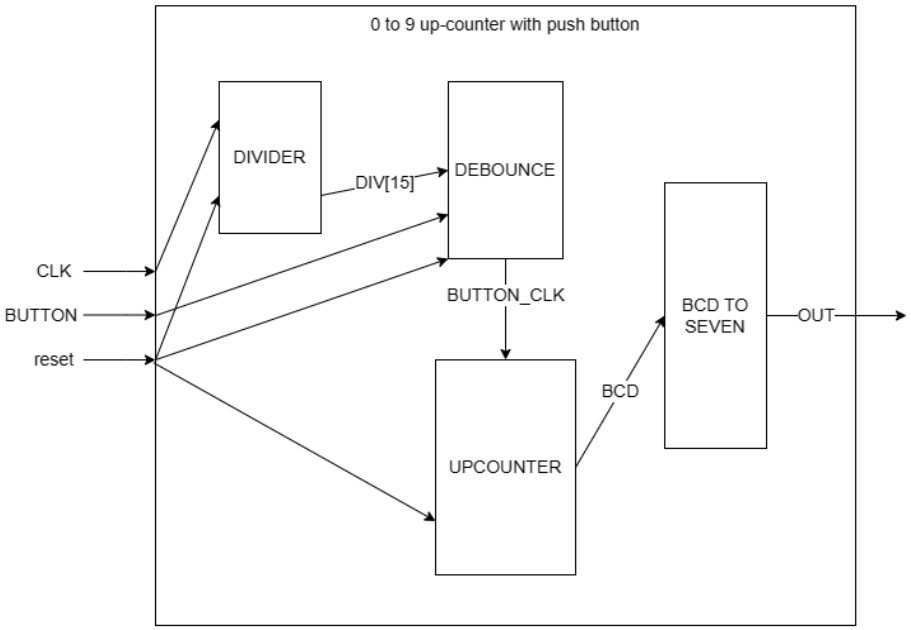
Lab 5 report

Name: Oscar Stark Student ID: F11015127

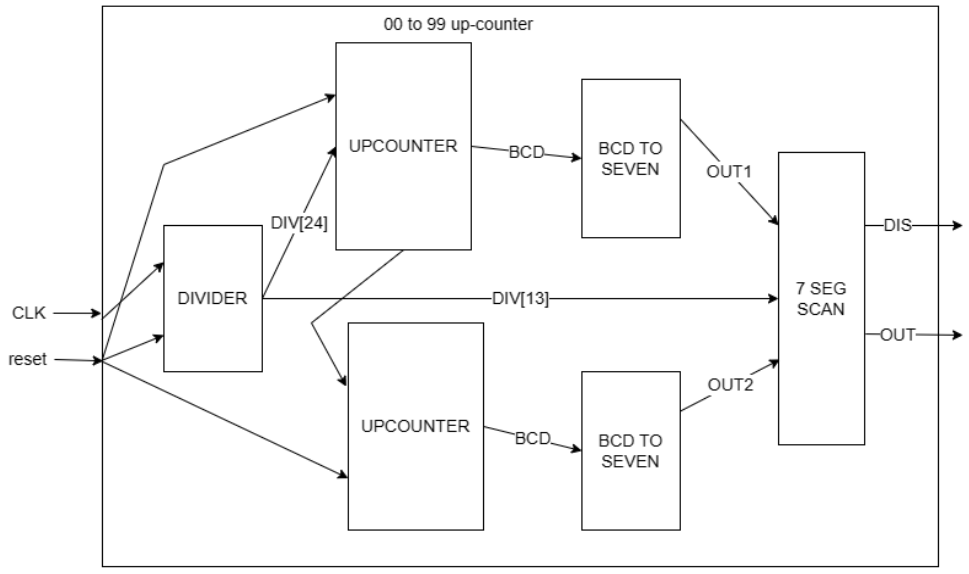
In this lab I implemented a 0 to 9 up-counter with a push button, and a 00 to 99 up-counter.

