HW#3

UCEC Extension – Digital Design using FPGA
Jae-Yang Park (jaeyangp@gmail.com)

Problem 3.7.4

Implement and test the circuit shown in the 3.14

> **Testing Switches and LEDs**
> When SW7 is ON, LED7 must be ON. All other LEDs are OFF.
> When SW7 is OFF and SW6 is ON, LED6 must be ON. All other LEDs are OFF.
> When SW7, SW6 is OFF and SW5 is ON, LED5 must be ON. All other LEDs are OFF.
> When SW7…SW1 is OFF and SW0 is ON, LED0 must be ON. All other LEDs are OFF.

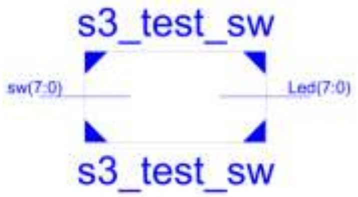
1. What is the name of the circuit that you are designing?
   Priority Selector
   SW7 has the highest priority and SW0 has the lowest priority.
   When higher priority SW is ON, lower priority SW is ignored.
2. See below design
3. 8 LUTs are required, but MUX and BUFFER are used in the design.
4. Design – below, 10 LUTs are required in the structural LUT design.

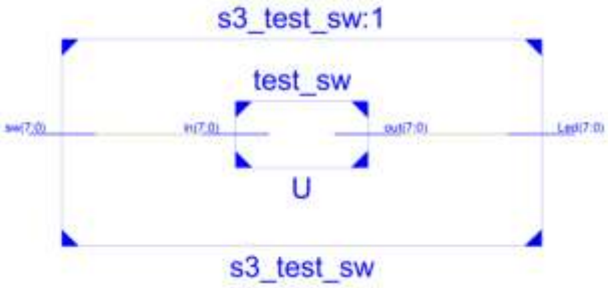
SW (1 = ON, 0 = OFF, X = Don't care)								LED (1 = ON, 0 = OFF)								
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**	
1	X	X	X	X	X	X	X	1	0	0	0	0	0	0	0	
0	1	X	X	X	X	X	X	0	1	0	0	0	0	0	0	
0	0	1	X	X	X	X	X	0	0	1	0	0	0	0	0	
0	0	0	1	X	X	X	X	0	0	0	1	0	0	0	0	
0	0	0	0	1	X	X	X	0	0	0	0	1	0	0	0	
0	0	0	0	0	1	X	X	0	0	0	0	0	1	0	0	
0	0	0	0	0	0	1	X	0	0	0	0	0	0	1	0	
0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	

