UCSC Extension – Digital Design with FPGA

Jae-Yang Park (jaeyangp@gmail.com)

Problem 2.6.1

1. Show that lutmask 8000 implements 4-inputs AND gate

a	b	c	d	F
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	**1**	**1**	**1**	**1**

F = a b c d

LUTMASK = 8000

2. Implement a 4 inputs XOR gate using a LUT4

a	b	c	d	F
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	1
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0

F = a’b’c’d +a’b’cd’ +a’bc’d’ +a’bc’d’ +ab’c’d’ +ab’cd +abc’d+abcd’

LUTMASK = 6996

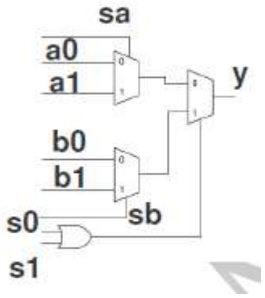
3. Implement the function below using a LUT4

a	b	c	d	F
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

F = abcd + a’b’c’d’ + ad’ + a’bc’ + b’d + a’bc’
+ abc’d + bcd + a’cd’
= TAUTOLOGY

LUTMASK = FFFF

Problem 2.6.2



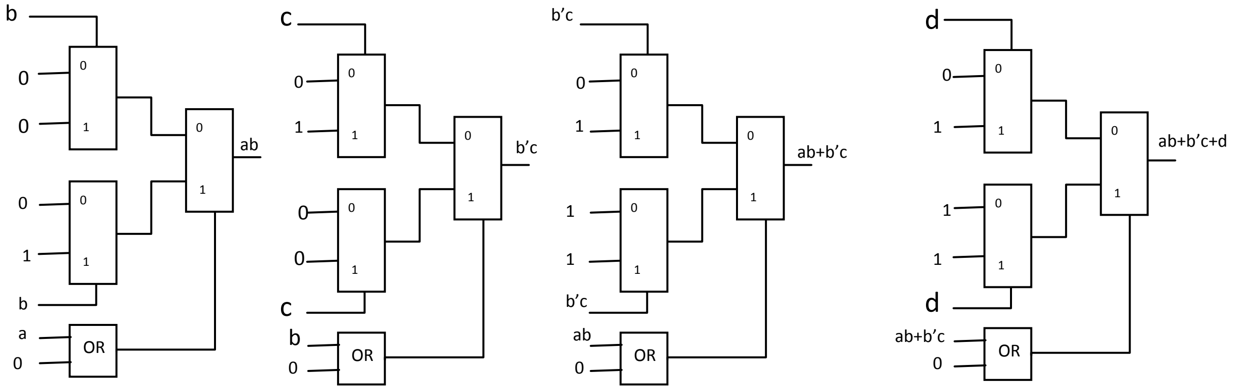
1. Implement all the 16 possible two-input functions using the above block Complete the table as follows:

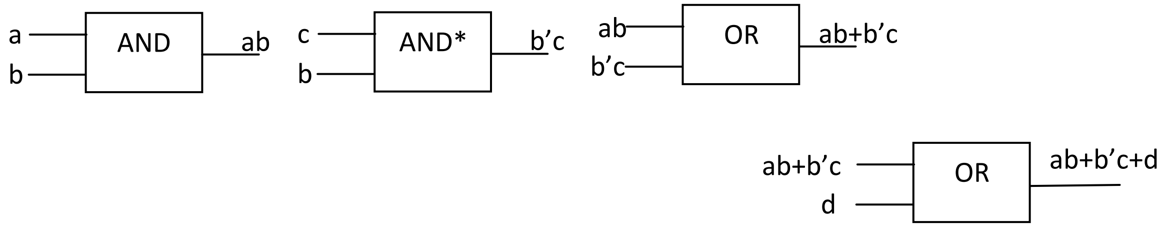
		a0	a1	sa	b0	b1	sb	s0	s1
1	XOR (a, b)	0	1	b	1	0	b	a	0
2	AND (a, b)	0	0	b	0	1	b	a	0
3	OR (a, b)	0	1	b	1	1	b	a	0
4	NAND (a, b)	1	1	b	1	0	b	a	0
5	NOR (a, b)	1	0	b	0	0	b	a	0
6	XNOR (a, b)	1	0	b	0	1	b	a	0
7	A < B	0	1	b	0	0	b	a	0
8	A > B	0	0	b	1	0	b	a	0
9	A LE B	1	1	b	0	1	b	a	0
10	A GE B	1	0	b	1	1	b	a	0
11	BUFFER (A)	0	0	b	1	1	b	a	0
12	BUFFER (B)	0	1	b	1	1	b	a	0
13	NOT (A)	1	1	b	0	0	b	a	0
14	NOT (B)	1	0	b	1	0	b	a	0
15	CONTRIDICTION	0	0	b	0	0	b	a	0
16	TAUTOLOGY	1	1	b	1	1	b	a	0

