EECS 3215 Digital Logic Design

Lab 1

Michael Founk

213641204

Sean O’Brien

213735741

Problem Statement

The objective of this lab was to use the DE2 board to design a medium complexity system. This system would run a stopwatch on the 7 segment LED display that is present on the board. The stopwatch should count up by seconds when the start switch is in the on position. The stopwatch has to stop at the current number of seconds when the stop switch is in the on position. And finally the stopwatch resets to 0 seconds (0 at all the spots on the 7 segment display) when the reset switch is in the on position.

Top-Level Design

The design is as follows: there will be 2 different states of the stopwatch; the first is when the stopwatch is running and the second is when the stopwatch is stopped.

We create a trigger for the reset key to always flip the numbers back to zeroes when the reset key is switched to on. The reset switch only resets the counter to zero when it moves from off to on, moving from on to off has no effect other than enabling the reset action. After the reset switch is moved from off to on, the stopwatch remains in the same state it was in before the reset switch was flipped so the clock will continue counting or continue waiting.

In the stop state the 7 segment display is kept at the same numbers and we wait for the switch start to be turned on which will bring it into the second state, other than that it just stays in state one. In state 2 we step through the seconds using a clock count. When the seconds are incremented we have to move up the time on the stopwatch by one. We track the digits separately in each of the 7 segment displays and when one of them reaches 10 we have to set it back to 0 and increment the higher digit. This is done twice in our code since we go up to only 99 seconds, but can be done as many times as needed. The way this is done is by taking advantage of the binary representation of 10, we simply add 6 to it and this is equal to 16, which has a binary representation of 1 in the 5th bit and 0 for bits 1-4, which for our purpose works out well because we want to add one to the higher order digit and represent 0 using 4 bits.

Last thing that needs to be done is to actually set the 7 segment displays for the digits that are supposed to be shown. There are a total of 10 different digits that can be shown and we can create a case for each of them. This case is then triggered any time that the digits change in the code above. Then the appropriate segments light up and display that digit.

Code

```
module lab1(Clk, HexA, HexB, start, stop, reset);

input Clk;

input start, stop, reset;

reg [25:0] clockCount;

reg [7:0] secondCount;
reg Cout;


reg [2:0] State;

parameter S0 = 3'b000;

parameter S1 = 3'b001;

parameter S2 = 3'b010;

parameter S3 = 3'b011;


always@(posedge Clk)

begin

        if(reset)

        begin

                State <= S0;

                secondCount <= 0;

        end

        else

        begin

                case(State)

                S0:begin

                        secondCount <= secondCount;

                        if(start)

                        begin

                                State <= S1;

                        end

                        else

                        begin

                                State <= S0;

                        end

                end
```

```
S1:begin
    if(reset)
    begin
        secondCount <= 0;
        State <= S0;
    end
    else if(stop)
    begin
        State <= S0;
    end
    else
    begin
        if(secondCount[3:0] > 9)
        begin
            {Cout,secondCount[3:0]} <= secondCount[3:0] + 6;
            secondCount[7:4] <= secondCount[7:4] + 1;
        end
        if(secondCount[7:4] > 9)
        begin
            secondCount[7:4] <= secondCount[7:4] + 6;
        end
        if(clockCount == 26'd50000000)
        begin
            secondCount <= secondCount + 1;
            clockCount <= 26'b0;
            State <= S1;
        end
        else
        begin
```

```
                    clockCount <= clockCount + 26'b1;
                    State <= S1;
                end
            end
        end
            default State <= S0;
        endcase
    end
end


output reg [6:0] HexA, HexB;
always@(secondCount[3:0])
begin
case(secondCount[3:0])
    4'b0000: HexA = ~(7'b0111111);
    4'b0001: HexA = ~(7'b0000110);
    4'b0010: HexA = ~(7'b1011011);
    4'b0011: HexA = ~(7'b1001111);
    4'b0100: HexA = ~(7'b1100110);
    4'b0101: HexA = ~(7'b1101101);
    4'b0110: HexA = ~(7'b1111101);
    4'b0111: HexA = ~(7'b0000111);
    4'b1000: HexA = ~(7'b1111111);
    4'b1001: HexA = ~(7'b1101111);
endcase
case(secondCount[7:4])
    4'b0000: HexB = ~(7'b0111111);
    4'b0001: HexB = ~(7'b0000110);
```

```
        4'b0010: HexB = ~(7'b1011011);
        4'b0011: HexB = ~(7'b1001111);
        4'b0100: HexB = ~(7'b1100110);
        4'b0101: HexB = ~(7'b1101101);
        4'b0110: HexB = ~(7'b1111101);
        4'b0111: HexB = ~(7'b0000111);
        4'b1000: HexB = ~(7'b1111111);
        4'b1001: HexB = ~(7'b1101111);
endcase
end

endmodule
```

Pin assignment

Here is the full pin assignment needed for this lab:

```
set_global_assignment -name FAMILY "Cyclone II"
set_global_assignment -name DEVICE EP2C35F672C6
set_global_assignment -name TOP_LEVEL_ENTITY lab1
set_global_assignment -name ORIGINAL_QUARTUS_VERSION "12.1 SP1"
set_global_assignment -name PROJECT_CREATION_TIME_DATE "16:14:28  JANUARY 07, 2019"
set_global_assignment -name LAST_QUARTUS_VERSION "12.1 SP1"
set_global_assignment -name PROJECT_OUTPUT_DIRECTORY output_files
set_global_assignment -name MIN_CORE_JUNCTION_TEMP 0
set_global_assignment -name MAX_CORE_JUNCTION_TEMP 85
set_global_assignment -name ERROR_CHECK_FREQUENCY_DIVISOR 1
set_global_assignment -name VERILOG_FILE stopwatch.v
set_global_assignment -name PARTITION_NETLIST_TYPE SOURCE -section_id Top
set_global_assignment -name PARTITION_FITTER_PRESERVATION_LEVEL PLACEMENT_AND_ROUTING -section_id Top
set_global_assignment -name PARTITION_COLOR 16764057 -section_id Top
set_global_assignment -name CDF_FILE output_files/Chain1.cdf
set_location_assignment PIN_N2 -to Clk
set_location_assignment PIN_N25 -to reset
set_location_assignment PIN_N26 -to start
set_location_assignment PIN_P25 -to stop
set_location_assignment PIN_AF10 -to HexA[0]
set_location_assignment PIN_AB12 -to HexA[1]
set_location_assignment PIN_AC12 -to HexA[2]
set_location_assignment PIN_AD11 -to HexA[3]
set_location_assignment PIN_AE11 -to HexA[4]
set_location_assignment PIN_V14 -to HexA[5]
set_location_assignment PIN_V13 -to HexA[6]
set_location_assignment PIN_V20 -to HexB[0]
set_location_assignment PIN_V21 -to HexB[1]
set_location_assignment PIN_W21 -to HexB[2]
set_location_assignment PIN_Y22 -to HexB[3]
set_location_assignment PIN_AA24 -to HexB[4]
set_location_assignment PIN_AA23 -to HexB[5]
set_location_assignment PIN_AB24 -to HexB[6]
set_instance_assignment -name PARTITION_HIERARCHY root_partition -to | -section_id Top
```