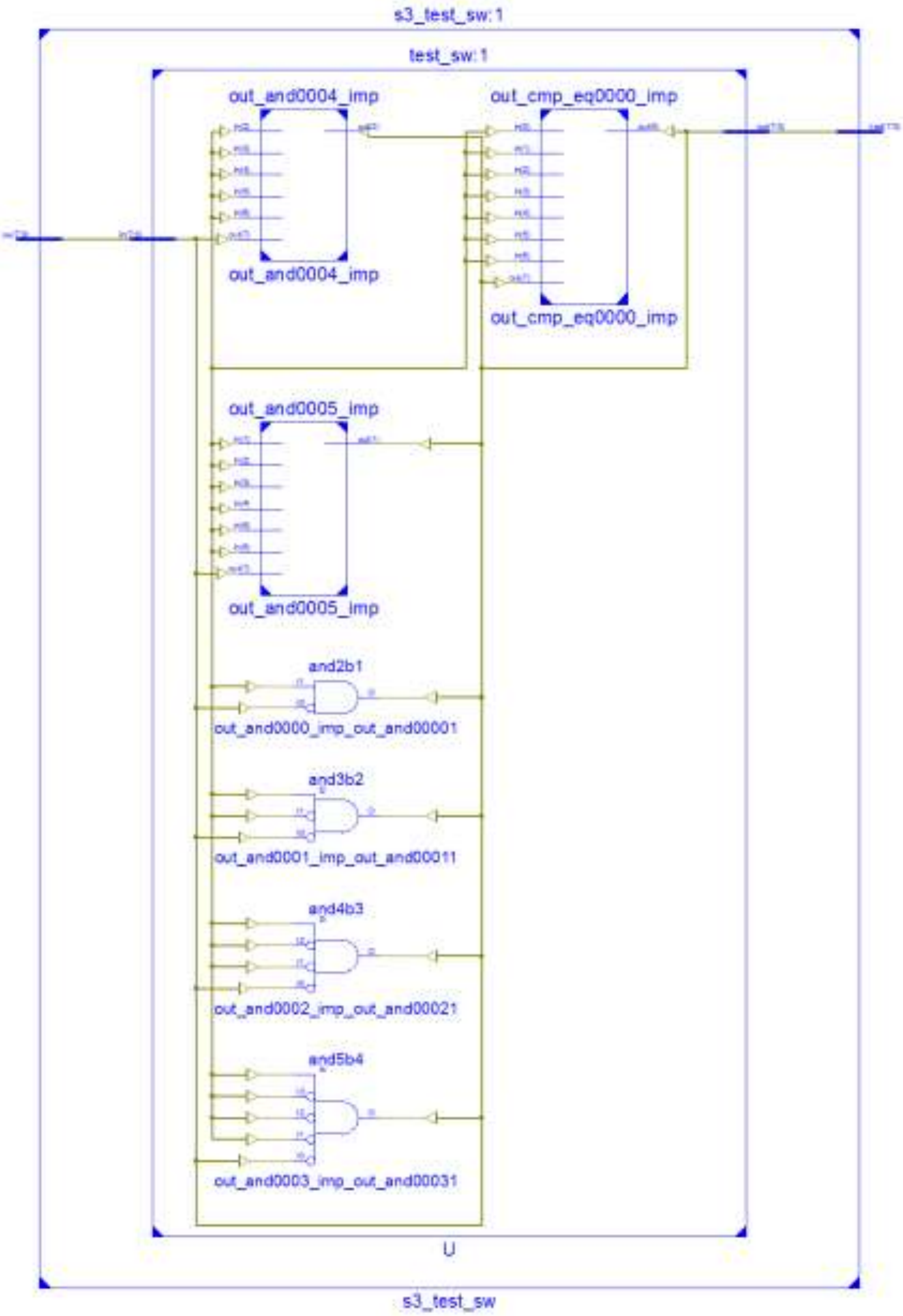
LED7 = SW7
LED6 = SW7' SW6
LED5 = SW7' SW6' SW5
LED4 = SW7' SW6' SW5' SW4
LED3 = SW7' SW6' SW5' SW4' SW3
LED2 = SW7' SW6' SW5' SW4' SW3' SW2
LED1 = SW7' SW6' SW5' SW4' SW3' SW2' SW1
LED0 = SW7' SW6' SW5' SW4' SW3' SW2' SW1' SW0

```
`timescale 1ns / 1ps
////////////////////////////////////////////////////////////////////////////////
// Company: UCSC Extension-Digital Design using FPGA
// Engineer: Jae-Yang Park
//
// Create Date:    18:10:41 02/02/2014
// Design Name:
// Module Name:    test_sw
// Project Name:
// Target Devices:
// Tool versions:
// Description:
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
////////////////////////////////////////////////////////////////////////////////
module test_sw(in, out);
        input [7:0] in;
        output [7:0] out;
        reg [7:0] out;

        always @(*) begin
                casex(in)
                        8'b1xxx_xxxx: out = 8'b10000000;
                        8'b01xx_xxxx: out = 8'b01000000;
                        8'b001x_xxxx: out = 8'b00100000;
                        8'b0001_xxxx: out = 8'b00010000;
                        8'b0000_1xxx: out = 8'b00001000;
                        8'b0000_01xx: out = 8'b00000100;
                        8'b0000_001x: out = 8'b00000010;
                        8'b0000_0001: out = 8'b00000001;
                        default: out = 8'b00000000;
                endcase
        end
endmodule

module s3_test_sw(sw, Led);
        input [7:0] sw;
        output [7:0] Led;

        test_sw U (.in(sw), .out(Led));
endmodule
```

