Computer Engineering Program

College of Engineering and Computer Science

Spring 2021

Project: Part A

EGCP 446-02 – 4/16/2021

Prepared by Scott Knowles, Rawabi Alayed, Jose Solano

Table of Contents

Introduction

In this project, we will implement portions from past labs into a single program utilizing what we've learned earlier in this class. The first section of the project utilizes a finite state machine that counts up from 0 to 9 that updates with a slow clock divider and outputs the state values to a seven-segment display. Once we have fully implemented the components, our board will show a value on each seven-segment display that counts from 0 to 9, with each switch allowing us to choose which displays are on or off.

Procedure

In this first part of the project, we were tasked with implementing a slow clock divider, a finite state machine, and a seven-segment decoder into one project, and connecting the three components with a top module. We went to work implementing the three individual components first, and once we had completed work on all three components, we then connected the components with the top module. Once we had completed the top module, we connected the package pins and generated the constraints and bitstream files. After we generated the bitstream file, we programmed the board and tested the program to ensure the board functioned properly.

Group Tasks

Scott Knowles – Implementation of 7-Segment Decoder and Clock Divider, Recorded Demo Video, Wrote Report

Rawabi Alayed – Implementation of Top Module, Assisted with Debugging and Testing, Contributed to Report

Jose Solano – Implementation of Counter, Generated Constraints File, Assisted with Debugging and Testing, Contributed to Report

Code Files and Constraints Files

Top.v

```
`timescale 1ns / 1ps

//////////////////////////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date: 04/06/2021 05:42:58 AM
// Design Name:
// Module Name: Top
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////
```

```
module Top(
    input Clk, Reset,
    input [7:0] AN,
    output [7:0] display,
    output [7:0] Anode
    );
   wire Clock1Hz;
   wire [3:0] Count;

    assign Anode = AN;
    /// Connect different building block together

    //1. Clock Divider 1Hz Frequency
    Clock_Divider BLOCK1 (.Clk(Clk), .Reset(Reset),
.Slow_Clock(Clock1Hz));

    //2. Counter 0-9
    //Counter is driven by  slow clock generated by clock divider
    Counter BLOCK2 (.Clk(Clock1Hz), .Reset(Reset),.Q(Count));

    //3. Binary to 7 segment
    Bin_7Segment BLOCK3 (.Bin(Count), .Seven_Segment(display));

endmodule
```

Clock_Divider.v

```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date: 04/06/2021 05:32:41 AM
// Design Name:
// Module Name: Clock_Divider
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////

module Clock_Divider(
input Clk, Reset,
output Slow_Clock
);
reg [25:0] Counter = 26'b0; //26 bit counter = 1 Hz clock speed

//// write a Verilog code to slow down the clock from 100 MHz to 1 Hz

//Clock Divider
always @(posedge Clk, posedge Reset)
if(Reset)
```

```
Counter <= 26'b0;
else
Counter <= Counter + 1'b1;
assign Slow_Clock = Counter[25];
endmodule
```

Counter.v

```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date: 04/06/2021 05:36:58 AM
// Design Name:
// Module Name: Counter
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////
```

```
module Counter(
    input Clk, Reset,
    output [3:0] Q
    );

// Make sure your Counter is driven by  slow clock generated by clock divider
parameter ST_ZERO = 4'b0000, ST_ONE = 4'b0001, ST_TWO = 4'b0010, ST_THREE = 4'b0011, ST_FOUR = 4'b0100, ST_FIVE = 4'b0101, ST_SIX = 4'b0110, ST_SEVEN = 4'b0111, ST_EIGHT = 4'b1000, ST_NINE = 4'b1001;

reg [3:0] state, n_state; //state = present state, n_state = next state

always @(posedge Clk, posedge Reset)
     if (Reset)
          state <= ST_ZERO;
          else
          state <= n_state;

always @(*)
     case (state)
          ST_ZERO:
               n_state <= ST_ONE;
          ST_ONE:
```

```
                n_state <= ST_TWO;
            ST_TWO:
                n_state <= ST_THREE;
            ST_THREE:
                n_state <= ST_FOUR;
            ST_FOUR:
                n_state <= ST_FIVE;
            ST_FIVE:
                n_state <= ST_SIX;
            ST_SIX:
                n_state <= ST_SEVEN;
            ST_SEVEN:
                n_state <= ST_EIGHT;
            ST_EIGHT:
                n_state <= ST_NINE;
            ST_NINE:
                n_state <= ST_ZERO;
            default:
                n_state <= ST_ZERO;
        endcase

assign Q = state;

endmodule
```

