 24'hFFFFF: count - 1;
34           end
35         end
36       end
37     end
38     // Reset has priority
39     if (reset == 1'b1) begin
40       count <= 24'd0;
41     end
42   end
43
44 endmodule
45
```

Pulse\_gen.v:

```
1  module pulse_gen(
2      input [5:0] divideby,
3      input en,
4      input reset,
5      input clk,
6      output reg go = 1'b0,
7      output [9:0] led,
8      output reg [5:0] count = 6'd0);
9
10     assign led = (divideby == 6'd0) ? 10'h3ff : 10'h000;
11
12     always @(posedge clk) begin
13         // Reset holds priority
14         if (reset == 1'b0) begin
15             if (en == 1'b1) begin
16                 if (divideby == 6'd0) begin
17                     count <= 6'd0;
18                     go <= 1'b0;
19                 end
20             else if (count == (divideby - 1)) begin
21                 count <= 6'd0;
22                 go <= 1'b1;
23             end
24             else begin
25                 count <= count + 6'd1;
26                 go <= 1'b0;
27             end
28         end
29     end
30     else begin
31         count <= 6'd0;
32         go <= 1'b0;
33     end
34 endmodule
35
36
```

## Pulse\_gen\_tb.v:

```
1 `timescale 1ps/1ps
2 module pulse_gen_tb;
3
4 reg [5:0] divideby;
5 reg en, reset, clk;
6 wire out;
7 wire [5:0] count;
8 wire [9:0] ledr;
9
10 //pulse_gen count1(
11 //    .divideby(divideby[5:0]),
12 //    .en(en),
13 //    .reset(reset),
14 //    .clk(clk),
15 //    .go(out),
16 //    .led(ledr[9:0]),
17 //    .count(count[5:0]));
18
19 initial begin
20     clk = 0;
21     en = 1;
22     reset = 0;
23     divideby = 6'd0;
24     $display("CASE 1: divideby = %b", divideby);
25     repeat(4) begin
26         #100
27         clk = 1;
28         $display("divideby = %b, go = %b, ledr = %b", divideby, clk, ledr);
29         #100
30         clk = 0;
31     end
32
33     #100
34     clk = 1;
35     reset = 1;
36     #100
37     clk = 0;
38     reset = 0;
39     divideby = 6'd2;
40     $display("CASE 2: divideby = %b", divideby);
41     repeat(8) begin
42         #100
43         clk = 1;
44         $display("divideby = %b, go = %b, count = %b, reset = %b, en = %b", divideby, out, count, reset, en);
45         #100
46         clk = 0;
47     end
48
49     #100
50     clk = 1;
51     reset = 1;
52     #100
53     clk = 0;
54     reset = 0;
55     divideby = 6'd63;
56     $display("CASE 3: divideby = %b", divideby);
57     repeat(65) begin
58         #100
59         clk = 1;
60         $display("divideby = %b, go = %b, count = %b, reset = %b, en = %b", divideby, out, count, reset, en);
61         #100
62         clk = 0;
63     end
64
65     #100
66     clk = 1;
67     reset = 1;
68     #100
69     reset = 0;
70     en = 0;
71     divideby = 6'd3;
72     $display("CASE 3: ENABLE OFF: en = %b", en);
73     repeat(5) begin
74         #100
75         clk = 1;
76         $display("divideby = %b, go = %b, count = %b, reset = %b, en = %b", divideby, out, count, reset, en);
77         #100
78         clk = 0;
79     end
80 end
81
82 endmodule
```

## Printout:

```

# CASE 3: ENABLE OFF: en = 0
# divideby = 000011, go = 0, count = 000000, reset = 0, en = 0
# divideby = 000011, go = 0, count = 000000, reset = 0, en = 0
# divideby = 000011, go = 0, count = 000000, reset = 0, en = 0
# divideby = 000011, go = 0, count = 000000, reset = 0, en = 0
# divideby = 000011, go = 0, count = 000000, reset = 0, en = 0

```

Top\_tb.v:

