University of Miami

EEN 414 Computer Organization and Design

EXPERIMENT #3. Stopwatch

Submitted by: Brandon Rubio, Isabela Bandrich, Rainier Young, Nikeem Dunkelly-Allen

On: 10/10/22

to : Dr. Onur Tigli

Objective

For this lab, we must design and verify the correct operation of a digital stopwatch using mixed modeling techniques in Verilog. Coding, verification, simulation, synthesis and timing analysis will be carried out using Xilinx ISE tools. Verified designs will be implemented on Digilent's NEXSYS4 FPGA board to show that the operation is working correctly. The Stopwatch should showcase minutes, seconds, and 1/10 seconds for a maximum of ten minutes. It should have a reset button as well as a start/pause button.

Verilog HDL and Testbench Code

MAIN CODE:

Counter:

```
21  module Counter(
22      input clk,
23      input reset,
24      input [3:0] num,
25      output [3:0] counter
26      );
27      reg [3:0] c;
28
29      always @(posedge clk or posedge reset)
30      begin
31        if(reset)
32           c <= 4'b0000;
33        else if(c==num)
34           c <= 4'b0000;
35        else
36           c <= c + 4'b0001;
37     end
38
39     assign counter = c;
40
41  endmodule
```

Time Counter:

```
module Time_Counter(
    input clk,
    input reset,
    input pause,
    output [3:0] num_1,
    output [3:0] num_2,
    output [3:0] num_3,
    output [3:0] num_4
    );
    wire [3:0] out;
    wire [2:0] minute = 2'b10;
    wire [3:0] second = 3'b100;
    wire [17:0] ten_second = 18'd250000;
    wire [3:0] double_sec = 4'b0101;
    wire [3:0] single_sec = 4'b1001;

    Time_Location #(18) b4(clk, reset, pause, ten_second, single_sec, out[0], num_4);
    Time_Location #(4) b3(out[0], reset, pause, second, single_sec, out[1], num_3);
    Time_Location #(4) b2(out[1], reset, pause, second, double_sec, out[2], num_2);
    Time_Location #(3) b1(out[2], reset, pause, minute, single_sec, out[3], num_1);

endmodule
```

Clock:

```
module Clock(
    input clk,
    input reset,
    input pause,
    input [nBit-1:0] limit,
    output reg out
    );
    parameter nBit = 18;
    reg [nBit-1:0] counter;

    always @(posedge clk or posedge reset or posedge pause)
    begin
      if(reset)
      begin
         counter <= 1'b0;
         out <= 1'b1;
      end
      else
      begin
         if(pause)
            counter <= counter;
         else
         begin
            if(counter==limit)
            begin
               counter <= 1'b0;
               out <= ~out;
               end
            else
            begin
               counter <= counter + 1'b1;
               out <= out;
            end
         end
      end
   end

endmodule
```

Time Location:

```
21  module Time_Location(
22      input clk,
23      input reset,
24      input pause,
25      input [nBit-1:0] limit,
26      input [3:0] stop,
27      output next,
28      output [3:0] num
29      );
30      parameter nBit = 18;
31      wire count;
32
33      Clock #(nBit) place(clk, reset, pause, limit, count);
34      Counter increment(count, reset, stop, num);
35      assign next = count;
36
37  endmodule
```

Seven Segment Decoder:

```
21  module Sev_Seg_Decoder(
22      input [3:0] num_in,
23      output [7:1] sev_seg_leds
24      );
25     reg [7:1] tmp;
26     always @(*)
27     begin
28        case(num_in)
29           4'h0: tmp = 7'b1000000;
30           4'h1: tmp = 7'b1111001;
31           4'h2: tmp = 7'b0100100;
32           4'h3: tmp = 7'b0110000;
33           4'h4: tmp = 7'b0011001;
34           4'h5: tmp = 7'b0010010;
35           4'h6: tmp = 7'b0000010;
36           4'h7: tmp = 7'b1111000;
37           4'h8: tmp = 7'b0000000;
38           4'h9: tmp = 7'b0010000;
39           default: tmp = 7'b0111111;
40        endcase
41     end
42
43     assign sev_seg_leds[7:1] = tmp;
44
45  endmodule
```

Seven Segment Display with Clock:

```
21  module sev_seg_with_clk(
22      input [3:0] num_1,
23      input [3:0] num_2,
24      input [3:0] num_3,
25      input [3:0] num_4,
26      input clk,
27      input reset,
28      output [7:1] sev_seg_leds,
29      output[3:0] led_disable,
30      output reg [3:0] led_enable
31      );
32      reg [2:0] sel;
33      reg [3:0] bus;
34      assign led_disable = 4'b1111;

36     always @ (posedge clk or posedge reset)
37     begin
38       if (reset)
39       begin
40          sel = 3'b000;
41          led_enable = 4'b1111;
42       end
43       else
44       begin
45          sel[2] = sel[1] & sel[0];
46          sel[1] = sel[1] ^ sel[0];
47          sel[0] = ~sel[0];
48          if (sel==3'b001)
49          begin
50             led_enable = 4'b0111;
51             bus = num_1;
52          end
53          else if (sel==3'b010)
54          begin
55             led_enable = 4'b1011;
56             bus = num_2;
57          end
58          else if (sel==3'b011)
59          begin
60             led_enable = 4'b1101;
61             bus = num_3;
62          end
63          else if (sel==3'b100)
64          begin
65             led_enable = 4'b1110;
66             bus = num_4;
67          end
68       end
69    end
70
71    Sev_Seg_Decoder dec1(bus, sev_seg_leds);
72
73  endmodule
```

Clock Stop:

```
21  module Stop_Clock(
22      input clk,
23      input reset,
24      input pause,
25      output reg out
26      );
27      parameter nBit = 18;
28      reg [nBit-1:0] counter;
29
30      always @(posedge clk or posedge reset)
31      begin
32        if(reset)
33        begin
34           counter <= 1'b0;
35           out <= 1'b1;
36        end
37        else
38        begin
39           if(pause)
40           begin
41           end
42           else
43              counter <= counter;
44        end
45      end
46
47  endmodule
```

