Name: Junzhe Wu Net id: Junzhew3

1. How many total digital input/output pins does the XEM 7310-A75 board have?
   10
2. What is the maximum clocking speed of the XEM 7310-A75 board?
   200 MHz
3. How does the XEM 6002 board compare to XEM 7310-A75 in terms of logic gate count, transfer speeds between the board and PC, external memory, and clocking speed of digital logic?

	XEM 6002 board	XEM 7310-A75
logic gate count	medium gate-count	high gate-count
transfer speeds	USB 2.0 36+ MiB/s	USB 3.0 340+ MiB/s
external memory	None	1 GiB
clocking speed of digital logic	1 MHz to 150 MHz	200 MHz

4. Why is the clkdiv register 24-bit long?
   Because clkdiv needs to count 10000000 and 2^{23}-1 = 8388607 < 10000000.
5. If clkdiv is declared as an 8-bit register, what is the minimum frequency you can achieve for the slow_clk signal using these FSMs?
   2^8-1 = 255. So the minimum frequency = 20MHz/255 ≈ 78431.37Hz
6. Look at the Project Summary window. In the Utilization section, you will find out the number of look up tables (LUT), flip flops (FF), input/output pins (IO) and buffers (BUFG) are used for your design. How many resources on the FPGA are used to implement your code?

Resource	Utilization	Available	Utilization%
LUT	30	47200	0.06
FF	33	94400	0.03
IO	14	285	4.91
BUFG	1	32	3.13

7. Include a printout of your Verilog code with your report.

```
8. `timescale 1ns / 1ps
9.
10.module intro(
11.    input [3:0] button,
12.    output [7:0] led,
13.    input sys_clkn,
14.    input sys_clkp
15.    );
16.
17.    reg [23:0] clkdiv;
18.    reg [7:0] counter;
19.    reg slow_clk;
20.
21.    // This section defines the main system clock from two
22.    //differential clock signals: sys_clkn and sys_clkp
23.    // Clk is a high speed clock signal running at ~200MHz
```

```
24.    wire clk;
25.    IBUFGDS osc_clk(
26.        .O(clk),
27.        .I(sys_clkp),
28.        .IB(sys_clkn)
29.    );
30.
31.    initial begin
32.        clkdiv = 0;
33.        counter = 8'h00;
34.    end
35.
36.    assign led = ~counter;
37.
38.    // This code creates a slow clock from the high speed Clk signal
39.    // You will use the slow clock to run your finite state machine
40.    // The slow clock is derived from the fast 20 MHz clock by
   dividing it 10,000,000 time
41.    // Hence, the slow clock will run at 2 Hz
42.    always @(posedge clk) begin
43.        clkdiv <= clkdiv + 1'b1;
44.        if (clkdiv == 10000000) begin
45.            slow_clk <= ~slow_clk;
46.            clkdiv <= 0;
47.        end
48.    end
49.
50.    //The main code will run fr0m the slow clock.  The rest of the
   code will be in this section.
51.    //The counter will increment when button 0 is pressed and on the
   rising edge of the slow clk
52.    //The counter will decrement when button 0 is pressed and on the
   rising edge of the slow clk
53.    always @(posedge slow_clk) begin
54.        if (button [2] == 1'b0 && button [3] == 1'b1 && button [0] ==
   1'b1 && button [1] == 1'b1) begin
55.            counter <= counter + 2'b10;
56.        end
57.        else if (button [3] == 1'b0  && button [2] == 1'b1 && button
   [0] == 1'b1 && button [1] == 1'b1) begin
58.            counter <= counter - 2'b10;
59.        end
60.        else if (button [0] == 1'b0  && button [3] == 1'b1 && button
   [2] == 1'b1 && button [1] == 1'b1) begin
```

```
61.             counter <= 8'hff;
62.         end
63.         else if (button [1] == 1'b0  && button [3] == 1'b1 && button
    [0] == 1'b1 && button [2] == 1'b1) begin
64.             counter <= 8'h00;
65.         end
66.     end
67.endmodule
```