Bin_7Segment.v

```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date: 04/06/2021 05:33:47 AM
// Design Name:
// Module Name: Bin_7Segment
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////

module Bin_7Segment(
input [3:0] Bin,
output reg [7:0] Seven_Segment
);

always@(*)
begin
case(Bin)
4'b0000: Seven_Segment = 8'hC0; //0
4'b0001: Seven_Segment = 8'hF9; //1
4'b0010: Seven_Segment = 8'hA4; //2
```

```
4'b0011: Seven_Segment = 8'hB0; //3
4'b0100: Seven_Segment = 8'h99; //4
4'b0101: Seven_Segment = 8'h92; //5
4'b0110: Seven_Segment = 8'h82; //6
4'b0111: Seven_Segment = 8'hF8; //7
4'b1000: Seven_Segment = 8'h80; //8
4'b1001: Seven_Segment = 8'h90; //9
endcase
end
endmodule
```

projpartA.xdc

```
set_property IOSTANDARD LVCMOS18 [get_ports {AN[7]}]
set_property IOSTANDARD LVCMOS18 [get_ports {AN[6]}]
set_property IOSTANDARD LVCMOS18 [get_ports {AN[5]}]
set_property IOSTANDARD LVCMOS18 [get_ports {AN[4]}]
set_property IOSTANDARD LVCMOS18 [get_ports {AN[3]}]
set_property IOSTANDARD LVCMOS18 [get_ports {AN[2]}]
set_property IOSTANDARD LVCMOS18 [get_ports {AN[1]}]
set_property IOSTANDARD LVCMOS18 [get_ports {AN[0]}]
set_property IOSTANDARD LVCMOS18 [get_ports {Anode[7]}]
set_property IOSTANDARD LVCMOS18 [get_ports {Anode[6]}]
set_property IOSTANDARD LVCMOS18 [get_ports {Anode[5]}]
set_property IOSTANDARD LVCMOS18 [get_ports {Anode[4]}]
set_property IOSTANDARD LVCMOS18 [get_ports {Anode[3]}]
set_property IOSTANDARD LVCMOS18 [get_ports {Anode[2]}]
set_property IOSTANDARD LVCMOS18 [get_ports {Anode[1]}]
set_property IOSTANDARD LVCMOS18 [get_ports {Anode[0]}]
set_property IOSTANDARD LVCMOS18 [get_ports {display[7]}]
set_property IOSTANDARD LVCMOS18 [get_ports {display[6]}]
set_property IOSTANDARD LVCMOS18 [get_ports {display[5]}]
set_property IOSTANDARD LVCMOS18 [get_ports {display[4]}]
set_property IOSTANDARD LVCMOS18 [get_ports {display[3]}]
```

```
set_property IOSTANDARD LVCMOS18 [get_ports {display[2]}]

set_property IOSTANDARD LVCMOS18 [get_ports {display[1]}]

set_property IOSTANDARD LVCMOS18 [get_ports {display[0]}]

set_property IOSTANDARD LVCMOS18 [get_ports Clk]

set_property IOSTANDARD LVCMOS18 [get_ports Reset]

set_property PACKAGE_PIN E3 [get_ports Clk]

set_property PACKAGE_PIN J15 [get_ports Reset]

set_property PACKAGE_PIN L16 [get_ports {AN[0]}]

set_property PACKAGE_PIN T8 [get_ports {AN[7]}]

set_property PACKAGE_PIN R13 [get_ports {AN[6]}]

set_property PACKAGE_PIN U18 [get_ports {AN[5]}]

set_property PACKAGE_PIN T18 [get_ports {AN[4]}]

set_property PACKAGE_PIN R17 [get_ports {AN[3]}]

set_property PACKAGE_PIN R15 [get_ports {AN[2]}]

set_property PACKAGE_PIN M13 [get_ports {AN[1]}]

set_property PACKAGE_PIN U13 [get_ports {Anode[7]}]

set_property PACKAGE_PIN K2 [get_ports {Anode[6]}]

set_property PACKAGE_PIN T14 [get_ports {Anode[5]}]

set_property PACKAGE_PIN P14 [get_ports {Anode[4]}]

set_property PACKAGE_PIN J14 [get_ports {Anode[3]}]

set_property PACKAGE_PIN T9 [get_ports {Anode[2]}]

set_property PACKAGE_PIN J18 [get_ports {Anode[1]}]
```

```
set_property PACKAGE_PIN J17 [get_ports {Anode[0]}]

set_property PACKAGE_PIN H15 [get_ports {display[7]}]

set_property PACKAGE_PIN L18 [get_ports {display[6]}]

set_property PACKAGE_PIN T11 [get_ports {display[5]}]

set_property PACKAGE_PIN P15 [get_ports {display[4]}]

set_property PACKAGE_PIN K13 [get_ports {display[3]}]

set_property PACKAGE_PIN K16 [get_ports {display[2]}]

set_property PACKAGE_PIN R10 [get_ports {display[1]}]

set_property PACKAGE_PIN T10 [get_ports {display[0]}]
```