Seven Segment Display with Clock - Top Module:

```
21  module sev_seg_with_clk_top(
22      input clk_main,
23      input reset,
24      input pause,
25      output [7:1] sev_seg_leds,
26      output [3:0] led_disable,
27      output reg [3:0] led_enable
28      );
29      wire [3:0] num_1;
30      wire [3:0] num_2;
31      wire [3:0] num_3;
32      wire [3:0] num_4;
33      wire clk_slw;
34      wire clk_stop;
35      wire [3:0] display;
36      reg [14:0] clk_div;
37
38      Time_Counter counter(clk_slw, reset, pause, num_1, num_2, num_3, num_4);
39      Stop_Clock hold(clk_slw, reset, pause, clk_stop);
40
41      always @(*)
42      begin
43        if(pause==1)
44           led_enable = clk_stop?4'b1111:display;
45        else
46           led_enable = display;
47      end
48
49      sev_seg_with_clk dd(num_1, num_2, num_3, num_4, clk_div[14], reset, sev_seg_leds, led_disable, display);
50      ip_clk_div clk_5M(clk_main, clk_slw);
51
52      always @(posedge clk_slw)
53      begin
54        if(reset)
55           clk_div <= 15'd0;
56        else
57           clk_div <= clk_div + 15'd1;
58      end
59
60  endmodule
```

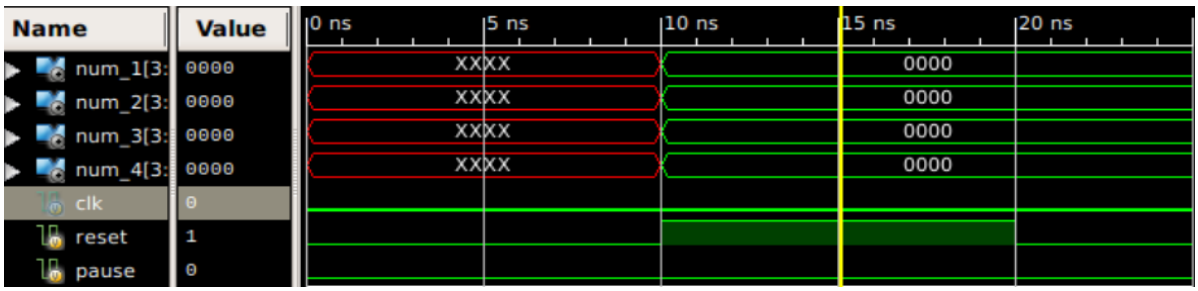
TESTBENCH:

```
module tb_StopMatch;

    // Inputs
    reg clk;
    reg reset;
    reg pause;

    // Outputs
    wire [3:0] num_1;
    wire [3:0] num_2;
    wire [3:0] num_3;
    wire [3:0] num_4;

    // Instantiate the Unit Under Test (UUT)
    Time_Counter uut (
        .clk(clk),
        .reset(reset),
        .pause(pause),
        .num_1(num_1),
        .num_2(num_2),
        .num_3(num_3),
        .num_4(num_4)
    );
    integer i, j, k;

    initial begin
        // Initialize Inputs
        clk = 0;
        reset = 0;
        pause = 0;
        #10;

        clk = 0;
        reset = 1;
        pause = 0;
        #10;

        clk = 0;
        reset = 0;
        pause = 0;
        #10;
```

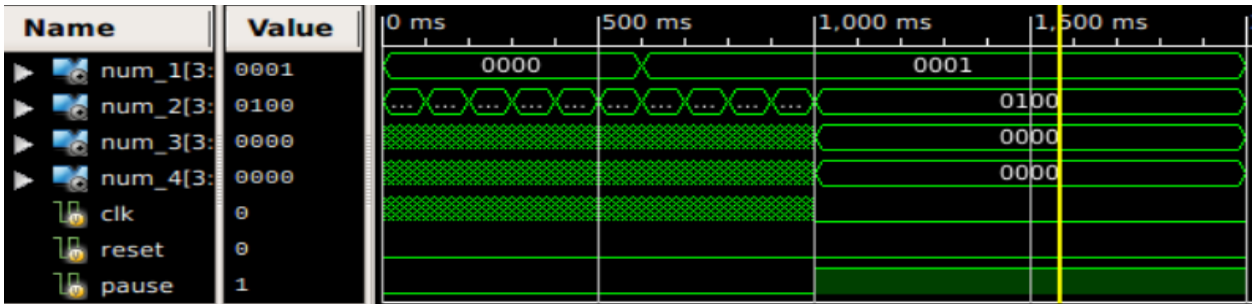
```
67          for(i=0;i<31'd500000000;i=i+1)
68             begin
69                clk = 0;
70                #1;
71                clk = 1;
72                #1;
73             end
74
75          clk = 1;
76          reset = 0;
77          pause = 0;
78          #10;
79
80          for(i=0;i<31'd500000;i=i+1)
81             #1 clk = ~clk;
82
83          clk = 0;
84          reset = 0;
85          pause = 0;
86          #100;
87
88          for(j=0;j<31'd2500;j=j+1)
89             #1 clk = ~clk;
90
91          clk = 0;
92          reset = 0;
93          pause = 1;
94          #100;
95
96          for(k=0;k<31'd2500;k=k+1)
97             #1 clk = ~clk;
98       end
99
100   endmodule
```