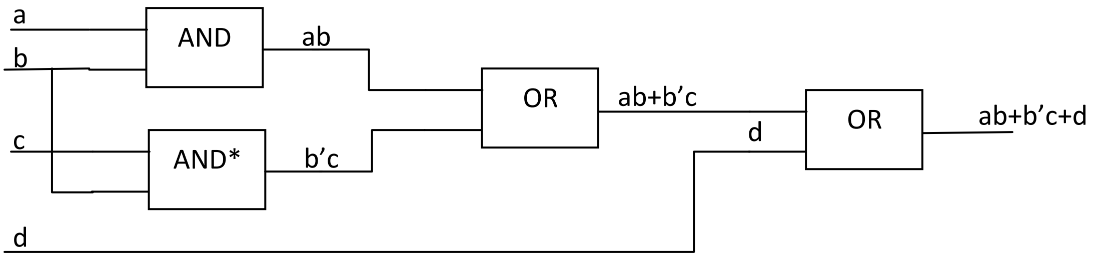
2. Implement f = ab + b’c + d using the above block

a	b	c	d	ab	b’c	ab+b’c+d
0	0	0	0	0	0	0
0	0	0	1	0	0	1
0	0	1	0	0	1	1
0	0	1	1	0	1	1
0	1	0	0	0	0	0
0	1	0	1	0	0	1
0	1	1	0	0	0	0
0	1	1	1	0	0	1
1	0	0	0	0	0	0
1	0	0	1	0	0	1
1	0	1	0	0	1	1
1	0	1	1	0	1	1
1	1	0	0	1	0	1
1	1	0	1	1	0	1
1	1	1	0	1	0	1
1	1	1	1	1	0	1

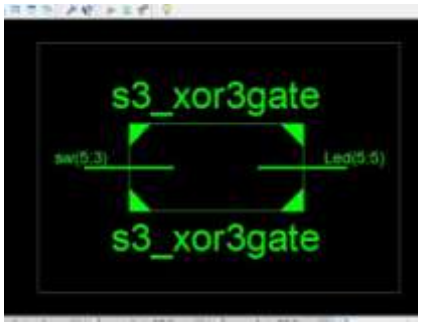
		a0	a1	sa	b0	b1	sb	s0	s1
1	AND (a, b)	0	0	b	0	1	b	a	0
3	OR (a, b)	0	1	b	1	1	b	a	0