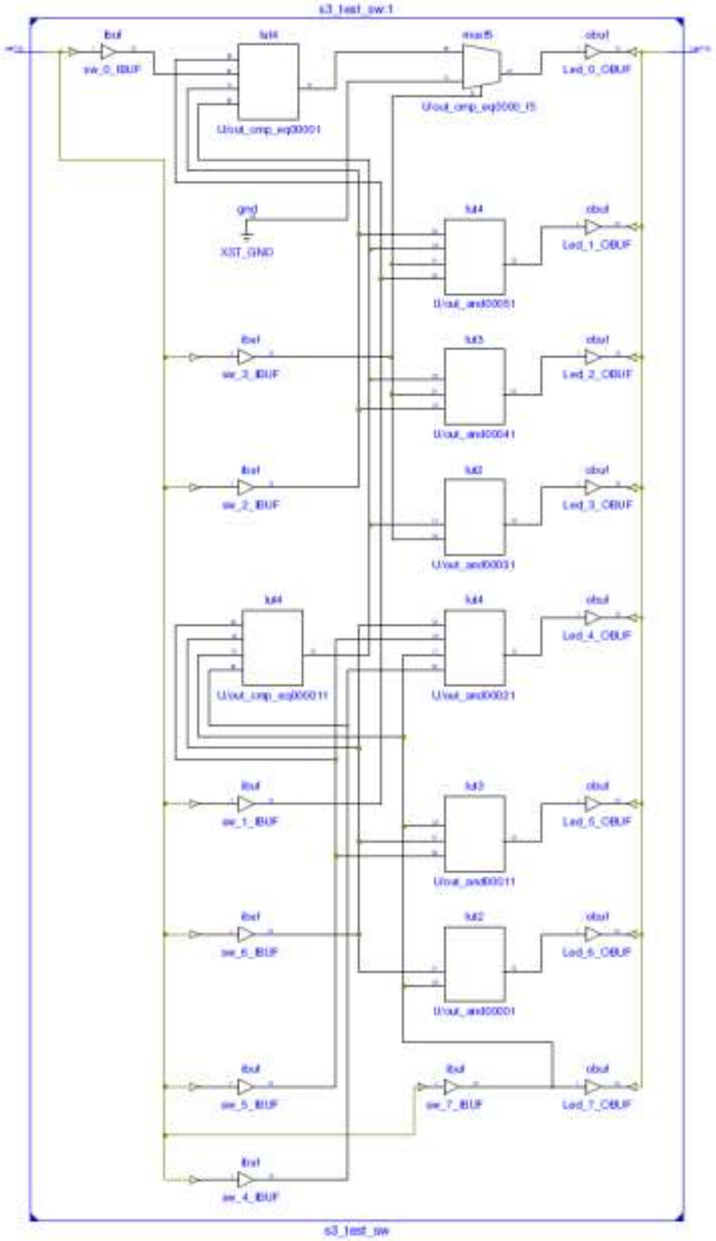
RTL Schematics






s3_test_sw:1
test_sw:1
out_and0004_imp
out_cmp_eq0000_imp
out_and0004_imp
out_cmp_eq0000_imp
out_and0005_imp
out_and0005_imp
and2b1
out_and0000_imp_out_and00001
and3b2
out_and0001_imp_out_and00011
and4b3
out_and0002_imp_out_and00021
and5b4
out_and0003_imp_out_and00031
U
s3_test_sw