Simulation Results

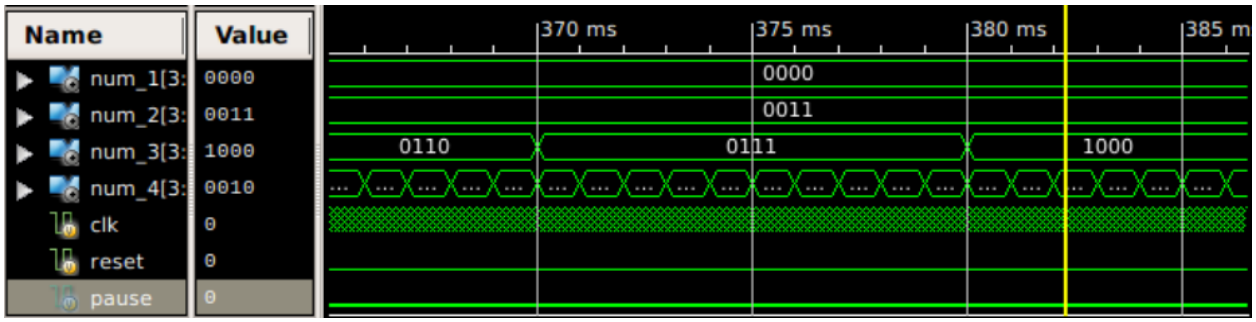
(Reset high forces counter to 0)

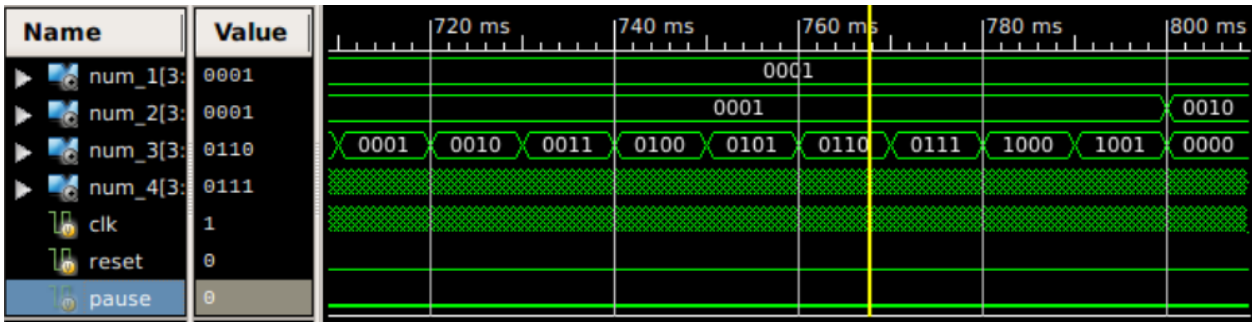


(Pause high forces counter to pause/become constant)



(Normal Counting)





Pad2pad timing constraints

	Met	Constraint	Check	Worst Case Slack	Best Case Achievable	Timing Errors	Timing Score
1	Yes	TS_pad2pad = MAXDELAY FROM TIMEGRP "inputs" TO TIMEGRP "o...	MAXDE...	6.933ns	13.067ns	0	0

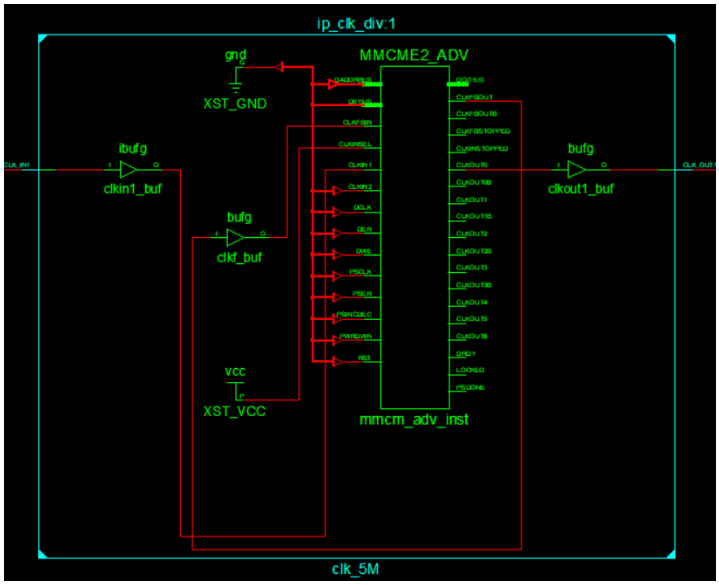
Device Utilization Summary

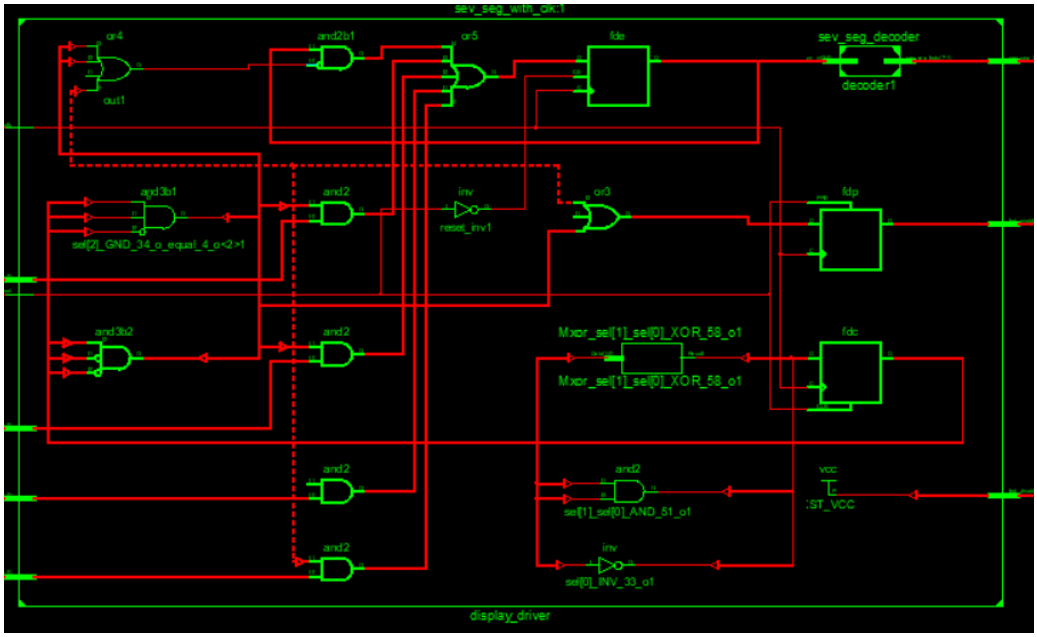
Seven Segment Display:

Device Utilization Summary				[-]
Slice Logic Utilization	**Used**	**Available**	**Utilization**	**Note(s)**
Number of Slice Registers	132	126,800	1%	
Number used as Flip Flops	103			
Number used as Latches	29			
Number used as Latch-thrus	0			
Number used as AND/OR logics	0			
Number of Slice LUTs	139	63,400	1%	
Number used as logic	138	63,400	1%	
Number using O6 output only	63			
Number using O5 output only	13			
Number using O5 and O6	62			
Number used as ROM	0			
Number used as Memory	0	19,000	0%	
Number used exclusively as route-thrus	1			
Number with same-slice register load	0			
Number with same-slice carry load	1			
Number with other load	0			
Number of occupied Slices	113	15,850	1%	
Number of LUT Flip Flop pairs used	192			
Number with an unused Flip Flop	66	192	34%	
Number with an unused LUT	53	192	27%	
Number of fully used LUT-FF pairs	73	192	38%	
Number of unique control sets	94			
Number of slice register sites lost to control set restrictions	628	126,800	1%	
Number of bonded IOBs	18	210	8%	
Number of LOCed IOBs	18	18	100%	
Number of RAMB36E1/FIFO36E1s	0	135	0%	
Number of RAMB18E1/FIFO18E1s	0	270	0%	
Number of BUFG/BUFGCTRLs	2	32	6%	
Number used as BUFGs	2			
Number used as BUFGCTRLs	0			
Number of IDELAYE2/IDELAYE2_FINEDELAYs	0	300	0%	
Number of ILOGICE2/ILOGICE3/ISERDESE2s	0	300	0%	
Number of ODELAYE2/ODELAYE2_FINEDELAYs	0			
Number of OLOGICE2/OLOGICE3/OSERDESE2s	0	300	0%	
Number of PHASER_IN/PHASER_IN_PHYs	0	24	0%	
Number of PHASER_OUT/PHASER_OUT_PHYs	0	24	0%	
Number of BSCANs	0	4	0%	
Number of BUFHCEs	0	96	0%	
Number of BUFRs	0	24	0%	
Number of CAPTUREs	0	1	0%	
Number of DNA_PORTs	0	1	0%	
Number of DSP48E1s	0	240	0%	
Number of EFUSE_USRs	0	1	0%	
Number of FRAME_ECCs	0	1	0%	
Number of IBUFDS_GTE2s	0	4	0%	
Number of ICAPs	0	2	0%	
Number of IDELAYCTRLs	0	6	0%	
Number of IN_FIFOs	0	24	0%	
Number of MMCME2_ADVs	1	6	16%	
Number of OUT_FIFOs	0	24	0%	
Number of PCIE_2_1s	0	1	0%	
Number of PHASER_REFs	0	6	0%	
Number of PHY_CONTROLs	0	6	0%	
Number of PLLE2_ADVs	0	6	0%	
Number of STARTUPs	0	1	0%	
Number of XADCs	0	1	0%	
Average Fanout of Non-Clock Nets	2.70			

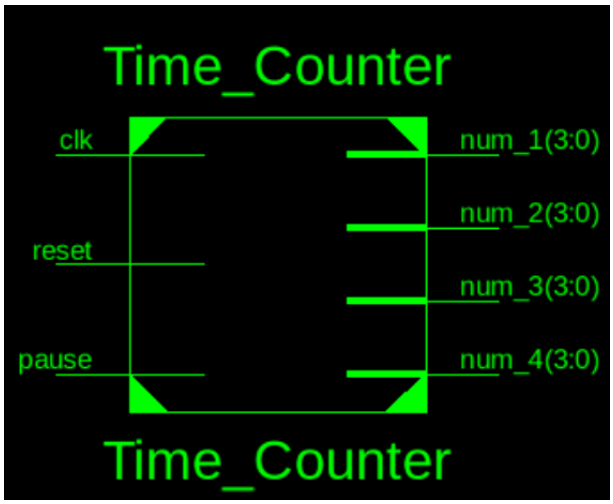
RTL Schematic and Technology Schematic

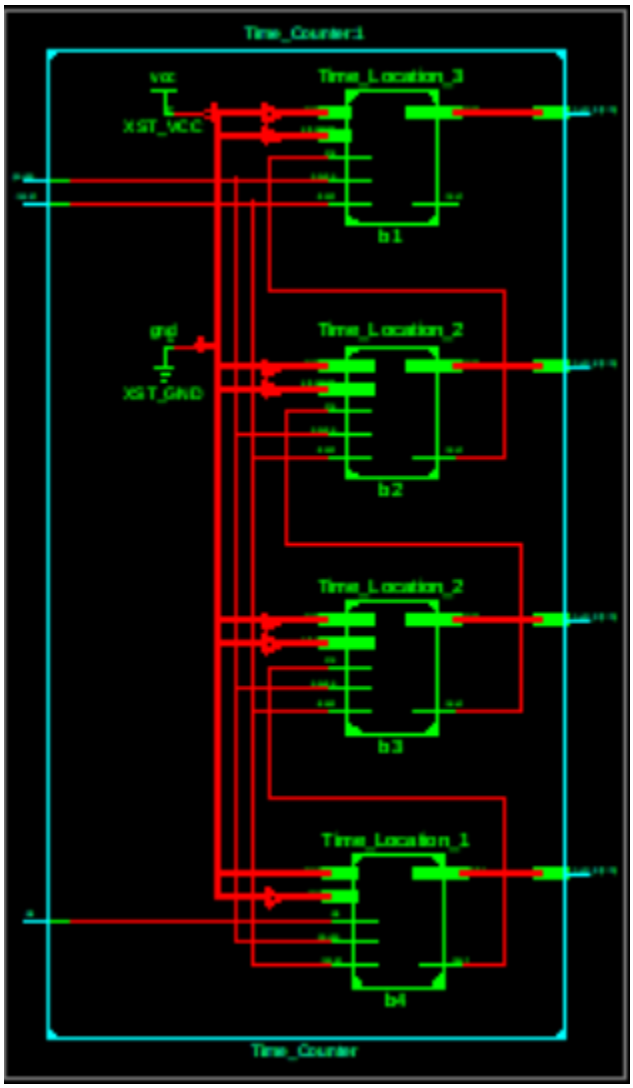
(Ip_clk_div)