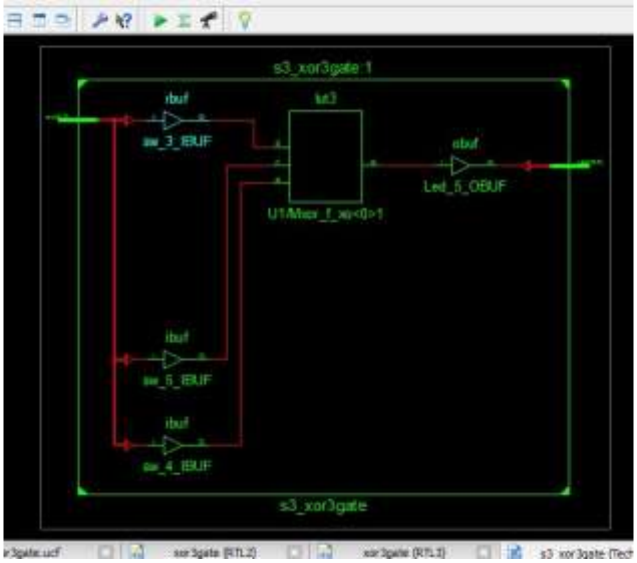


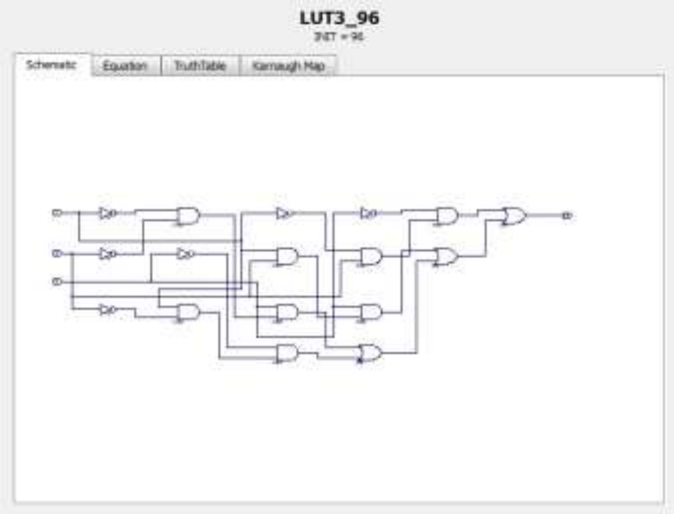


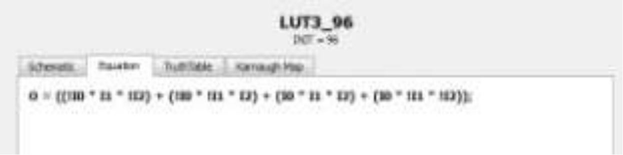


Problem 3.7.1

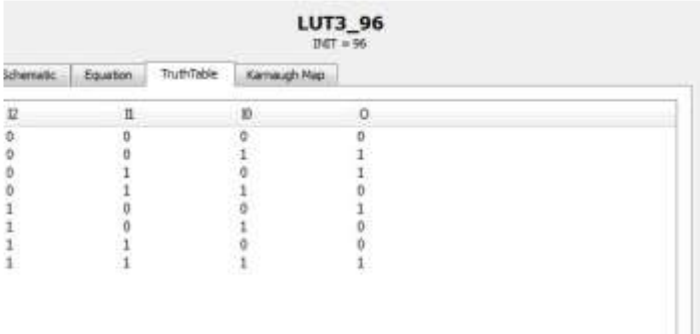




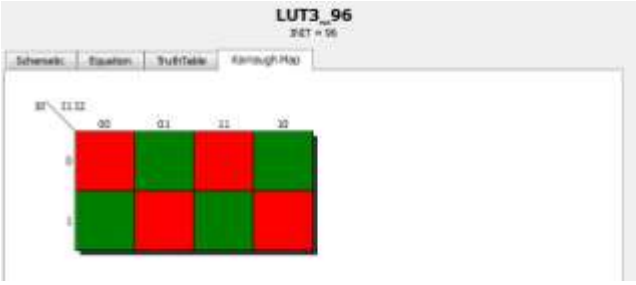




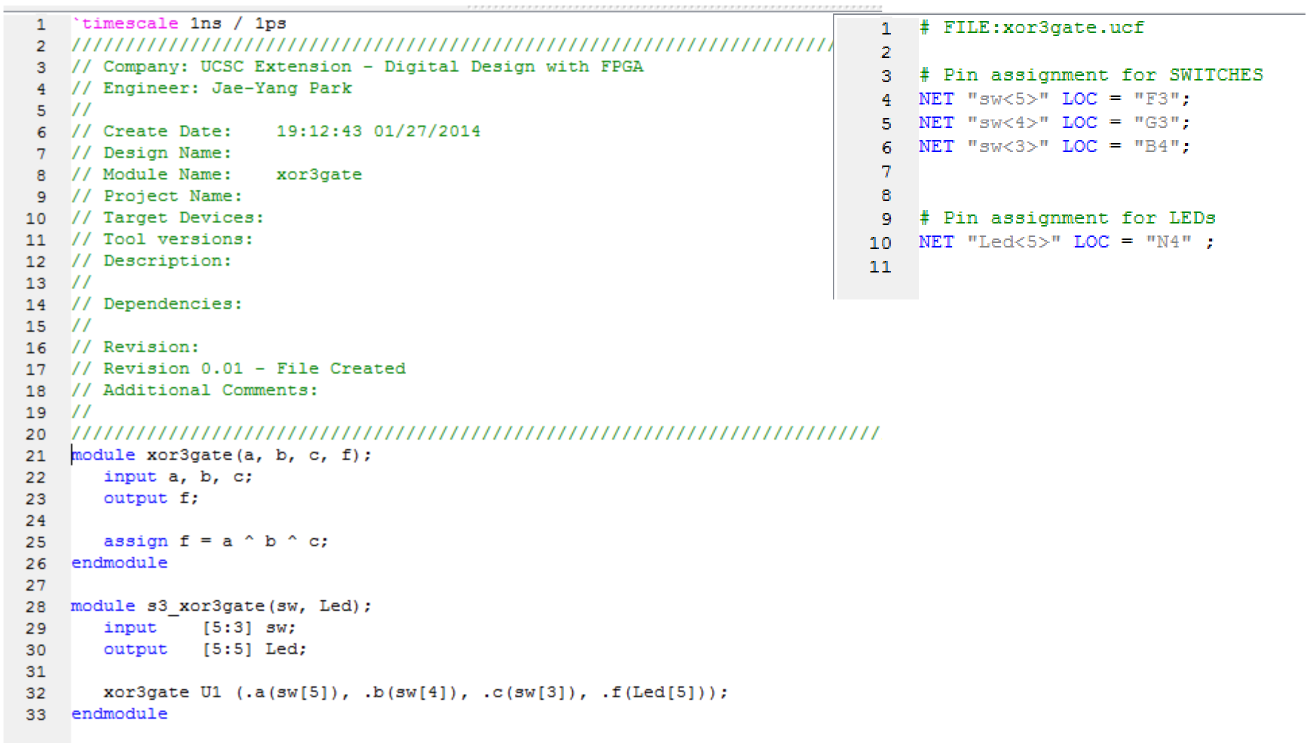
```
O = ((!I0 * I1 * !I2) + (!I0 * !I1 * I2) + (I0 * I1 * I2) + (I0 * !I1 * !I2));
```



I2	I1	I0	O
