Term Project

EGCP 446: Advanced Digital Design using Verilog HDL

Total points – ***130***

Submit your completed Proposal via Canvas on/before the due date.

Please recall that academic dishonesty will not be tolerated. The penalty for academic dishonesty will result in a score of “0 (zero)” for the Proposal and a possible final letter grade of “F” in the course, along with an entry in the student records.

This is a group assignment as per the term project groups. Also, include the team contribution sheet.You can use any format of your liking (do not use line spacing more than 1.2).Include the following in the report

- **[5 pts]** Team contribution sheet (if applicable) – the template is available on canvas under the project module

- **Your name and your partner’s name/names**: Jeremy Escobar, Duy Nguyen, Spencer Williams

- **[8 pts] Introduce and explain the project :**

The objective of this project is to keep scores for two rival basketball teams and keep time using a shot clock countdown timer from 12:00 to 0:00. Three buttons will be used to assign one, two, and three points. A ball possession (team) switch will be used to keep track of the game between the two teams and assign points. Every score will be recorded as either 1,2 or 3 points depending on the type of score made.. These functionalities allow us to mimic a basketball scoreboard.

- **[6 pts]** System diagram of the project (it is not schematic from Vivado, if it is using only one module, make a block diagram with input and output) and state diagram

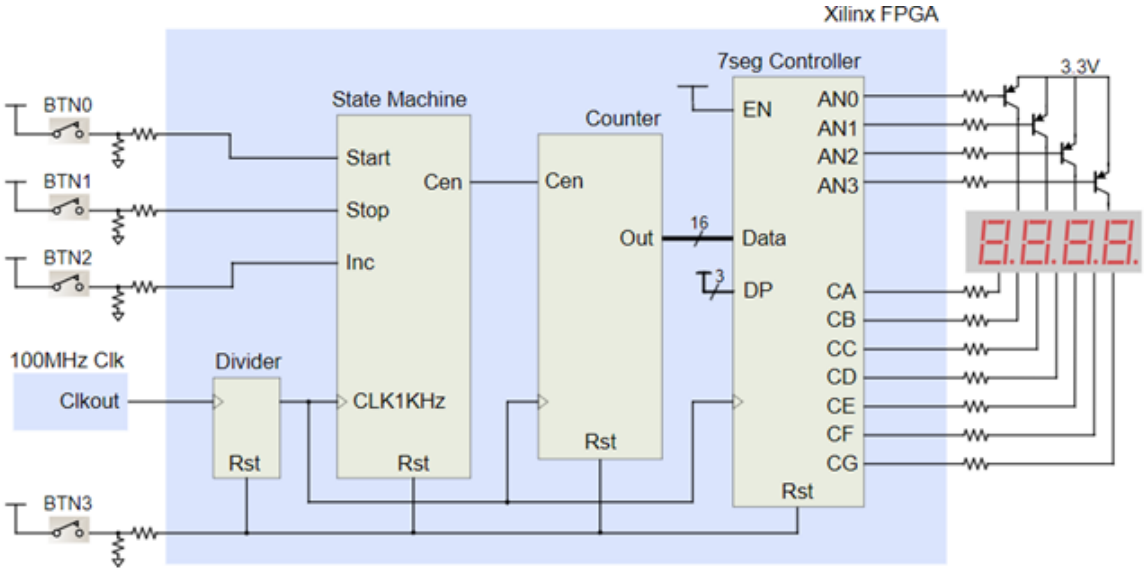
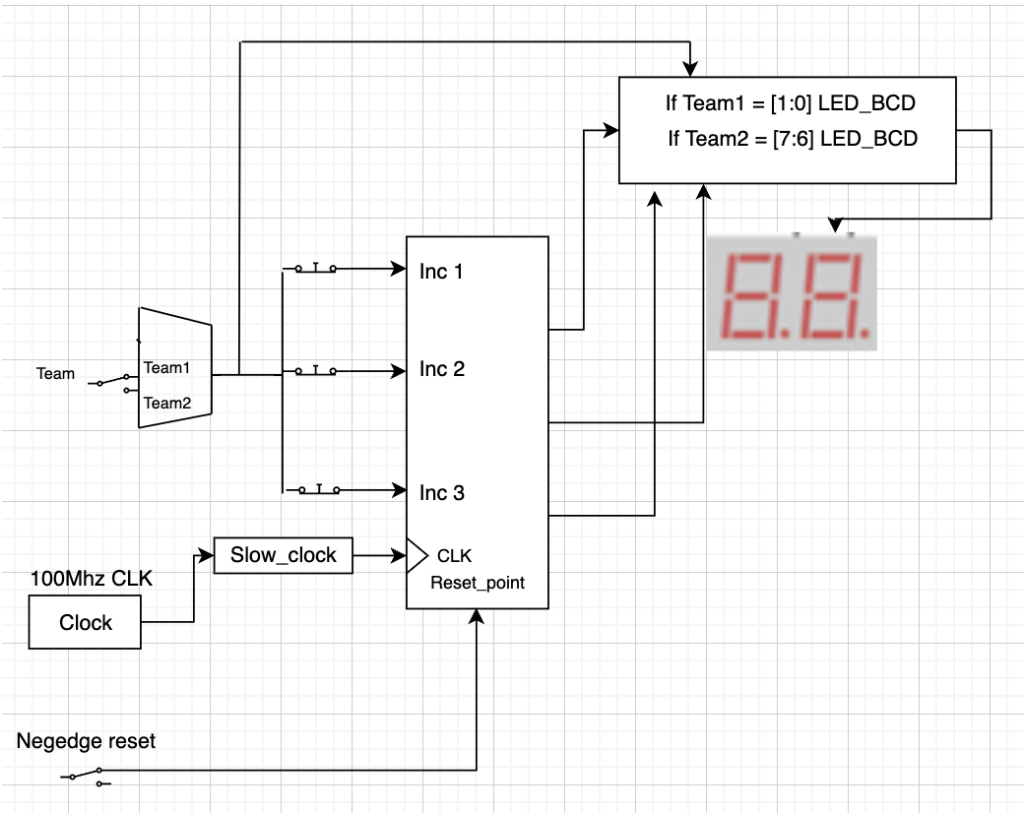
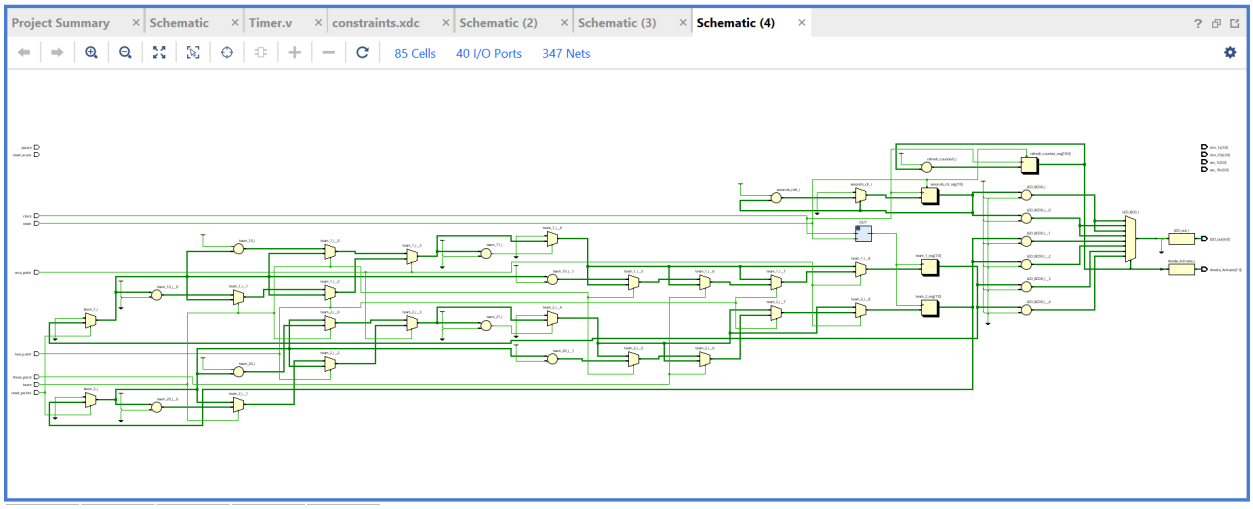


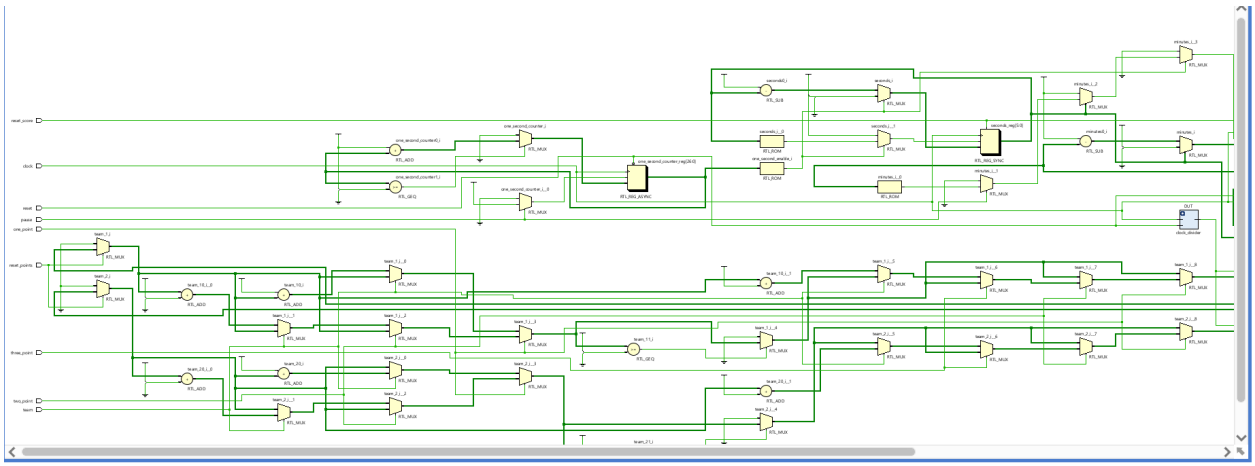
Figure 1. Stopwatch block diagram.