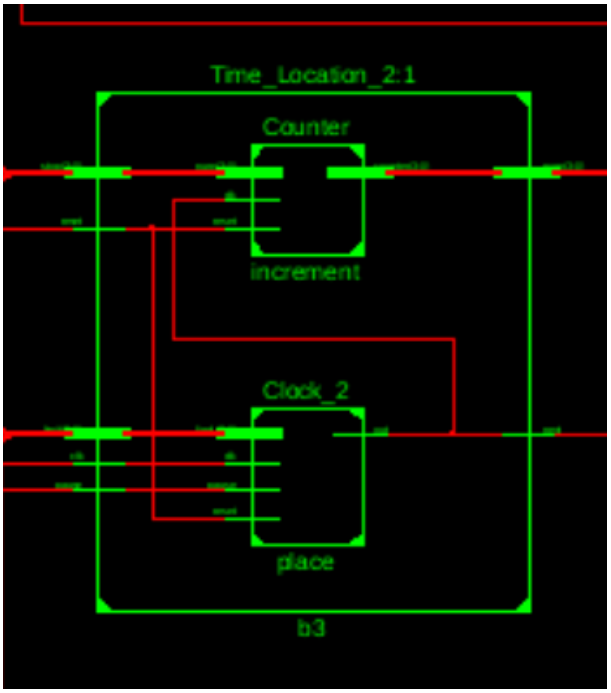


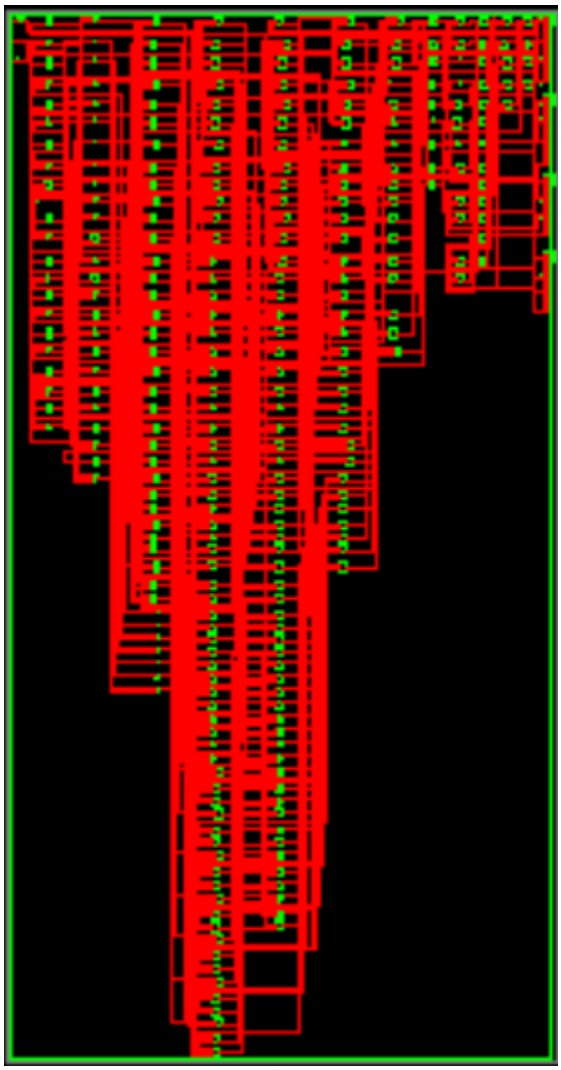


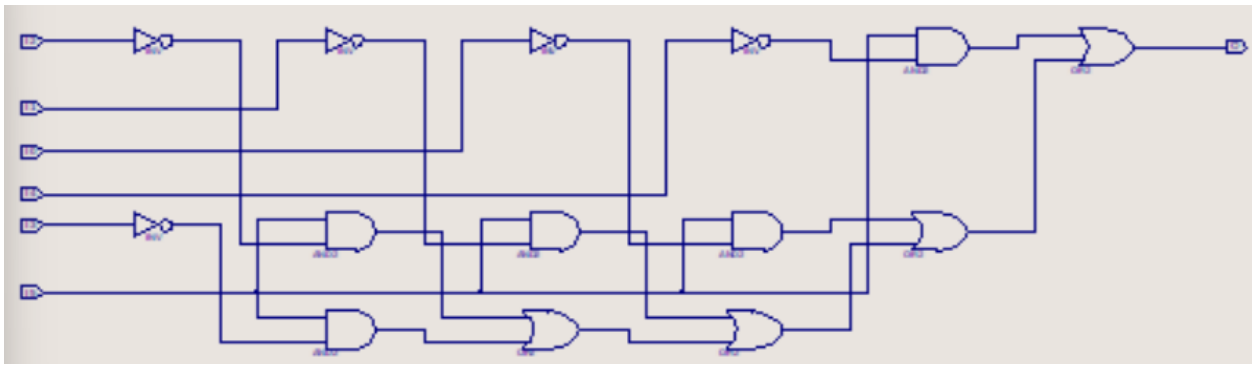
(Time Counter)


Time_Counter
clk
reset
pause
num_1(3:0)
num_2(3:0)
num_3(3:0)
num_4(3:0)
Time_Counter