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

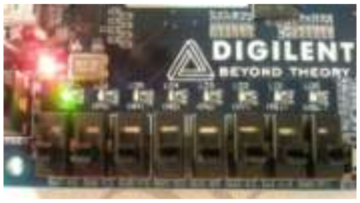
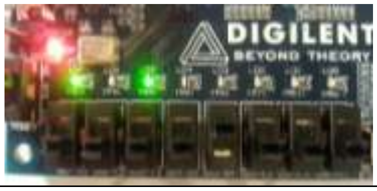
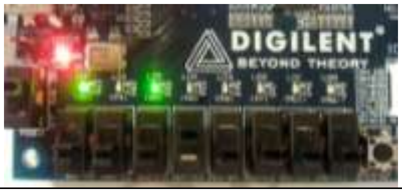
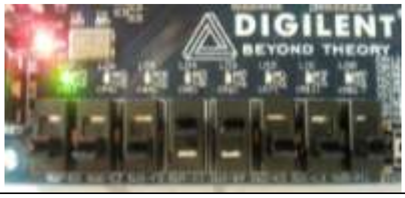


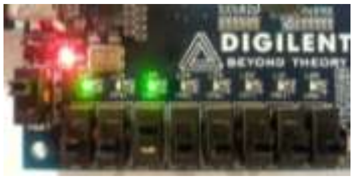
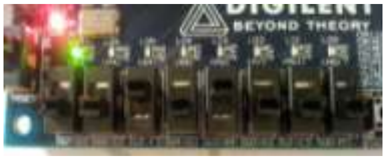
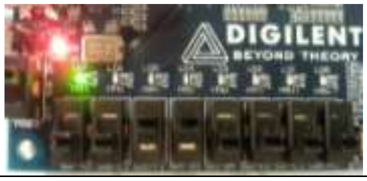
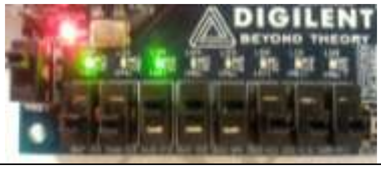
I0 \ I1 I2	00	01	11	10
0	🟥	🟩	🟥	🟩
1	🟩	🟥	🟩	🟥