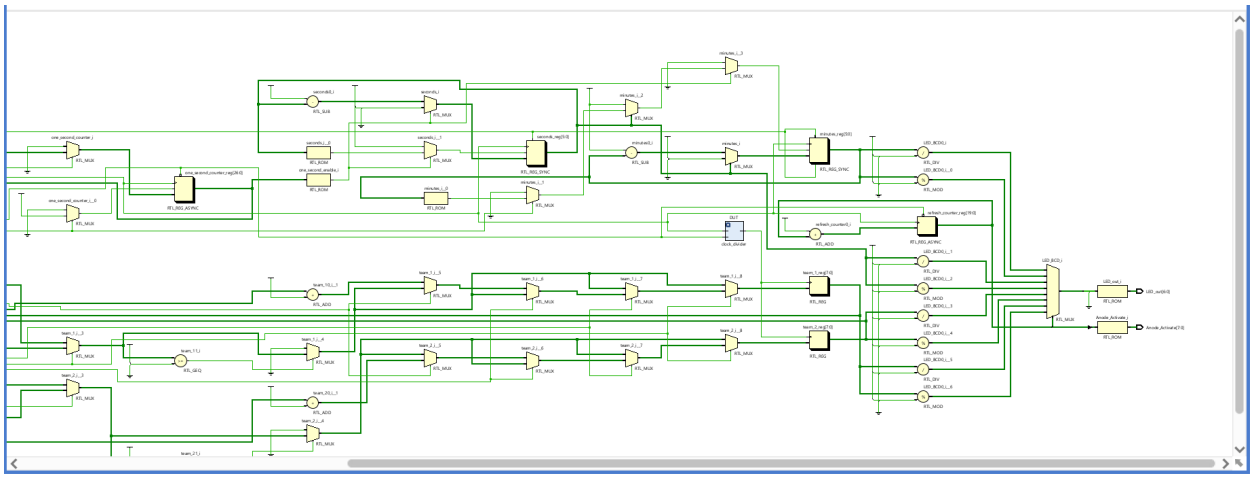


- **[2 pts]** RTL schematic (had to do multiple screenshots, schematic is large).