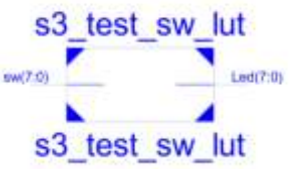
Technology Schematics

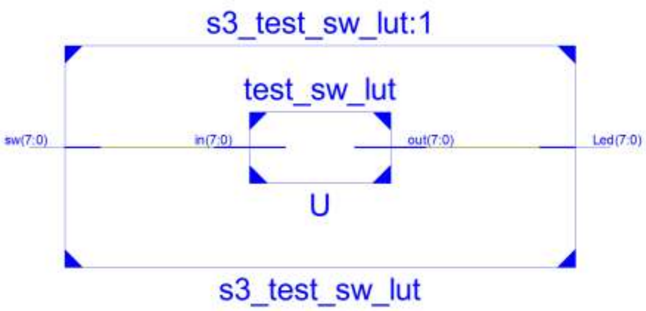


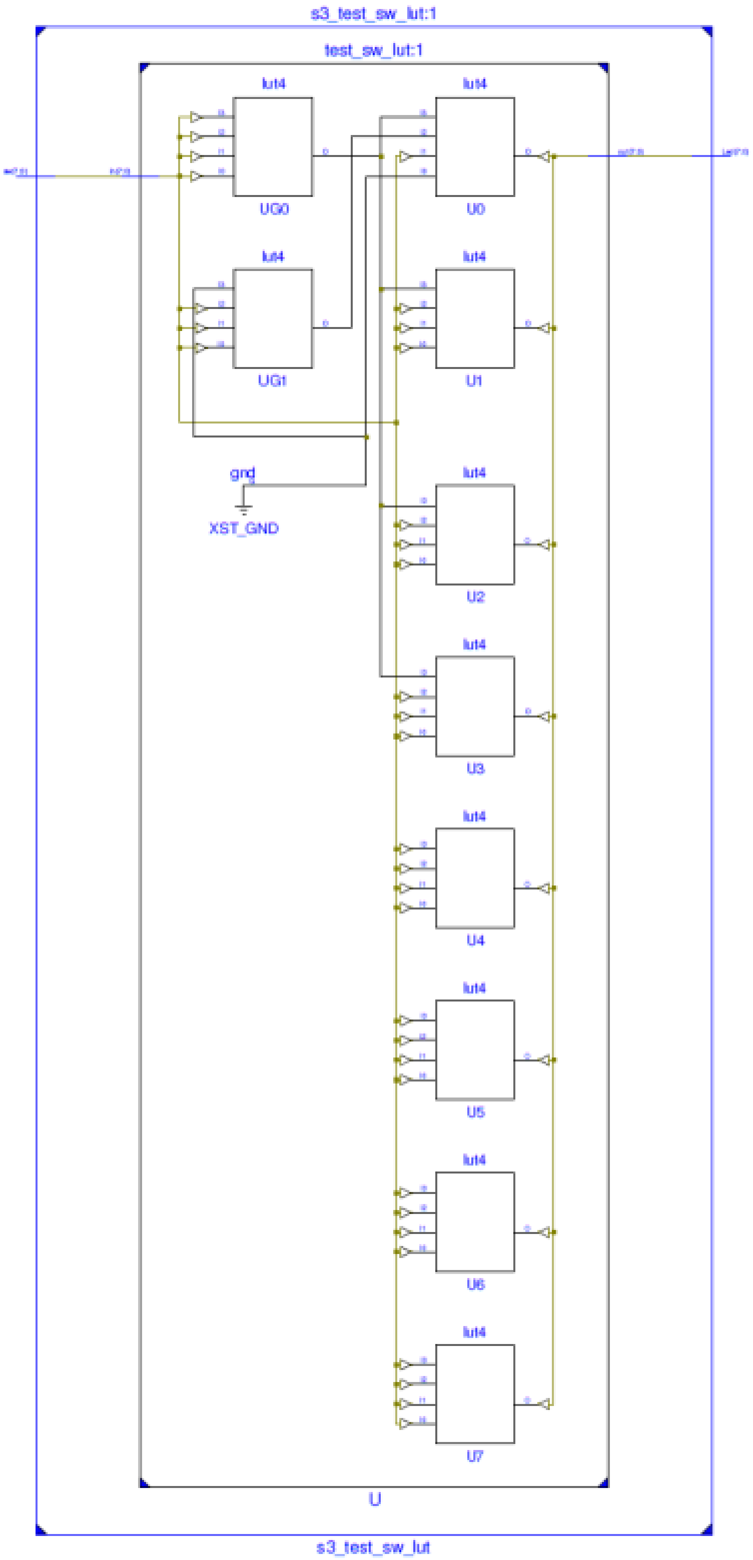
4. Structural Design using LUT

```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
// Company: UCSC Extension-Digital Design using FPGA
// Engineer: Jae-Yang Park
//
// Create Date:    18:10:41 02/02/2014
// Design Name:
// Module Name:    test_sw
// Project Name:
// Target Devices:
// Tool versions:
// Description:
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//////////////////////////////////////////////////////////////////////////////////
module test_sw_lut(in, out);
        input [7:0] in;
        output [7:0] out;
        wire out7654, out321;

        LUT4 #(16'hFF00) U7 (out[7], in[4], in[5], in[6], in[7]);
        LUT4 #(16'h00F0) U6 (out[6], in[4], in[5], in[6], in[7]);
        LUT4 #(16'h000C) U5 (out[5], in[4], in[5], in[6], in[7]);
        LUT4 #(16'h0002) U4 (out[4], in[4], in[5], in[6], in[7]);

        LUT4 #(16'h0001) UG0 (out7654, in[4], in[5], in[6], in[7]);

        LUT4 #(16'hF000) U3 (out[3], in[1], in[2], in[3], out7654);
        LUT4 #(16'h0C00) U2 (out[2], in[1], in[2], in[3], out7654);
        LUT4 #(16'h0200) U1 (out[1], in[1], in[2], in[3], out7654);

        LUT4 #(16'h0001) UG1 (out321, in[1], in[2], in[3], 1'b0);

        LUT4 #(16'h4000) U0 (out[0], 1'b0, in[0], out321, out7654);

Endmodule

module s3_test_sw_lut(sw, Led);
        input [7:0] sw;
        output [7:0] Led;

        test_sw_lut U (.in(sw), .out(Led));
endmodule
```