Circuit block diagrams

Task 1



Task 2



Verilog Code

Task 1 main code

```
module task1(CLK, BUTTON, OUT, DIS, reset);
    input CLK, BUTTON, reset;
    output wire [6:0] OUT; //the output for the 7SD
    output DIS;

    wire [31:0] DIV;
    wire BUTTON_CLK;
    assign DIS = 1; //used to turn on the desired 7SD

    divider DIVIDER(CLK, DIV, reset);
    debounce DEB(DIV[15], BUTTON, BUTTON_CLK, reset);
    upcounter UPC(BUTTON_CLK, BCD, reset);
    seven SEV(BCD, OUT);
endmodule
```

Task 2 main code

```
module task1(CLK, OUT, DIS, reset);
    input CLK, reset;
    output reg [6:0] OUT; //output for the 7SD(7 segment display)
    output reg [1:0] DIS; //output for choosing which 7SD lights up

    //wires
    wire [31:0] DIV;
    wire BUTTON_CLK;
    wire [3:0] BCD1, BCD2;
    wire [6:0] OUT1, OUT2;
    reg CLK_UPC2;

    divider DIVIDER(CLK, DIV, reset);
    upcounter UPC1(DIV[24], BCD1, reset);//upcounter for first digit
    always@(posedge DIV[24])//when the first digit reaches 9,
        begin                //the second digit is raised
            if(BCD1==9)
                CLK_UPC2 <= 1;
            else
                CLK_UPC2 <= 0;
         end
    upcounter UPC2(CLK_UPC2, BCD2, reset);//upcounter for second digit
    seven SEV1(BCD1, OUT1);//decoder for first digit
    seven SEV2(BCD2, OUT2);//decoder for second digit
```

```
    always@(DIV[13])//7 seg scan
        begin
            if(DIV[13])
            begin
                OUT <= OUT1;//display digit 1
                DIS <= 2'b01;
            end
            else
            begin
                OUT <= OUT2;//display digit 2
                DIS <= 2'b10;
            end
        end
endmodule
```

Divider

```
module divider(CLK, DIV, reset);
    input CLK, reset;
    output reg [31:0] DIV;

    always @(posedge CLK, posedge reset)
    begin
        if(reset)
            DIV <= 0; //reset button sets DIV to 0
        else
            DIV <= DIV+1; //posedge CLK adds 1 to the DIV
    end
endmodule
```

Debouncer

```
module debounce(CLK, BUTTON, BUTTON_CLK, reset);
    input CLK, BUTTON, reset;
    output reg BUTTON_CLK;

    wire[2:0]W;

    //3 D Flip Flops
    DFF DFF1(CLK, BUTTON, W[0], reset);
    DFF DFF2(CLK, W[0], W[1], reset);
    DFF DFF3(CLK, W[1], W[2], reset);

    always@(W)
        BUTTON_CLK <= W[0] & W[1] & W[2];

endmodule
```

D Flip Flop

```
module DFF(CLK, D, Q, reset);
    input CLK, D, reset;
    output reg Q;

    always @(posedge CLK, posedge reset)
    begin
        if(reset)   Q <= 1'b0; //reset button sets Q to 0
        else        Q <= D; //CLK sets Q = D
    end
endmodule
```

Up Counter

```
module upcounter(BUTTON_CLK, BCD, reset);
    input BUTTON_CLK, reset;
    output reg [3:0] BCD; //the number ranges from 0 to 15

    always @(posedge BUTTON_CLK, posedge reset)
        begin
            if(reset) //resets to 0
                BCD <= 0;
            else
                BCD <= BCD + 1; //CLK adds 1
        end
endmodule
```