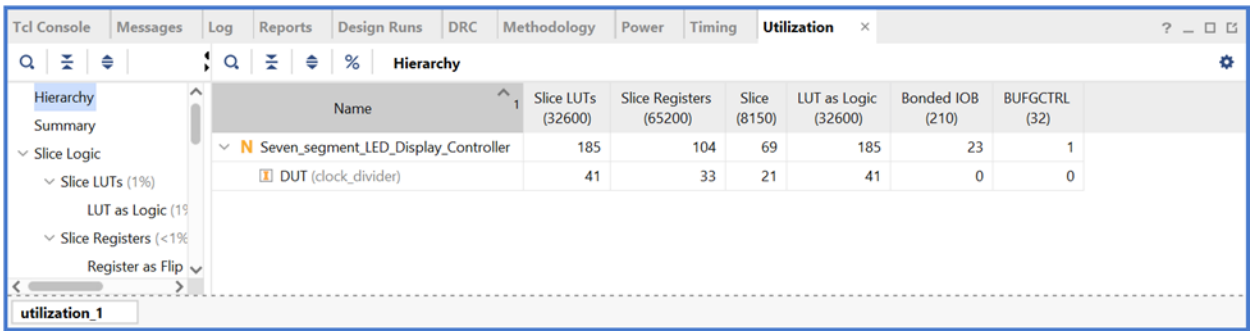


Zoomed

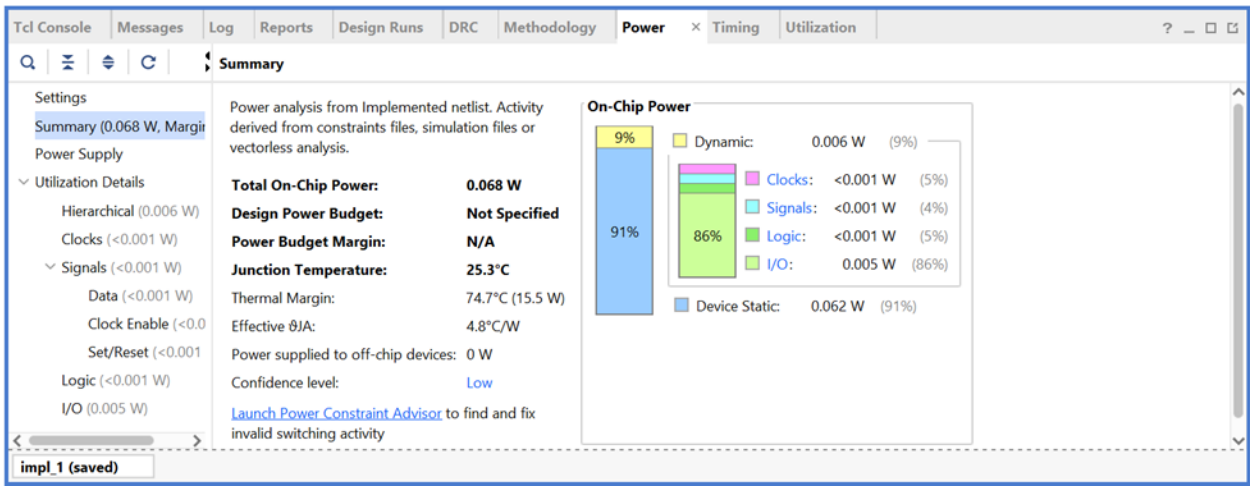




- **[2 pts]** Area report (after implementation)

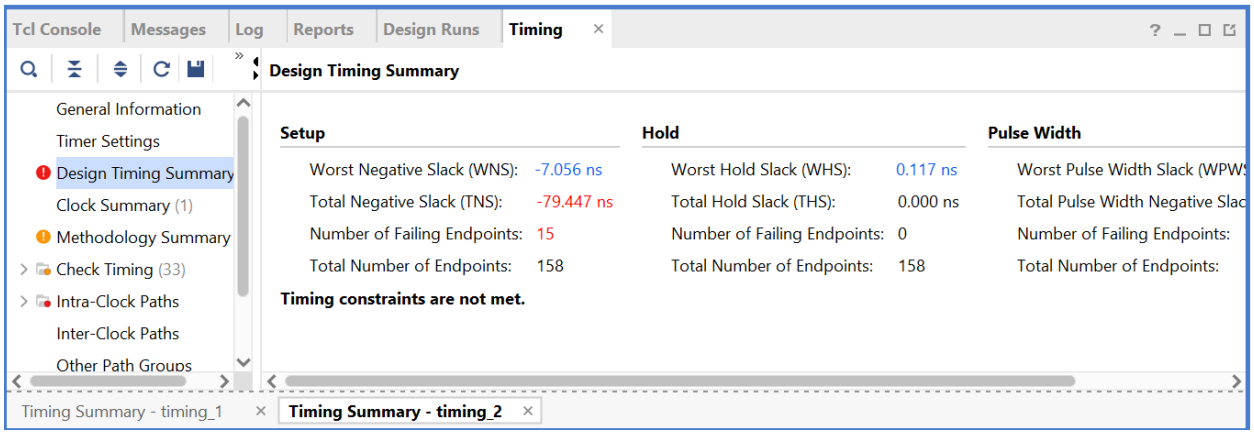


- **[2 pts]** Power report (after implementation)