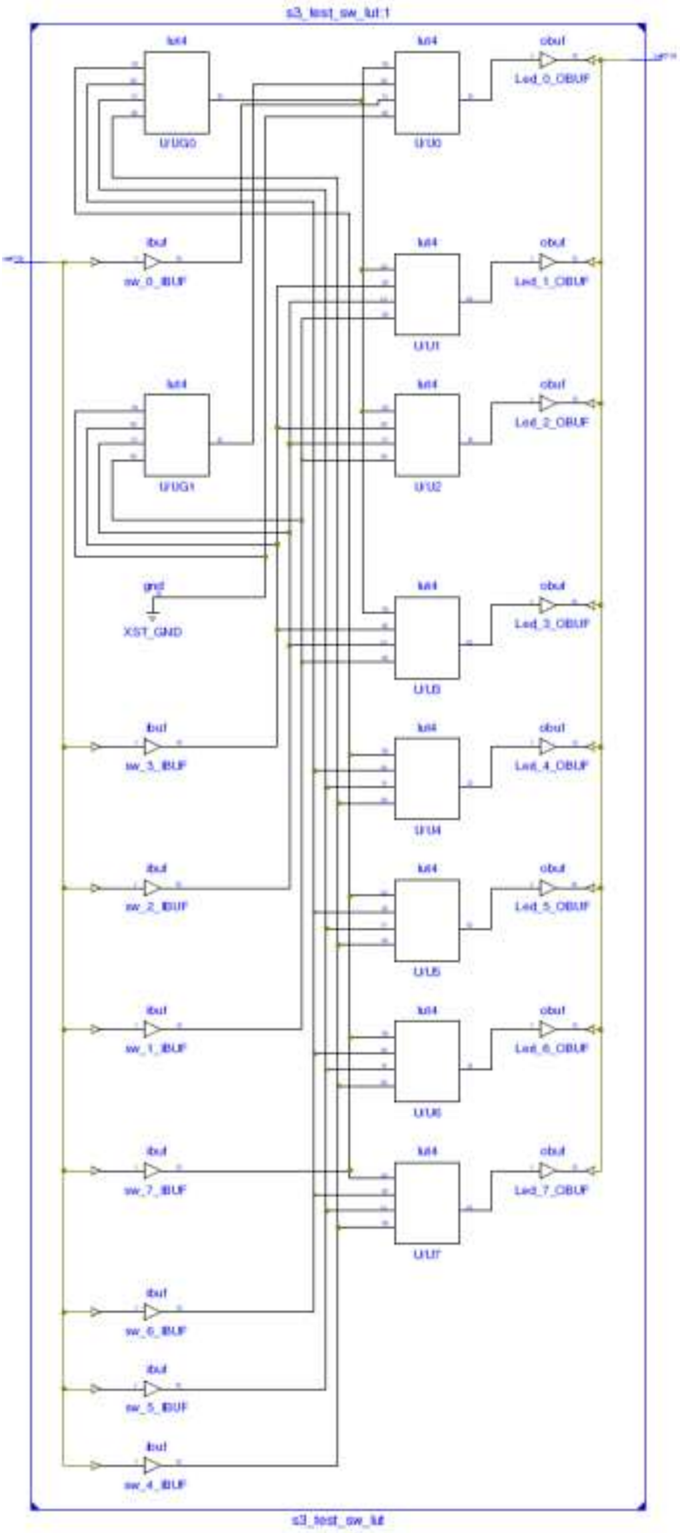
RTL Schematics

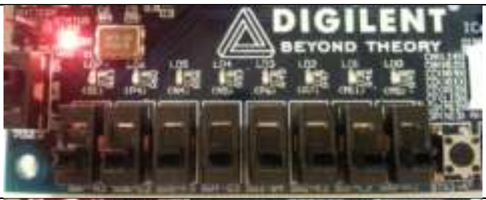
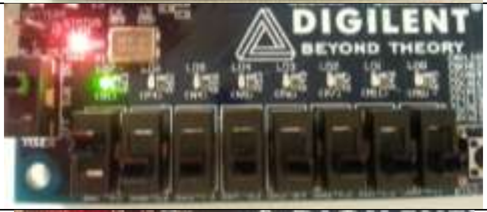
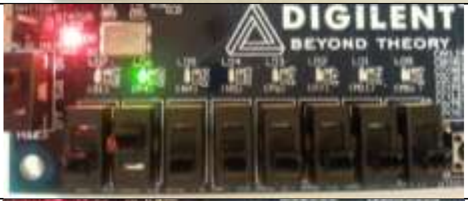
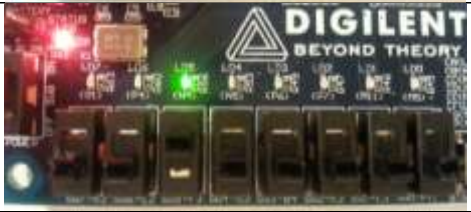
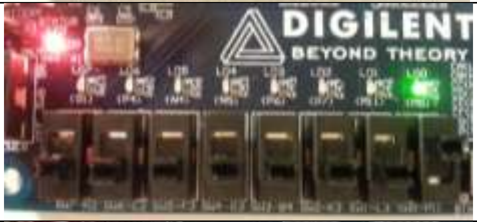
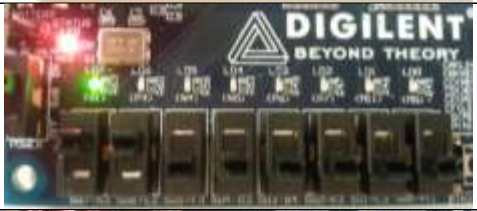
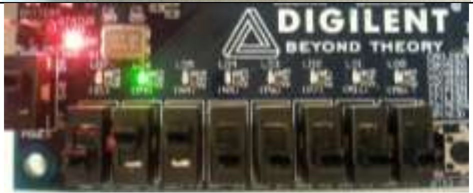
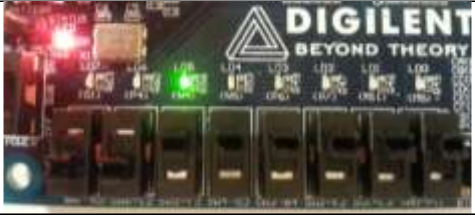





s3_test_sw_lut:1
test_sw_lut:1
lut4
lut4
UG0
U0
lut4
lut4
UG1
U1
gnd
XST_GND
lut4
U2
lut4
U3
lut4
U4
lut4
U5
lut4
U6
lut4
U7
U
s3_test_sw_lut

