Group1 Cmod A7 stopwatch report

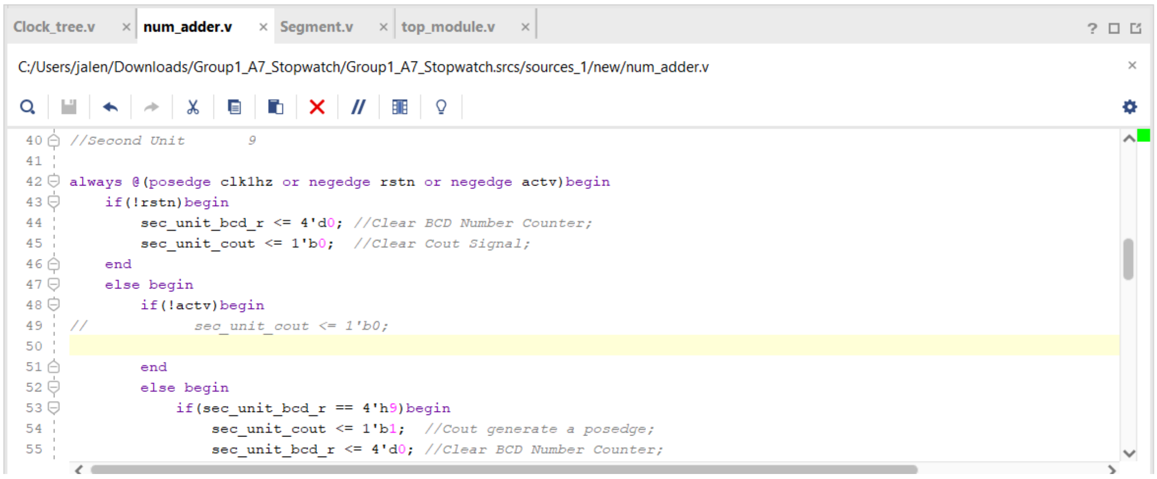
Jarell, Lucas, Jalen, Stephanie, Syarifuddin

Overall, the design of versatile and modular code like this was very tricky yet fun. It was a good brain teaser for all of us, to try and figure out how and where in the code the different segments come into play together.

What we did well, is that we managed to understand the hardware of this project well. We had a thorough understanding of how the 7 segment display functions, what anode is connected to what wire, how the DP and digits work, and how the 8 bit number affects the digits being shown on the display.

Some interesting things we tried to implement were the pause and play functions. We also tried to modify the LED blinking, making it blink in different patterns, or making it dependent on the 7 segment display. And last but not least, we tried to make the DP blink at a constant frequency (2Hz). We were mostly successful. The code for the DP and the Pause play is shown below. We found the DP blinking to be the most challenging yet fruitful challenge, as it involved us making new always blocks, and introducing new variables into the SEGMENT module, to allow it to be shown. The tricky thing here was that digit 2 and dp are activated by the same anode, so we had to have them take turns being flashed, which is why we added a new COUNTT variable.

Doing the DP blink function was challenging, because to do that, we really had to understand the specifics of how the 7 segment and the code was functioning. We tried multiple times, almost giving up, but in the end we managed to solve it, and make it functioning.