Seven Segment Decoder

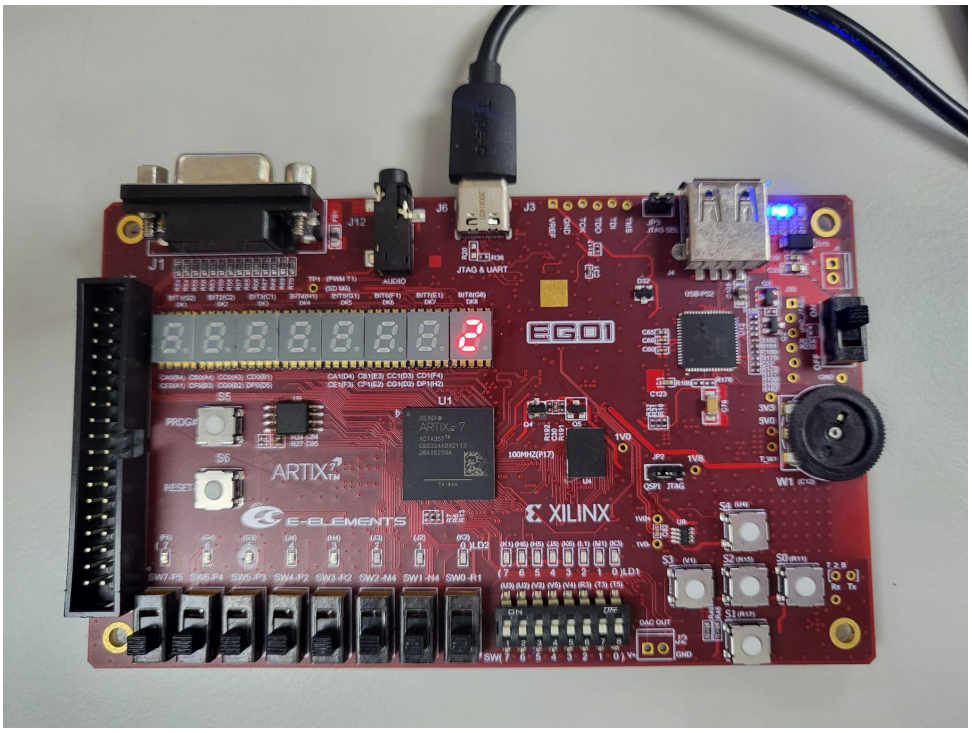
```
module seven(BCD, OUT);
    input [3:0] BCD; //the input in BCD
    output reg [6:0] OUT; //the out put for the 7SD

    always@(BCD) // this controls how the 7SD lights up
        case({BCD})
            4'b0000: {OUT} = 7'b1111110; // 0
            4'b0001: {OUT} = 7'b0110000; // 1
            4'b0010: {OUT} = 7'b1101101; // 2
            4'b0011: {OUT} = 7'b1111001; // 3
            4'b0100: {OUT} = 7'b0110011; // 4
            4'b0101: {OUT} = 7'b1011011; // 5
            4'b0110: {OUT} = 7'b1011111; // 6
            4'b0111: {OUT} = 7'b1110000; // 7
            4'b1000: {OUT} = 7'b1111111; // 8
            4'b1001: {OUT} = 7'b1111011; // 9
        endcase
endmodule
```