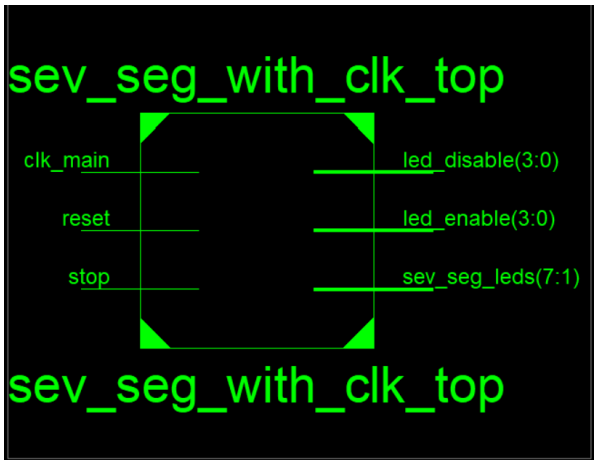

Time_Counter:1
vcc
XST_VCC
Time_Location_3
b1
gnd
XST_GND
Time_Location_2
b2
Time_Location_2
b3
Time_Location_1
b4
Time_Counter

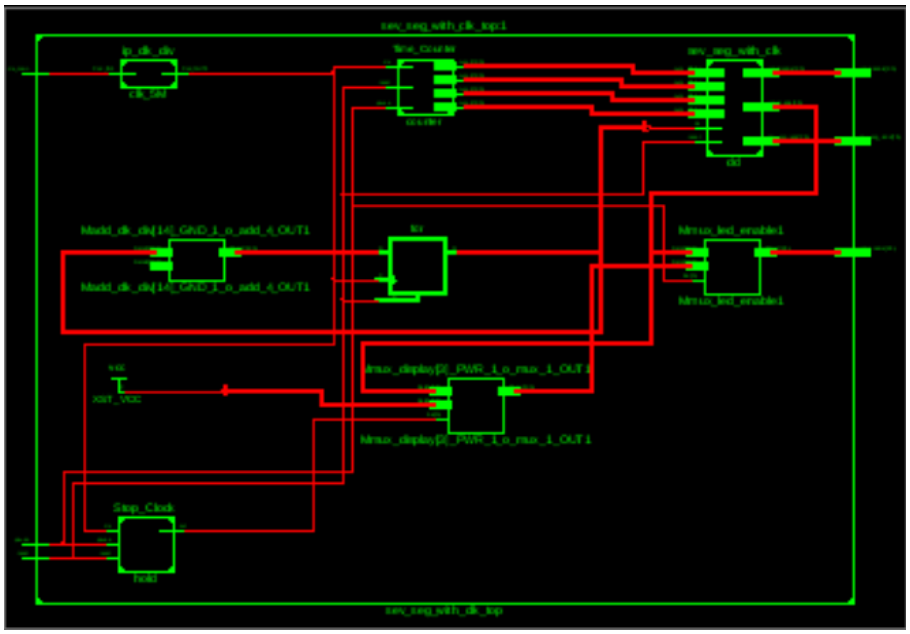

Time_Location_2:1
Counter
increment
Clock_2
place
b3

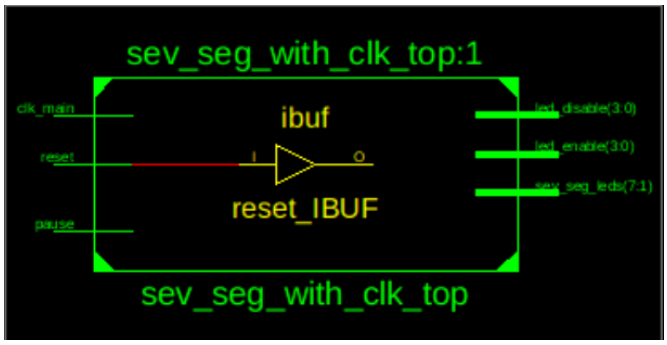


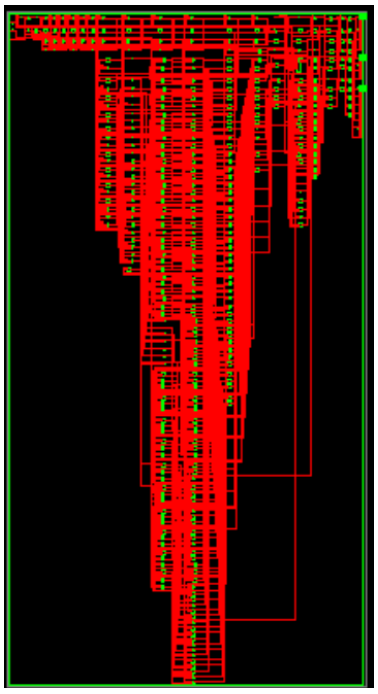


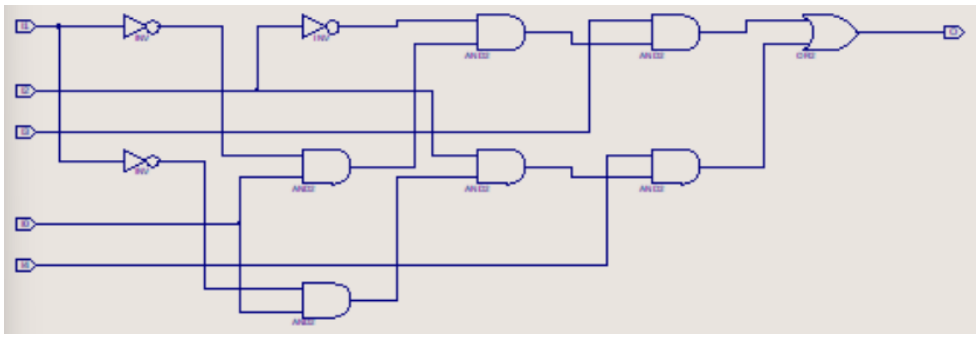
(Seven Segment Display with Clock - Top Module)


sev_seg_with_clk_top
clk_main
reset
stop
led_disable(3:0)
led_enable(3:0)
sev_seg_leds(7:1)
sev_seg_with_clk_top




sev_seg_with_clk_top:1
clk_main
reset
pause
ibuf
I
O
reset_IBUF
led_disable(3:0)
led_enable(3:0)
sev_seg_leds(7:1)
sev_seg_with_clk_top