Technology Schematics



All Switchs are OFF	
SW7 is ON, LED7 is ON	
SW7 OFF, SW6 ON, LED6 ON	
SW7 OFF, SW6 OFF, SW5 ON LED5 ON	
SW7..SW1 OFF, SW0 ON LED0 ON	
SW7 ON, SW6 ON LED7 ON ➔ SW7 has higher priority than SW6	
SW6 ON, SW5 ON LED6 ON ➔ SW6 has higher priority than SW6	
SW5..SW0 ON LED5 ON ➔ SW5 has the highest priority	

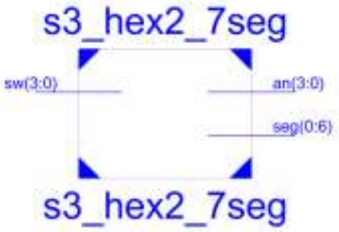
Problem 3.7.5

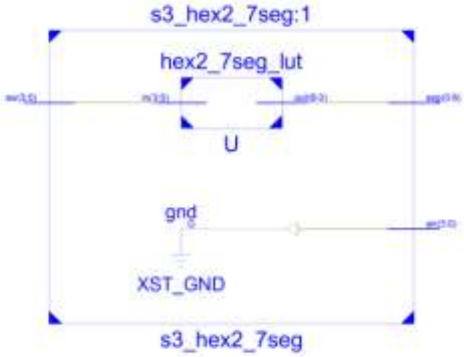
Implement the module hex2_7seg.v using structural Verilog. You should use only LUTs in your Verilog file. Use the module s3_hex2_7seg as is.

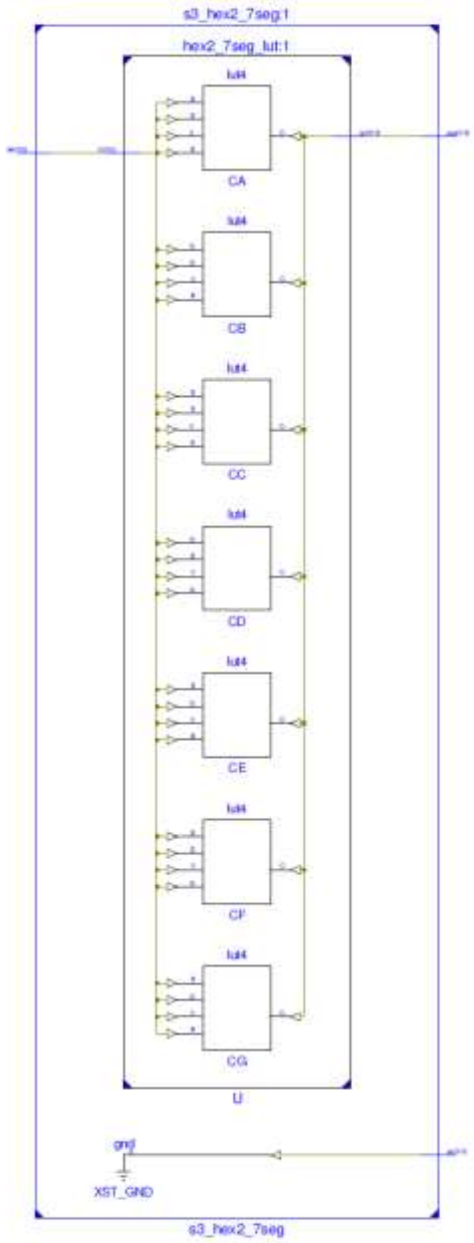
SW (0 = OFF, 1 = ON)				7 SEGMENT (0 = ON, 1 = OFF)							
SW3	SW2	SW1	SW0	a	b	c	d	e	f	g	Display
0	0	0	0	0	0	0	0	0	0	1	0
0	0	0	1	1	0	0	1	1	1	1	1
0	0	1	0	0	0	1	0	0	1	0	2
0	0	1	1	0	0	0	0	1	1	0	3
0	1	0	0	1	0	0	1	1	0	0	4
0	1	0	1	0	1	0	0	1	0	0	5
0	1	1	0	0	1	0	0	0	0	0	6
0	1	1	1	0	0	0	1	1	1	1	7
1	0	0	0	0	0	0	0	0	0	0	8
1	0	0	1	0	0	0	0	1	0	0	9
1	0	1	0	0	0	0	1	0	0	0	A
1	0	1	1	1	1	0	0	0	0	0	B
1	1	0	0	0	1	1	0	0	0	1	C
1	1	0	1	1	0	0	0	0	1	0	d
1	1	1	0	0	1	1	0	0	0	0	E
1	1	1	1	0	1	1	1	0	0	0	F
LUTMASK				**2812**	**D860**	**D004**	**8492**	**02BA**	**208E**	**1083**	

Code:

```
`timescale 1ns / 1ps
////////////////////////////////////////////////////////////////////////////////
// Company: UCSC Extension - Digital Design using FPGA
// Engineer: Jae-Yang Park
//
// Create Date:    18:53:04 02/03/2014
// Design Name:
// Module Name:    hex2_7seg_lut
// Project Name:
// Target Devices:
// Tool versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
////////////////////////////////////////////////////////////////////////////////
module hex2_7seg_lut(in, out);
        input [3:0] in;
        output [6:0] out;

        LUT4 #(16'h2812) CA (out[6], in[0], in[1], in[2], in[3]);   // a
        LUT4 #(16'hd860) CB (out[5], in[0], in[1], in[2], in[3]);   // b
        LUT4 #(16'hd004) CC (out[4], in[0], in[1], in[2], in[3]);   // c
        LUT4 #(16'h8492) CD (out[3], in[0], in[1], in[2], in[3]);   // d
        LUT4 #(16'h02ba) CE (out[2], in[0], in[1], in[2], in[3]);   // e
        LUT4 #(16'h208e) CF (out[1], in[0], in[1], in[2], in[3]);   // f
        LUT4 #(16'h1083) CG (out[0], in[0], in[1], in[2], in[3]);   // g

endmodule

module s3_hex2_7seg(sw, seg, an);
        input [3:0] sw;
        output [0:6] seg;
        output [3:0] an;

        assign an = 4'b0000;
        hex2_7seg_lut U(.in(sw), .out(seg));
endmodule
```