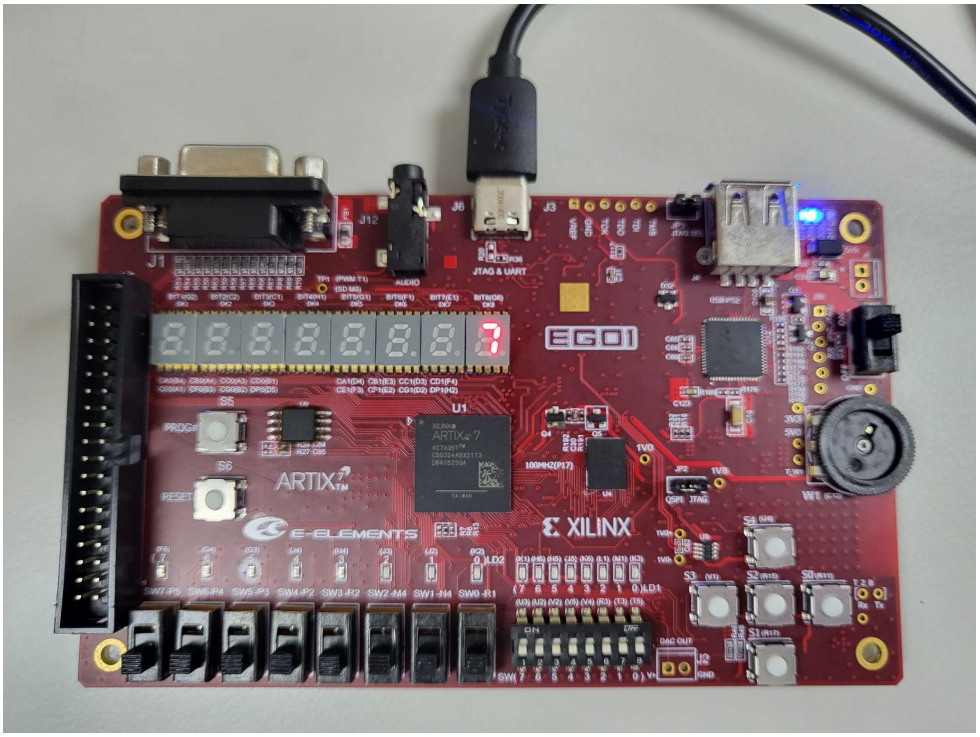
FPGA Results

Task 1

The counter goes from 0 to 9. The number is displayed on the 7 segment display. One button is used to reset the count to 0 and another button is used to add 1 to the count. Here the displayed count is 2:

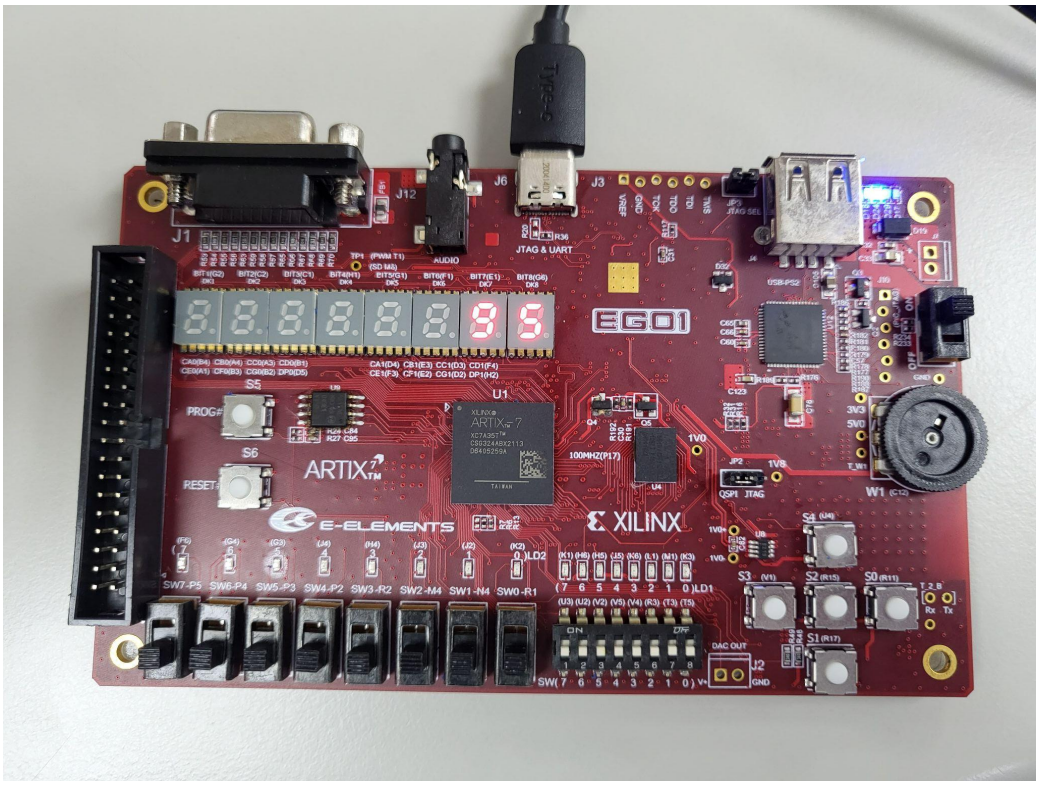


Here the displayed count is 7:



Task 2

The counter goes from 0 to 99 automatically (without pressing a button). The reset button sets the count to 0.
Here the displayed number is 95:



Here the displayed number is 45: