Lab no 03: Implement BCD Counter

The purpose of this Lab is to learn to:

1) Implement a 4-bits BCD counter on FPGA. You will write the behavioral description for the BCD counter using Verilog.
2) Connect sub-modules on a top-level module. You will connect the BCD counter and the seven-segment decoder. **Refer to Lab 2 to review the seven-segment decoder.**
3) Use the push button on the FPGA board as an input clock to the BCD Counter and understand the debouncing problem.

Parts: -

1. Code the behavioral description of the 4-bit BCD Counter.
2. Connect the BCD counter and the seven-segment decoder on the top-level module and run it on FPGA.

Part 1. Code the BCD Counter

A BCD counter is a sequential arithmetic circuit with clock and reset inputs and 4-bit outputs. Reset initializes the output to 0. It counts from 0000 (0) to 1001 (9) in decimal form on the application of the clock signal.

Behavioral description Verilog code for BCD Counter

```
module bcd_counter(  clk,   reset,  count);
input clk , reset ;
//  4 bits output
output [3:0] count;
//  4 bits reg to hold the value of the output
reg [3:0] count;
//  BCD counter
always @ (posedge clk or posedge reset)
      begin
            if(reset) //  reset the counter circuit to initial (zero)
                  count <= 0;
            else
                  begin
                        //  check the count value equal nine to reset
                        if (count == 9)
                              count <= 4'b0;
                        //  if less than nine, add one
                        else
                              count <= count + 1'b1;
                  end
      end
endmodule
```

Verilog code for Decoder to 7 segments

```
module decoder_7seg (A, B, C, D, led_a, led_b, led_c,
led_d, led_e, led_f, led_g);
      input A, B, C, D;
      output led_a, led_b, led_c, led_d, led_e, led_f,
             led_g;

      assign led_a = ~(A | C | B&D | ~B&~D);
      assign led_b = ~(~B | ~C&~D | C&D);
      assign led_c = ~(B | ~C | D);
      assign led_d = ~(~B&~D | C&~D | B&~C&D | ~B&C |A);
      assign led_e = ~(~B&~D | C&~D);
      assign led_f = ~(A | ~C&~D | B&~C | B&~D);
      assign led_g = ~(A | B&~C | ~B&C | C&~D);
endmodule
```

Part 2. Connect the BCD counter and the seven-segment decoder

```
module counter_to_decoder ( clk, reset, segment_leds);

input clk , reset ;
output [6:0] segment_leds;
wire [3:0] count_wire;

// Instantiation of the BCD counter
bcd_counter bcd_counter_dut ( clk, reset, count_wire);

// Instantiation of the seven segments decoder
decoder_7seg decoder_7seg_dut ( count_wire[3],
count_wire[2], count_wire[1], count_wire[0],
segment_leds[6], segment_leds[5], segment_leds[4],
segment_leds[3], segment_leds[2], segment_leds[1],
segment_leds[0] );

endmodule
```

Run the integrated design (Counter + Decoder) on FPGA.

Refer to Lab 2 to program the FPGA by **Quartus**

Use DE-10lite kit, Altera MAX 10 based FPGA board
Check DE10-lite user manual (Here) for pin assignment.

Note: Assign Clock to the push button (KEY0) and

Assign Reset to the switch button (SW0) on FPGA.