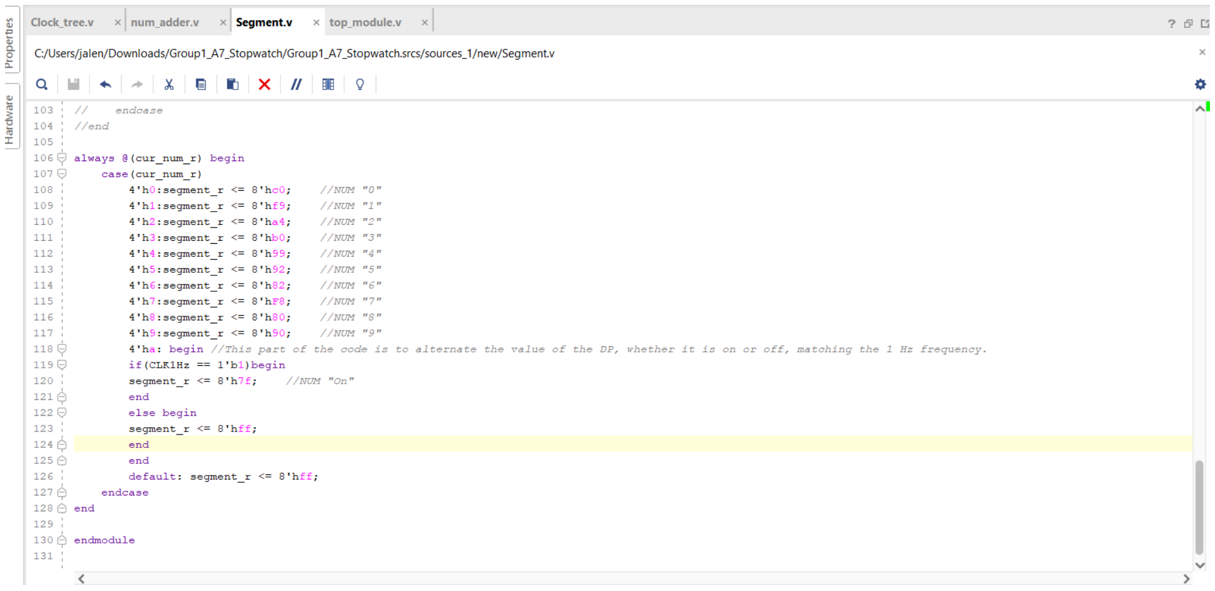
C:/Users/jalen/Downloads/Group1_A7_Stopwatch/Group1_A7_Stopwatch.srcs/sources_1/new/num_adder.v

```
//Second Unit       9

always @(posedge clk1hz or negedge rstn or negedge actv)begin
    if(!rstn)begin
        sec_unit_bcd_r <= 4'd0; //Clear BCD Number Counter;
        sec_unit_cout <= 1'b0;  //Clear Cout Signal;
    end
    else begin
        if(!actv)begin
//              sec_unit_cout <= 1'b0;

        end
        else begin
            if(sec_unit_bcd_r == 4'h9)begin
                sec_unit_cout <= 1'b1;  //Cout generate a posedge;
                sec_unit_bcd_r <= 4'd0; //Clear BCD Number Counter;
```

Figure 1: pause and play implementation



C:/Users/jalen/Downloads/Group1_A7_Stopwatch/Group1_A7_Stopwatch.srcs/sources_1/new/Segment.v

```verilog
//    endcase
//end

always @(cur_num_r) begin
    case(cur_num_r)
        4'h0:segment_r <= 8'hc0;    //NUM "0"
        4'h1:segment_r <= 8'hf9;    //NUM "1"
        4'h2:segment_r <= 8'ha4;    //NUM "2"
        4'h3:segment_r <= 8'hb0;    //NUM "3"
        4'h4:segment_r <= 8'h99;    //NUM "4"
        4'h5:segment_r <= 8'h92;    //NUM "5"
        4'h6:segment_r <= 8'h82;    //NUM "6"
        4'h7:segment_r <= 8'hF8;    //NUM "7"
        4'h8:segment_r <= 8'h80;    //NUM "8"
        4'h9:segment_r <= 8'h90;    //NUM "9"
        4'ha: begin //This part of the code is to alternate the value of the DP, whether it is on or off, matching the 1 Hz frequency.
        if(CLK1Hz == 1'b1)begin
        segment_r <= 8'h7f;    //NUM "On"
        end
        else begin
        segment_r <= 8'hff;
        end
        end
        default: segment_r <= 8'hff;
    endcase
end

endmodule
```

Figure 2: if else statement to set DP to high or low.

```verilog
// an always block to start count 0, 1, 0, 1, 0, 1.....
// so that when count is 0, the digit at anode 2 is displayed. at count 1, the dp is displayed.
// Hence, as the loop goes through the anodes, digit 2 and the dp take turns being shown at the high frequency
always @(an_r)begin
    if(an_r == 4'b0100)begin
        if(COUNTT == 1'b1) begin
            COUNTT <= 1'b0;
        end
        else begin
            COUNTT <= COUNTT + 1'b1;
        end
    end
end
```

Figure 3: COUNTT implementation, to alternate between digit 2 and DP