```

1  `timescale 1ps/1ps
2  module top_tb;
3
4  wire [7:0] HEX0, HEX1, HEX2, HEX3, HEX4, HEX5;
5  wire [9:0] LEDR;
6
7  reg clk;
8  reg [9:0] SW;
9  reg [1:0] KEY;
10
11  top test(
12    .MAX10_CLK1_50(clk),
13    .HEX0(HEX0),
14    .HEX1(HEX1),
15    .HEX2(HEX2),
16    .HEX3(HEX3),
17    .HEX4(HEX4),
18    .HEX5(HEX5),
19    .KEY(KEY),
20    .LEDR(LEDR),
21    .SW(SW));
22
23  initial begin
24    clk = 0;
25    SW[0] = 1; // Enable on
26    SW[1] = 0; // initialize go
27    SW[2:1] = 2'b00; // FREE_RUN && UP_DOWN
28    SW[9:4] = 6'b00_0000; // Divideby
29    KEY[0] = 1'b1; // reset is off
30
31    #100
32    clk = 1;
33    $display("Initially LEDs are on and displays are 0");
34    $display("LEDR = %b, HEX0=%b, HEX1=%b, HEX2=%b, HEX3=%b, HEX4=%b, HEX5=%b", LEDR, HEX0, HEX1, HEX2, HEX3, HEX4, I
35
36    #100
37    clk = 0;
38    SW[2:1] = 2'b11; // FREE_RUN && UP_DOWN
39    SW[9:4] = 6'b00_0010; // Divideby
40    $display("Free-run is high and up_down is high");
41  end
42  repeat(7) begin
43    #100
44    clk = 1;
45    $display("LEDR = %b, HEX0=%b, HEX1=%b, HEX2=%b, HEX3=%b, HEX4=%b, HEX5=%b", LEDR, HEX0, HEX1, HEX2, HEX3, HEX
46    #100
47    clk = 0;
48  end

```

```

48
49     SW[2:1] = 2'b10; // FREE_RUN && UP_DOWN
50     SW[9:4] = 6'b00_0010; // Divideby
51     $display("Free-run is high and up_down is low");
52     repeat(6) begin
53         #100
54         clk = 1;
55         $display("LEDR = %b, HEX0=%b, HEX1=%b, HEX2=%b, HEX3=%b, HEX4=%b, HEX5=%b", LEDR, HEX0, HEX1, HEX2, HEX3, HEX4,
56         #100
57         clk = 0;
58     end
59
60     SW[2:1] = 2'b01; // FREE_RUN && UP_DOWN
61     SW[9:4] = 6'b00_0010; // Divideby
62     $display("Free-run is low and up_down is high");
63     repeat(6) begin
64         #100
65         clk = 1;
66         $display("LEDR = %b, HEX0=%b, HEX1=%b, HEX2=%b, HEX3=%b, HEX4=%b, HEX5=%b", LEDR, HEX0, HEX1, HEX2, HEX3, HEX4,
67         #100
68         clk = 0;
69     end
70
71     #100
72     clk = 1;
73     KEY[0] = 1'b0;
74     #100
75     clk = 0;
76     SW[2:1] = 2'b00; // FREE_RUN && UP_DOWN
77     SW[9:4] = 6'b00_0010; // Divideby
78     KEY[0] = 1'b1;
79     $display("Free-run is low and up_down is low");
80     repeat(7) begin
81         #100
82         clk = 1;
83         $display("LEDR = %b, HEX0=%b, HEX1=%b, HEX2=%b, HEX3=%b, HEX4=%b, HEX5=%b", LEDR, HEX0, HEX1, HEX2, HEX3, HEX4,
84         #100
85         clk = 0;
86     end
87
88     #100
89     clk = 1;
90     KEY[0] = 1'b0;
91     #100
92     clk = 0;
93     $display("RESET THE COUNTER");
94     $display("LEDR = %b, HEX0=%b, HEX1=%b, HEX2=%b, HEX3=%b, HEX4=%b", LEDR, HEX0, HEX1, HEX2, HEX3, HEX4, HEX5);
95 endmodule

```

## Printout:

```

VSIM 35> run -all
# Initially LEDs are on and displays are 0
# LEDR = 111111111, HEX0=11000000, HEX1=11000000, HEX2=11000000, HEX3=11000000, HEX4=11000000, HEX5=11000000
# Free-run is high and up_down is high
# LEDR = 000000000, HEX0=11000000, HEX1=11000000, HEX2=11000000, HEX3=11000000, HEX4=11000000, HEX5=11000000
# LEDR = 000000000, HEX0=11000000, HEX1=11000000, HEX2=11000000, HEX3=11000000, HEX4=11000000, HEX5=11000000
# LEDR = 000000000, HEX0=11000000, HEX1=11000000, HEX2=11000000, HEX3=11000000, HEX4=11000000, HEX5=11000000
# LEDR = 000000000, HEX0=11111001, HEX1=11000000, HEX2=11000000, HEX3=11000000, HEX4=11000000, HEX5=11000000
# LEDR = 000000000, HEX0=11111001, HEX1=11000000, HEX2=11000000, HEX3=11000000, HEX4=11000000, HEX5=11000000
# LEDR = 000000000, HEX0=10100100, HEX1=11000000, HEX2=11000000, HEX3=11000000, HEX4=11000000, HEX5=11000000
# LEDR = 000000000, HEX0=10100100, HEX1=11000000, HEX2=11000000, HEX3=11000000, HEX4=11000000, HEX5=11000000
# LEDR = 000000000, HEX0=10100100, HEX1=11000000, HEX2=11000000, HEX3=11000000, HEX4=11000000, HEX5=11000000
# Free-run is high and up_down is low
# LEDR = 000000000, HEX0=10110000, HEX1=11000000, HEX2=11000000, HEX3=11000000, HEX4=11000000, HEX5=11000000
# LEDR = 000000000, HEX0=10110000, HEX1=11000000, HEX2=11000000, HEX3=11000000, HEX4=11000000, HEX5=11000000
# LEDR = 000000000, HEX0=10100100, HEX1=11000000, HEX2=11000000, HEX3=11000000, HEX4=11000000, HEX5=11000000
# LEDR = 000000000, HEX0=10100100, HEX1=11000000, HEX2=11000000, HEX3=11000000, HEX4=11000000, HEX5=11000000
# LEDR = 000000000, HEX0=10100100, HEX1=11000000, HEX2=11000000, HEX3=11000000, HEX4=11000000, HEX5=11000000
# LEDR = 000000000, HEX0=11111001, HEX1=11000000, HEX2=11000000, HEX3=11000000, HEX4=11000000, HEX5=11000000
# LEDR = 000000000, HEX0=11111001, HEX1=11000000, HEX2=11000000, HEX3=11000000, HEX4=11000000, HEX5=11000000
# LEDR = 000000000, HEX0=10100100, HEX1=11000000, HEX2=11000000, HEX3=11000000, HEX4=11000000, HEX5=11000000
# LEDR = 000000000, HEX0=10100100, HEX1=11000000, HEX2=11000000, HEX3=11000000, HEX4=11000000, HEX5=11000000
# LEDR = 000000000, HEX0=10100100, HEX1=11000000, HEX2=11000000, HEX3=11000000, HEX4=11000000, HEX5=11000000
# Free-run is low and up_down is high
# LEDR = 000000000, HEX0=11000000, HEX1=11000000, HEX2=11000000, HEX3=11000000, HEX4=11000000, HEX5=11000000
# LEDR = 000000000, HEX0=11000000, HEX1=11000000, HEX2=11000000, HEX3=11000000, HEX4=11000000, HEX5=11000000
# LEDR = 000000000, HEX0=11111001, HEX1=11000000, HEX2=11000000, HEX3=11000000, HEX4=11000000, HEX5=11000000
# LEDR = 000000000, HEX0=11111001, HEX1=11000000, HEX2=11000000, HEX3=11000000, HEX4=11000000, HEX5=11000000
# LEDR = 000000000, HEX0=10100100, HEX1=11000000, HEX2=11000000, HEX3=11000000, HEX4=11000000, HEX5=11000000
# LEDR = 000000000, HEX0=10100100, HEX1=11000000, HEX2=11000000, HEX3=11000000, HEX4=11000000, HEX5=11000000
# LEDR = 000000000, HEX0=10100100, HEX1=11000000, HEX2=11000000, HEX3=11000000, HEX4=11000000, HEX5=11000000
# LEDR = 000000000, HEX0=11111001, HEX1=11000000, HEX2=11000000, HEX3=11000000, HEX4=11000000, HEX5=11000000
# LEDR = 000000000, HEX0=11111001, HEX1=11000000, HEX2=11000000, HEX3=11000000, HEX4=11000000, HEX5=11000000
# LEDR = 000000000, HEX0=10100100, HEX1=11000000, HEX2=11000000, HEX3=11000000, HEX4=11000000, HEX5=11000000
# LEDR = 000000000, HEX0=10100100, HEX1=11000000, HEX2=11000000, HEX3=11000000, HEX4=11000000, HEX5=11000000
# LEDR = 000000000, HEX0=10100100, HEX1=11000000, HEX2=11000000, HEX3=11000000, HEX4=11000000, HEX5=11000000
# RESET THE COUNTER
# LEDR = 000000000, HEX0=11000000, HEX1=11000000, HEX2=11000000, HEX3=11000000, HEX4=11000000, HEX5=11000000

```