Post Place and Route Static Timing

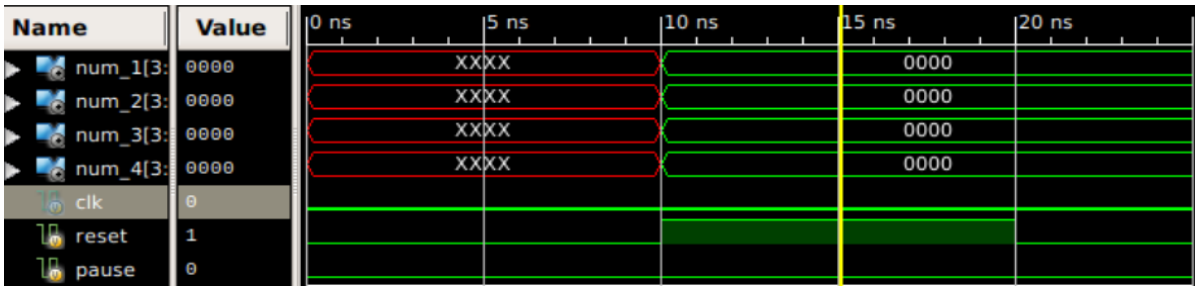
(Seven Segment Display with Clock - Top Module)

```
INFO:Timing:2698 - No timing constraints found, doing default enumeration.

INFO:Timing:3412 - To improve timing, see the Timing Closure User Guide (UG612).
INFO:Timing:2752 - To get complete path coverage, use the unconstrained paths
   option. All paths that are not constrained will be reported in the
   unconstrained paths section(s) of the report.
INFO:Timing:3339 - The clock-to-out numbers in this timing report are based on
   a 50 Ohm transmission line loading model.  For the details of this model,
   and for more information on accounting for different loading conditions,
   please see the device datasheet.
```

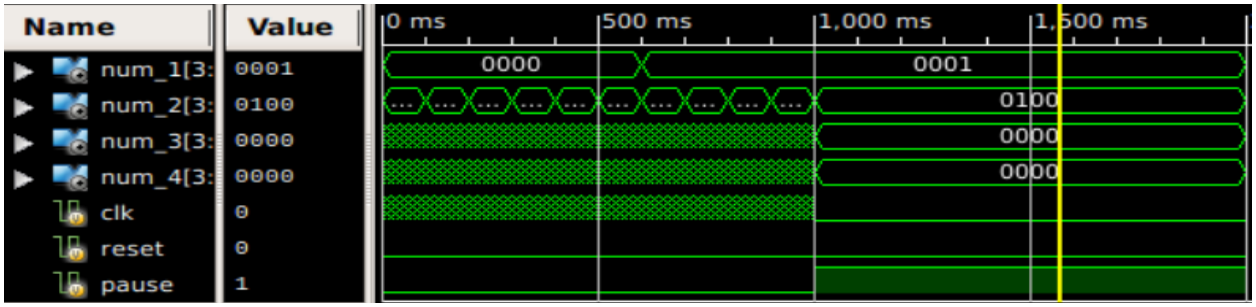
```
Data Sheet report:
-----------------
All values displayed in nanoseconds (ns)

Clock to Setup on destination clock clk_main
--------------+---------+---------+---------+---------+
              | Src:Rise| Src:Fall| Src:Rise| Src:Fall|
Source Clock  |Dest:Rise|Dest:Rise|Dest:Fall|Dest:Fall|
--------------+---------+---------+---------+---------+
clk_main      |    4.442|         |         |         |
pause         |    8.815|   13.553|         |         |
reset         |   10.874|   13.553|         |         |
--------------+---------+---------+---------+---------+

Clock to Setup on destination clock pause
--------------+---------+---------+---------+---------+
              | Src:Rise| Src:Fall| Src:Rise| Src:Fall|
Source Clock  |Dest:Rise|Dest:Rise|Dest:Fall|Dest:Fall|
--------------+---------+---------+---------+---------+
clk_main      |         |         |   -1.236|         |
pause         |         |         |    1.269|    2.744|
reset         |         |         |    3.438|    3.438|
--------------+---------+---------+---------+---------+

Clock to Setup on destination clock reset
--------------+---------+---------+---------+---------+
              | Src:Rise| Src:Fall| Src:Rise| Src:Fall|
Source Clock  |Dest:Rise|Dest:Rise|Dest:Fall|Dest:Fall|
--------------+---------+---------+---------+---------+
clk_main      |         |         |   -1.236|         |
pause         |         |         |    1.362|    2.744|
reset         |         |         |    1.484|    2.744|
--------------+---------+---------+---------+---------+

Pad to Pad
--------------+--------------+---------+
Source Pad    |Destination Pad|  Delay  |
--------------+--------------+---------+
pause         |led_enable<0> |    8.188|
pause         |led_enable<1> |    7.719|
pause         |led_enable<2> |    8.027|
pause         |led_enable<3> |    7.791|
--------------+--------------+---------+
```

Post Place and Route Simulation Results

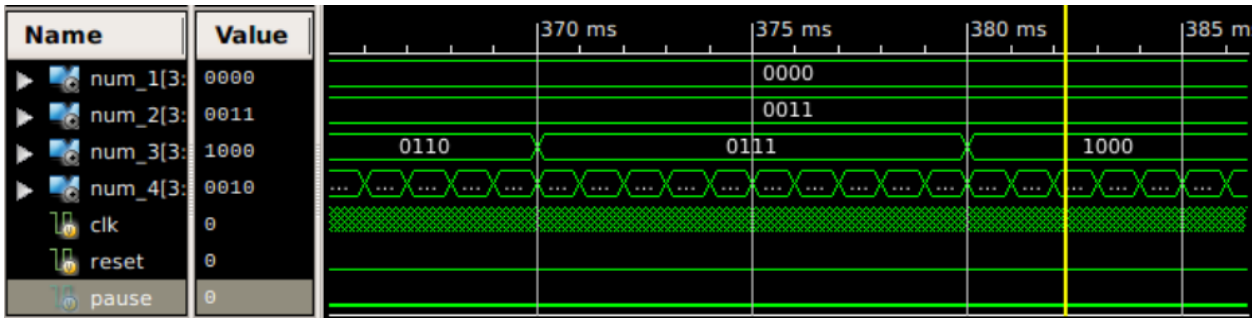
(Reset high forces counter to 0)