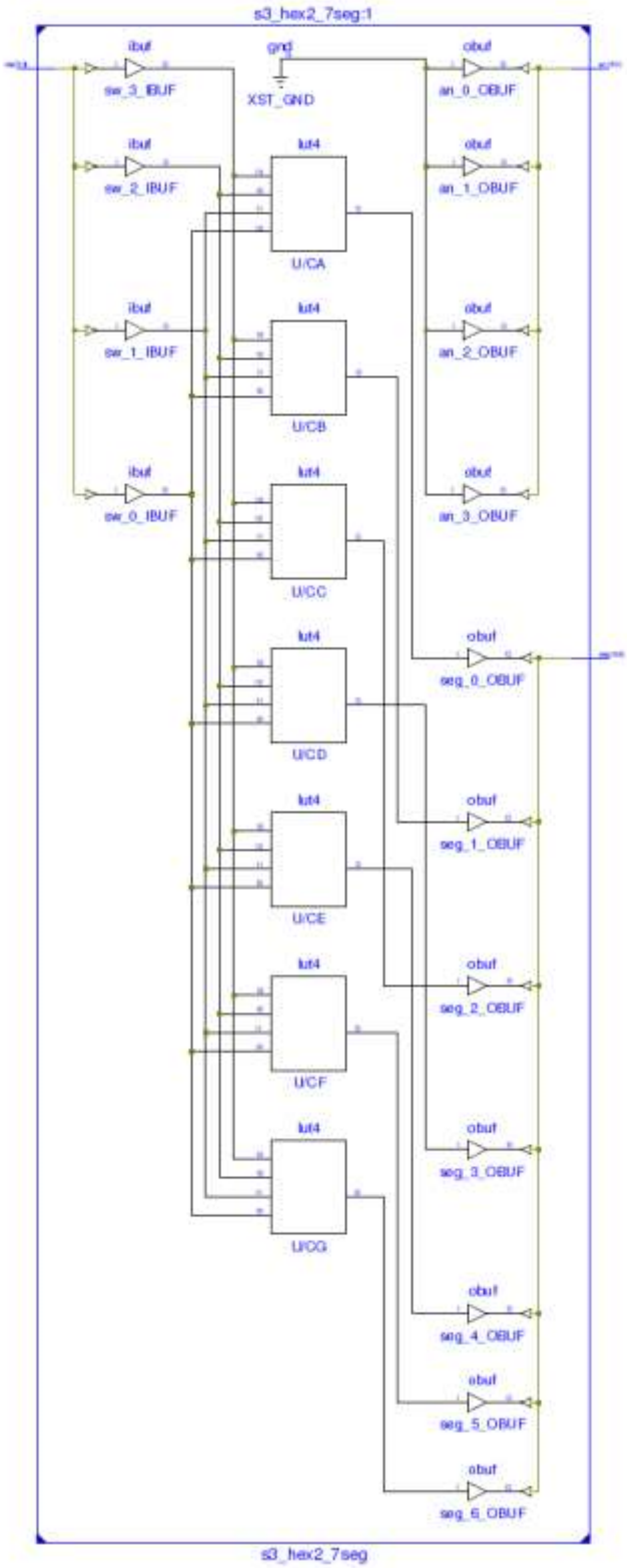




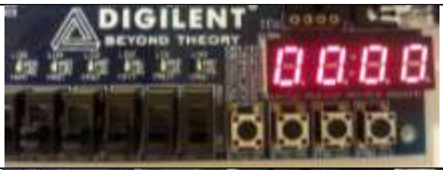
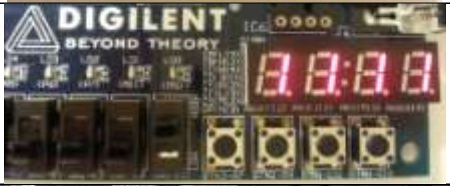
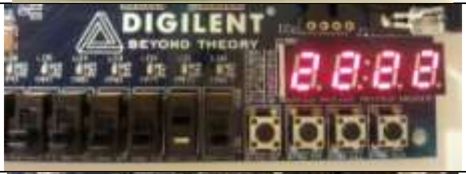
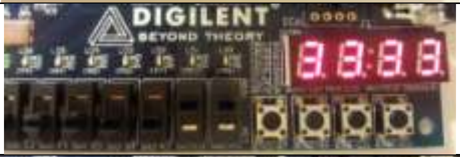
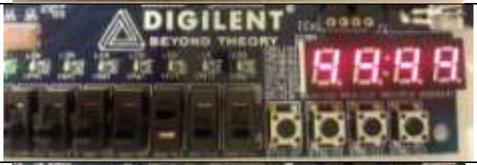
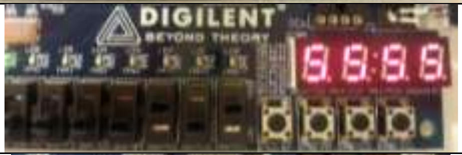
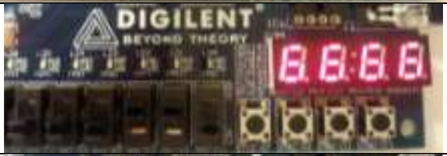
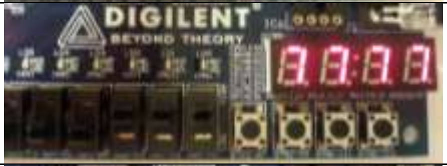
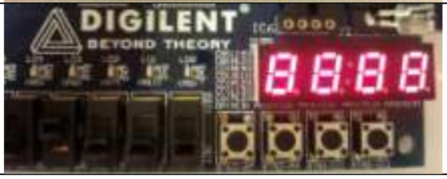


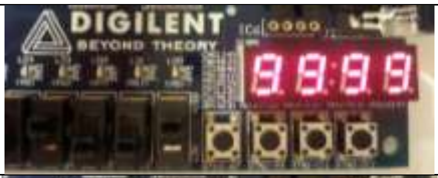
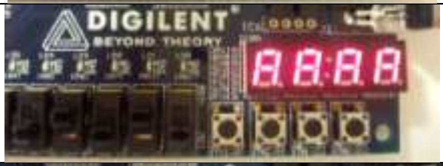
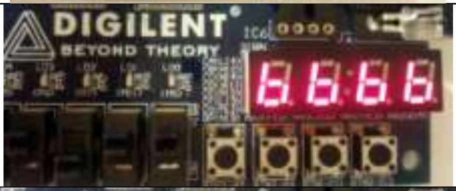
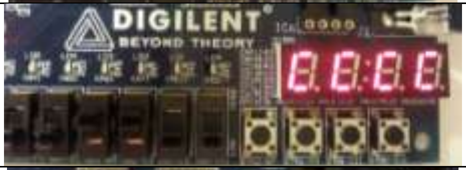
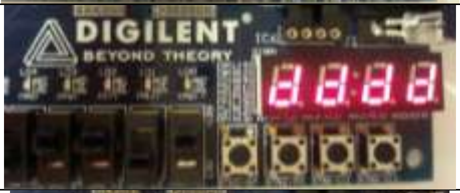
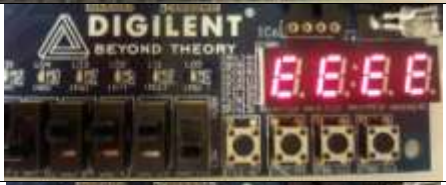
RTL Schematics

Technology Schematics



ON = 1, OFF = 0

SW3	SW2	SW1	SW0	DISPLAY
0	0	0	0	
0	0	0	1	
0	0	1	0	
0	0	1	1	
0	1	0	0	
0	1	0	1	
0	1	1	0	
0	1	1	1	
1	0	0	0	

1	0	0	1		
1	0	1	0		
1	0	1	1		
1	1	0	0		
1	1	0	1		
1	1	1	0		
1	1	1	1