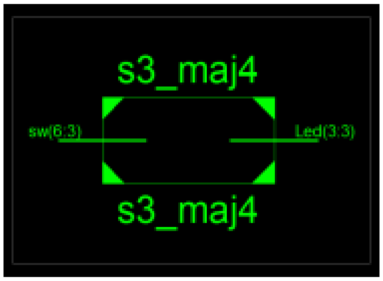
```
`timescale 1ns / 1ps
////////////////////////////////////////////////////////////////////////////////
// Company: UCSC Extension - Digital Design with FPGA
// Engineer: Jae-Yang Park
//
// Create Date:    19:12:43 01/27/2014
// Design Name:
// Module Name:    xor3gate
// Project Name:
// Target Devices:
// Tool versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
////////////////////////////////////////////////////////////////////////////////
module xor3gate(a, b, c, f);
   input a, b, c;
   output f;

   assign f = a ^ b ^ c;
endmodule

module s3_xor3gate(sw, Led);
   input    [5:3] sw;
   output   [5:5] Led;

   xor3gate U1 (.a(sw[5]), .b(sw[4]), .c(sw[3]), .f(Led[5]));
endmodule
```

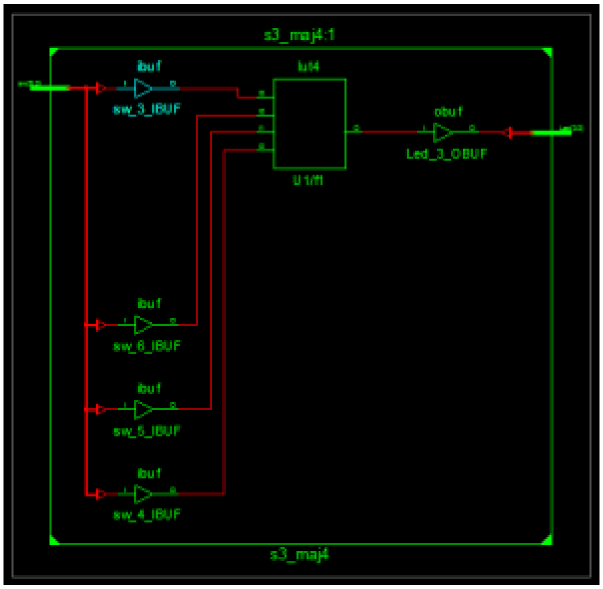
```
# FILE:xor3gate.ucf

# Pin assignment for SWITCHES
NET "sw<5>" LOC = "F3";
NET "sw<4>" LOC = "G3";
NET "sw<3>" LOC = "B4";


# Pin assignment for LEDs
NET "Led<5>" LOC = "N4" ;

```

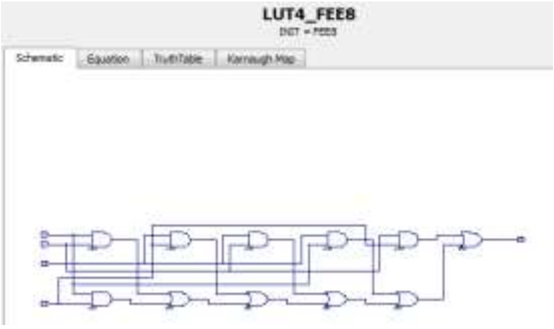
SW5	**SW4**	**SW3**	
OFF	OFF	OFF	
OFF	OFF	ON	
OFF	ON	OFF	
OFF	ON	ON	

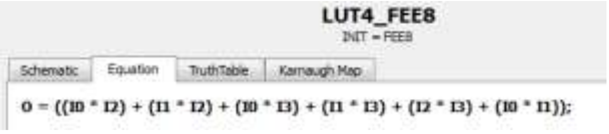
ON	OFF	OFF	
ON	OFF	ON	
ON	ON	OFF	
ON	ON	ON	