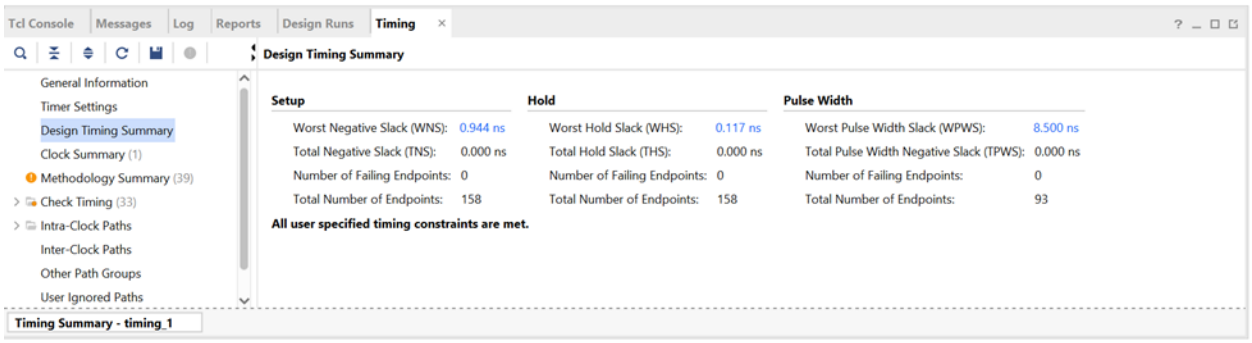


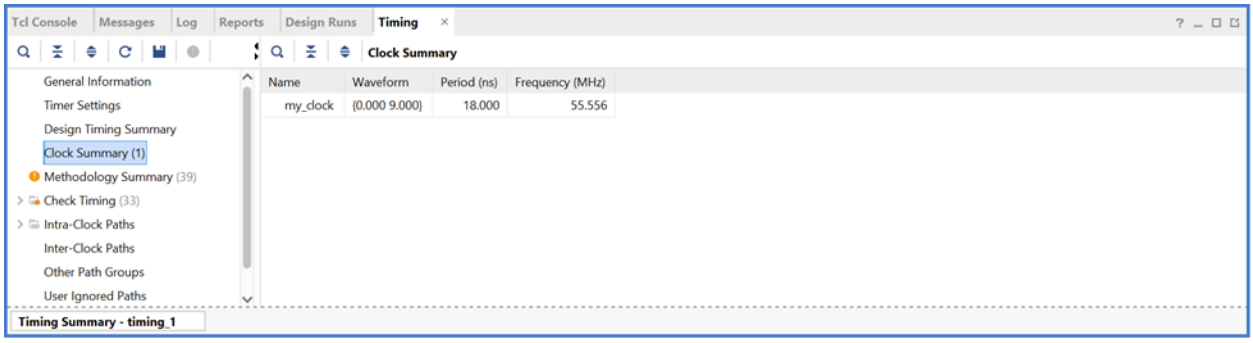
- **[5 pts]** Static timing analysis report, what is the maximum frequency used in your design?

Before



After


Tcl Console
Messages
Log
Reports
Design Runs
Timing
Design Timing Summary
General Information
Timer Settings
Design Timing Summary
Clock Summary (1)
Methodology Summary (39)
Check Timing (33)
Intra-Clock Paths
Inter-Clock Paths
Other Path Groups
User Ignored Paths
Setup
Worst Negative Slack (WNS): 0.944 ns
Total Negative Slack (TNS): 0.000 ns
Number of Failing Endpoints: 0
Total Number of Endpoints: 158
Hold
Worst Hold Slack (WHS): 0.117 ns
Total Hold Slack (THS): 0.000 ns
Number of Failing Endpoints: 0
Total Number of Endpoints: 158
Pulse Width
Worst Pulse Width Slack (WPWS): 8.500 ns
Total Pulse Width Negative Slack (TPWS): 0.000 ns
Number of Failing Endpoints: 0
Total Number of Endpoints: 93
All user specified timing constraints are met.
Timing Summary - timing_1


Tcl Console
Messages
Log
Reports
Design Runs
Timing
Clock Summary
General Information
Timer Settings
Design Timing Summary
Clock Summary (1)
Methodology Summary (39)
Check Timing (33)
Intra-Clock Paths
Inter-Clock Paths
Other Path Groups
User Ignored Paths
Name
Waveform
Period (ns)
Frequency (MHz)
my_clock
{0.000 9.000}
18.000
55.556
Timing Summary - timing_1

• **[20 pts]** *Verilog code (include as a text in the appendix.* Also, submit separate .v files (all modules) (testbench file if applicable) on google drive https://drive.google.com/drive/folders/1GMYU-FaxRc8-CIPhLNZoZOAyiJoIOMpr?usp=share_link. I made a separate folder for each group.