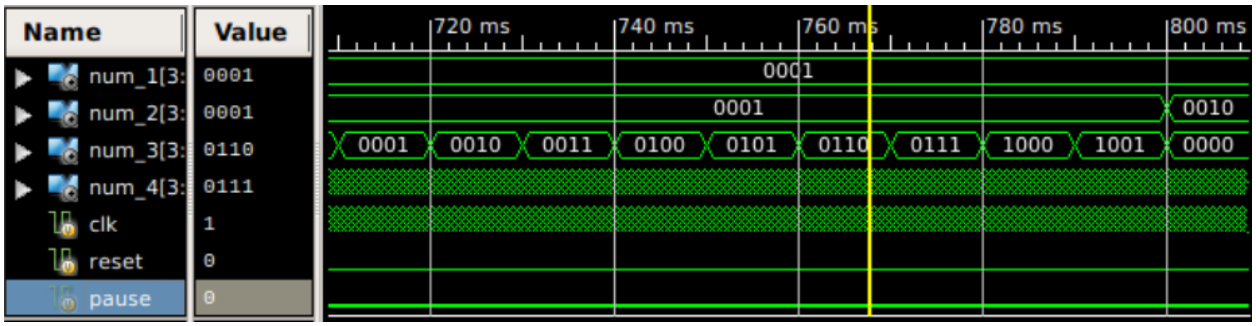


(Pause high forces counter to pause/become constant)

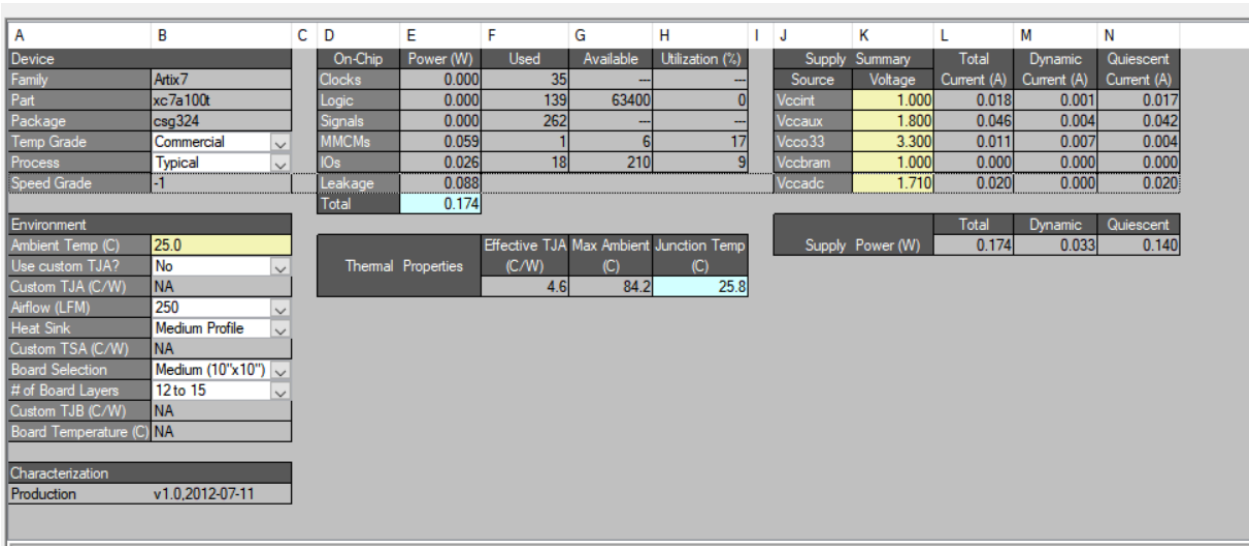


(Normal Counting)





<u>Power Data</u>



Device	
Family	Artix7
Part	xc7a100t
Package	csg324
Temp Grade	Commercial
Process	Typical
Speed Grade	-1

Environment	
Ambient Temp (C)	25.0
Use custom TJA?	No
Custom TJA (C/W)	NA
Airflow (LFM)	250
Heat Sink	Medium Profile
Custom TSA (C/W)	NA
Board Selection	Medium (10"x10")
# of Board Layers	12 to 15
Custom TJB (C/W)	NA
Board Temperature (C)	NA

Characterization	
Production	v1.0,2012-07-11

On-Chip	Power (W)	Used	Available	Utilization (%)
Clocks	0.000	35	--	--
Logic	0.000	139	63400	0
Signals	0.000	262	--	--
MMCMs	0.059	1	6	17
IOs	0.026	18	210	9
Leakage	0.088			
Total	0.174			

Thermal Properties	Effective TJA (C/W)	Max Ambient (C)	Junction Temp (C)
	4.6	84.2	25.8

Supply Summary		Total	Dynamic	Quiescent
Source	Voltage	Current (A)	Current (A)	Current (A)
Vccint	1.000	0.018	0.001	0.017
Vccaux	1.800	0.046	0.004	0.042
Vcco33	3.300	0.011	0.007	0.004
Vccbram	1.000	0.000	0.000	0.000
Vccadc	1.710	0.020	0.000	0.020

Supply Power (W)	Total	Dynamic	Quiescent
	0.174	0.033	0.140

Conclusions

In this lab we utilized a 5 MHz clock to drive a modulo counter and decoded those values for display on 7 segments with the Nexys 4 FPGA board. The biggest challenge in this lab was reconfiguring the given seven segment display code from lab 2 to work in this different case. The top level design needed to be changed to display on additional displays as well as to include a counter for driving the displayed values.

References

[1]: Nexys4™ FPGA Board Reference Manual, Nexys-4 rev. B; Revised November 19, 2013 by Diligent Beyond Theory

[2]: Introduction to FPGA ECE414 - Fall 2020 pdf by Randil Gajasinghe and Prof. Onur Tigli

[3]: ECE 414 Computer Organization and Design - Experiment 2. Adder/Subtractor with 7-Segment Display