Problem 3.7.1

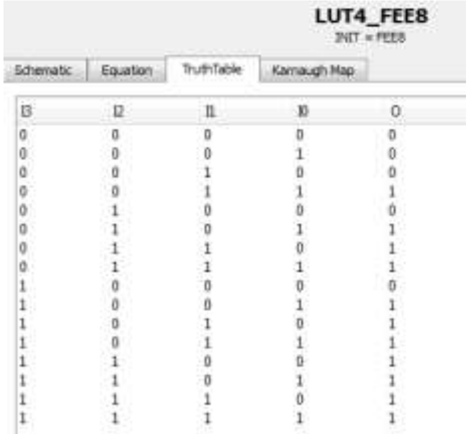




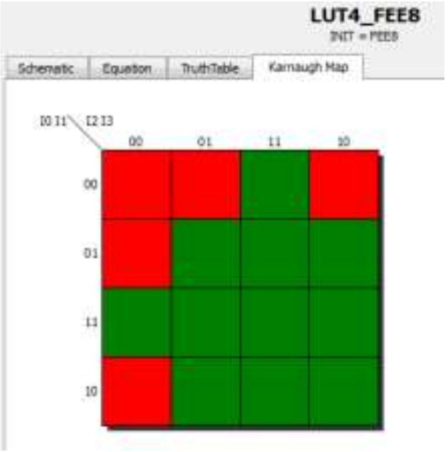




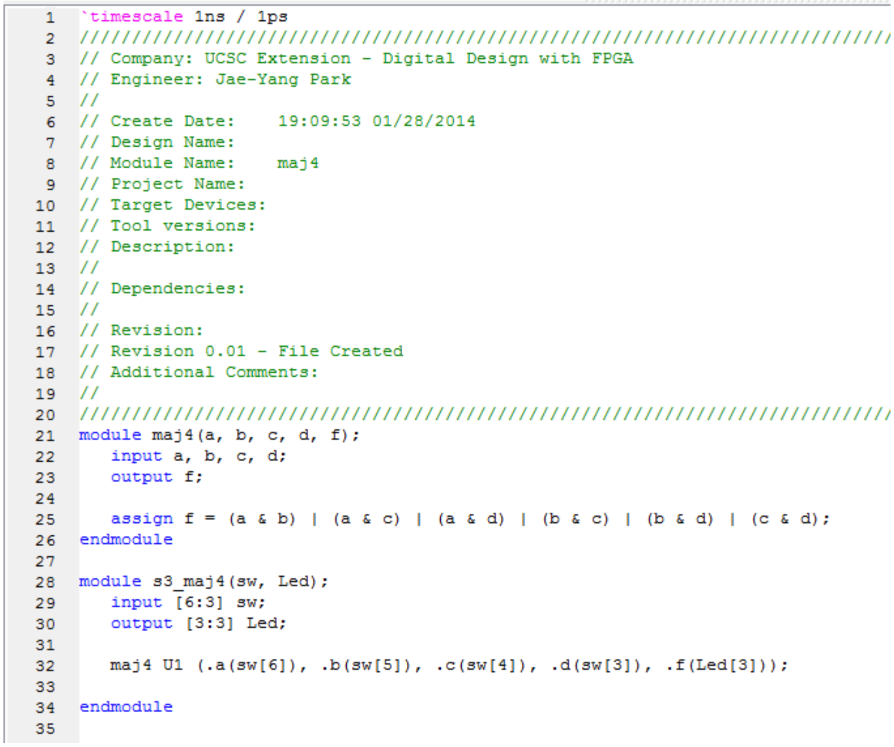
O = ((I0 * I2) + (I1 * I2) + (I0 * I3) + (I1 * I3) + (I2 * I3) + (I0 * I1));



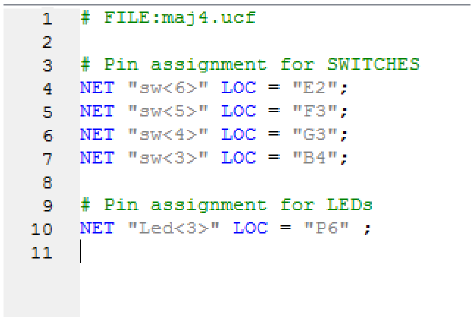
I3	I2	I1	I0	O
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	0
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1