❖ **Timer.v file**

```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
// school: CSUF
// Engineers: Jeremy Escobar, Duy Nguyen, Spencer Williams
//
// Create Date: 11/16/2021 07:26:26 PM
// Design Name: Basket ball score board
// Module Name: timer, scoreboard
// Project Name: 446 final project
// Target Devices: FPGA Nexys A7
//
// Additional Comments: group project
//
//////////////////////////////////////////////////////////////////////////////////

module Seven_segment_LED_Display_Controller(
    input clock, // 100 Mhz clock source on Basys 3 FPGA
    input reset, // reset
    input reset_points, // reset
    input reset_score,
    input team,
    input one_point,
    input two_point,
    input three_point,
    input pause,
    output reg [7:0] Anode_Activate, // anode signals of the 7-segment LED display
    output reg [6:0] LED_out// cathode patterns of the 7-segment LED display
    );

    reg [26:0] one_second_counter; // counter for generating 1 second clock enable
    wire one_second_enable;        // one second enable for counting numbers
    reg [19:0] refresh_counter; // 20-bit for creating 10.5ms refresh period or 380Hz refresh rate
    wire [2:0] LED_activating_counter;  //3 bits
    reg [15:0] displayed_number;    // counting number to be displayed
```

```
    reg [7:0] team_1;
    reg [7:0] team_2;
    reg [5:0] seconds;   // 0
    reg [5:0] minutes;   // 0

clock_divider DUT (.clk(clock),.reset(reset), .sclk(slow_clock));
always @(posedge clock or negedge reset)
    begin
        if(reset==0)
            one_second_counter <= 0;
        else if (pause)
            displayed_number = displayed_number;
        else begin
            if(one_second_counter>=99999999)
                one_second_counter <= 0;
            else
               one_second_counter <= one_second_counter + 1;
        end
    end
    assign one_second_enable = (one_second_counter==99999999)?1:0;

always @(posedge clock)
    begin

     if(reset_score) begin //reset timer  here
            minutes <= 12;
            seconds <= 00;//default to 12:00 min once enabled
        end
        else if (one_second_enable == 1)
                seconds <= seconds - 1;
        else if (seconds == 0) begin
                seconds <= 59;
                minutes <= minutes - 1;
        end
        else if (pause)
            seconds=seconds;
end

    always @(posedge clock or negedge reset)
    begin
        if  (reset==0)
            refresh_counter <= 0;
```

```
    else
        refresh_counter <= refresh_counter + 1;
end

assign LED_activating_counter = refresh_counter[19:17];
// anode activating signals for 4 LEDs, digit period of 2.6ms
// decoder to generate anode signals
always @(*)
begin
    case(LED_activating_counter)
    3'b000: begin
        Anode_Activate = 8'b11111110;
        // activate LED1 and Deactivate LED2, LED3, LED4
         LED_BCD = minutes / 10;//10s place
        // the first digit of the 16-bit number
          end
    3'b001: begin
        Anode_Activate = 8'b11111101;
        // activate LED2 and Deactivate LED1, LED3, LED4
        LED_BCD = minutes % 10;//1s place
        // the second digit of the 16-bit number
          end
    3'b010: begin
        Anode_Activate = 8'b11111011;
        // activate LED3 and Deactivate LED2, LED1, LED4
         LED_BCD = seconds / 10;
        // the third digit of the 16-bit number
            end
    3'b011: begin
        Anode_Activate = 8'b11110111;
        // activate LED4 and Deactivate LED2, LED3, LED1
         LED_BCD = seconds % 10;
        // the fourth digit of the 16-bit number
          end
    3'b100: begin
        Anode_Activate = 8'b11101111;
        // activate LED5 and Deactivate LED2, LED3, LED4
            LED_BCD = (team_2 / 10);
        // the first digit of the 16-bit number
          end
    3'b101: begin
        Anode_Activate = 8'b11011111;
        // activate LED6 and Deactivate LED1, LED3, LED4
```

```
            LED_BCD = (team_2 % 10);
          // the second digit of the 16-bit number
            end
        3'b110: begin
          Anode_Activate = 8'b10111111;
          // activate LED7 and Deactivate LED2, LED1, LED4
            LED_BCD = (team_1 / 10);
          // the third digit of the 16-bit number
            end
        3'b111: begin
          Anode_Activate = 8'b01111111;
          // activate LED8 and Deactivate LED2, LED3, LED1
          begin
            LED_BCD = (team_1 % 10);
          end
          // the fourth digit of the 16-bit number
            end
        endcase
    end

    // Cathode patterns of the 7-segment LED display
    always @(*)
    begin
        case(LED_BCD)
      0: LED_out = 7'b0000001; // "0"
      1: LED_out = 7'b1001111; // "1"
      2: LED_out = 7'b0010010; // "2"
      3: LED_out = 7'b0000110; // "3"
      4: LED_out = 7'b1001100; // "4"
      5: LED_out = 7'b0100100; // "5"
      6: LED_out = 7'b0100000; // "6"
      7: LED_out = 7'b0001111; // "7"
      8: LED_out = 7'b0000000; // "8"
      9: LED_out = 7'b0000100; // "9"
      default: LED_out = 7'b0000001; // "0"
        endcase
    end

always @(posedge slow_clock)
begin
    if(reset_points) begin //reset team scores here
        team_1 = 7'b0000000;
        team_2 = 7'b0000000;
```

```
        end

       if (one_point)
        begin
        if(team)
            team_1 = team_1 + 2'b01;
            else
            team_2 = team_2 + 2'b01;
        end
        else if (two_point) begin
            if(team)
            team_1 = team_1 + 2'b10;
            else
            team_2 = team_2 + 2'b10;
        end
        else if (three_point) begin
            if(team)
            team_1 <= team_1 + 2'b11;
            else
            team_2 <= team_2 + 2'b11;
        end

        if(team_1 >= 7'b1100100)
            team_1 = 7'b0000000;

        if(team_2 >= 7'b1100100)
            team_2 = 7'b0000000;
    end
endmodule
```

- **Clock_Divider.v File**

```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
// school: CSUF
// Engineers: Jeremy Escobar, Duy Nguyen, Spencer Williams
//
// Create Date: 11/16/2021 07:26:26 PM
// Design Name: Basket ball score board
// Module Name: timer, scoreboard
// Project Name: 446 final project
// Target Devices: FPGA Nexys A7
//
// Additional Comments: group project
//
//////////////////////////////////////////////////////////////////////////////////

module clock_divider(
input clk,
input reset,
output reg sclk
);
reg [31:0] count;
always@(posedge clk or negedge reset)
begin
if(reset == 1'b0) begin
   count <= 32'd0;
sclk <= 1'b0;
end else begin
if(count == 32'd9000000) begin
   count <= 32'd0;
sclk <= ~sclk;
end else begin
   count <= count + 1;
end
end
end
endmodule
```

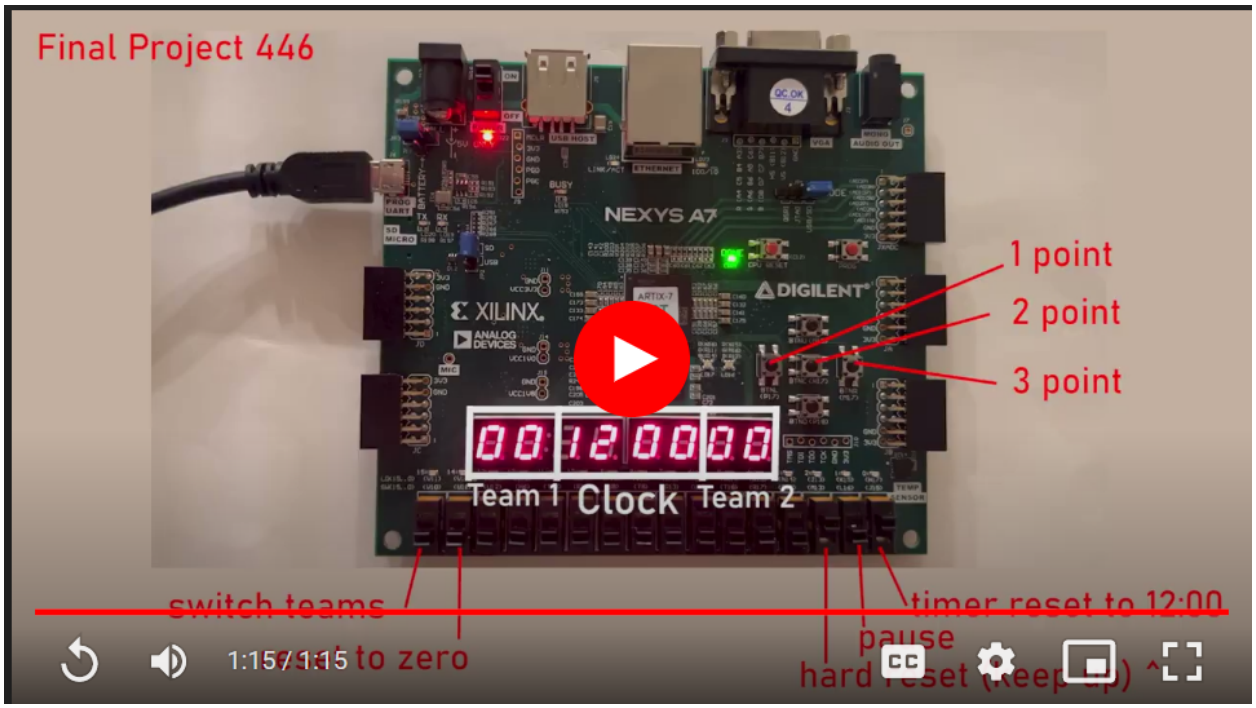
❖ Constraints.v File

```
## This file is a general .xdc for the Nexys A7-100T

## To use it in a project:

## - uncomment the lines corresponding to used pins

## - rename the used ports (in each line, after get_ports) according to the top level signal names in the project

#set_property BITSTREAM.GENERAL.COMPRESS TRUE [current_design]

#set_property BITSTREAM.CONFIG.SPI_BUSWIDTH 4 [current_design]

# Clock signal

set_property -dict {PACKAGE_PIN E3 IOSTANDARD LVCMOS33} [get_ports clock]

#create_clock -add -name sys_clk_pin -period 10.00 -waveform {0 5} [get_ports {clk_in}];

#set_property -dict {PACKAGE_PIN E3 IOSTANDARD LVCMOS33} [get_ports {clk}];


##Switches


set_property -dict {PACKAGE_PIN J15 IOSTANDARD LVCMOS33} [get_ports reset_score]

set_property -dict {PACKAGE_PIN L16 IOSTANDARD LVCMOS33} [get_ports pause]

set_property -dict {PACKAGE_PIN M13 IOSTANDARD LVCMOS33} [get_ports reset]

set_property -dict {PACKAGE_PIN U11 IOSTANDARD LVCMOS33} [get_ports reset_points]

set_property -dict {PACKAGE_PIN V10 IOSTANDARD LVCMOS33} [get_ports team]


##7 segment display


set_property -dict {PACKAGE_PIN T10 IOSTANDARD LVCMOS33} [get_ports {LED_out[6]}]

set_property -dict {PACKAGE_PIN R10 IOSTANDARD LVCMOS33} [get_ports {LED_out[5]}]

set_property -dict {PACKAGE_PIN K16 IOSTANDARD LVCMOS33} [get_ports {LED_out[4]}]

set_property -dict {PACKAGE_PIN K13 IOSTANDARD LVCMOS33} [get_ports {LED_out[3]}]

set_property -dict {PACKAGE_PIN P15 IOSTANDARD LVCMOS33} [get_ports {LED_out[2]}]
```

```
set_property -dict {PACKAGE_PIN T11 IOSTANDARD LVCMOS33} [get_ports {LED_out[1]}]

set_property -dict {PACKAGE_PIN L18 IOSTANDARD LVCMOS33} [get_ports {LED_out[0]}]


set_property -dict {PACKAGE_PIN J17 IOSTANDARD LVCMOS33} [get_ports {Anode_Activate[7]}]

set_property -dict {PACKAGE_PIN J18 IOSTANDARD LVCMOS33} [get_ports {Anode_Activate[6]}]

set_property -dict {PACKAGE_PIN T9 IOSTANDARD LVCMOS33} [get_ports {Anode_Activate[3]}]

set_property -dict {PACKAGE_PIN J14 IOSTANDARD LVCMOS33} [get_ports {Anode_Activate[2]}]

set_property -dict {PACKAGE_PIN P14 IOSTANDARD LVCMOS33} [get_ports {Anode_Activate[1]}]

set_property -dict {PACKAGE_PIN T14 IOSTANDARD LVCMOS33} [get_ports {Anode_Activate[0]}]

set_property -dict {PACKAGE_PIN K2 IOSTANDARD LVCMOS33} [get_ports {Anode_Activate[5]}]

set_property -dict {PACKAGE_PIN U13 IOSTANDARD LVCMOS33} [get_ports {Anode_Activate[4]}]


##Buttons

set_property -dict {PACKAGE_PIN N17 IOSTANDARD LVCMOS33} [get_ports two_point]

set_property -dict {PACKAGE_PIN P17 IOSTANDARD LVCMOS33} [get_ports one_point]

set_property -dict {PACKAGE_PIN M17 IOSTANDARD LVCMOS33} [get_ports three_point]



create_clock -period 15.000 -name my_clock -waveform {0.000 7.500} -add [get_ports clock]

set_input_delay -clock [get_clocks *] -add_delay 5.000 [get_ports {reset_points reset_score team
three_point two_point one_point pause}]

set_output_delay -clock [get_clocks *] -add_delay 5.000 [get_ports -filter { NAME =~  "*" && DIRECTION
== "OUT" }]
```

- **[30 pts] Working demonstration (test bench waveforms (if applicable), FPGA working video (submit as mp4 or a youtube link)**

https://drive.google.com/file/d/1bk-vpCnUkbz4tfjUjqhhPSHxbeLWmv30/view



◦ While making the video, make a title ppt slide with the project name and group member's name. Also, explain the working alongside while recording the demo.

- **[5 pts] Discussion:** Throughout the process of completing the lab we worked very closely together. We were able to make appropriate changes and implement changes to improve the design and overall code of our project. In the beginning we were using about 8.00 watts and after changes we only needed 0.068 watts.

- **[5 pts] Challenges you faced, and If you were to redo this project, what would you do differently? Why? (Answer this question even if you would make no changes to how you would complete this assignment) .**

Challenges we faced were getting all the LEDs to light initially. This was overcome by changing [19:18] LED activation to [19:17]. Once that was done task one was complete and we were finally able to see the LEDS all one. Next we ran into another issue with a debounce issue. When we pressed the button the points would increment 0-20, then 20-40 and so on. After that we added the 5 hz clock to the point counter. We were finally able to see the numbers increment correctly. If we had to do differently, maybe we add some LEDs to go along with the each score or to signify which team won

- **References:**

Just used all the powerpoints and .v files provided to us on canvas and each other.

Term project grade distribution (total 20% of grades)

Component	Proposal	Presentation and demonstration	Peer review	Written Report
Points	10 points	25 points	5 points	90 points

[25 pts] Project PowerPoint presentation and live demo.

[10 pts] Proposal (you already submitted it).