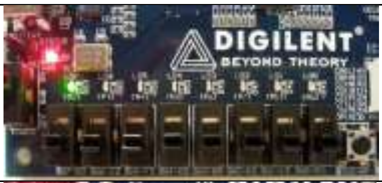
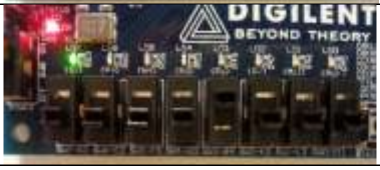
I0 I1 \ I2 I3	00	01	11	10
00	🟥	🟥	🟩	🟥
01	🟥	🟩	🟩	🟩
11	🟩	🟩	🟩	🟩
10	🟥	🟩	🟩	🟩

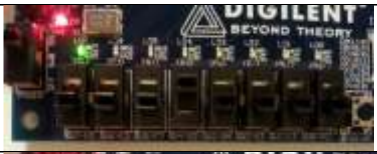
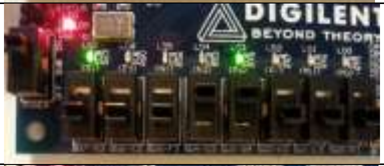
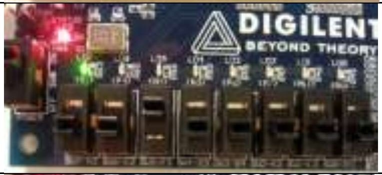
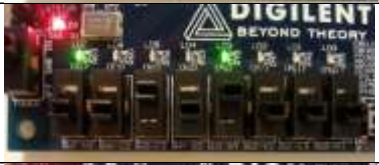
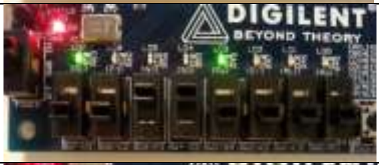
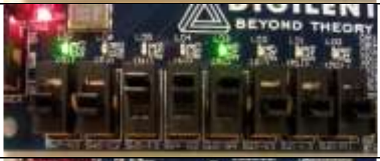
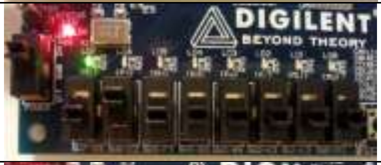
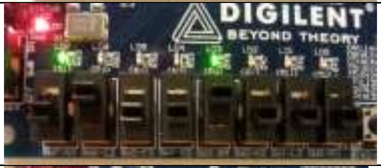
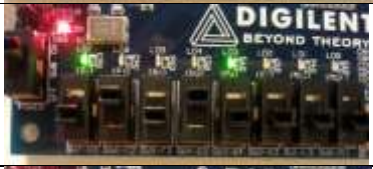
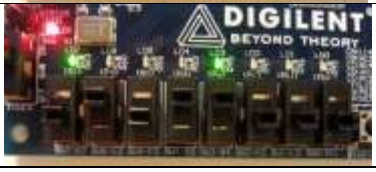


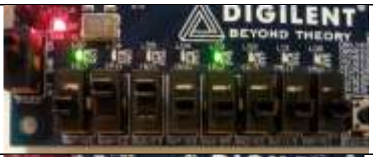
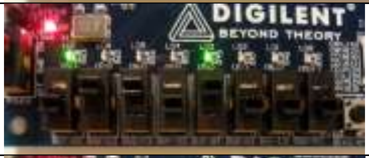
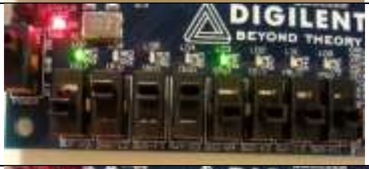
```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
// Company: UCSC Extension - Digital Design with FPGA
// Engineer: Jae-Yang Park
//
// Create Date:    19:09:53 01/28/2014
// Design Name:
// Module Name:    maj4
// Project Name:
// Target Devices:
// Tool versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////
module maj4(a, b, c, d, f);
   input a, b, c, d;
   output f;

   assign f = (a & b) | (a & c) | (a & d) | (b & c) | (b & d) | (c & d);
endmodule

module s3_maj4(sw, Led);
   input [6:3] sw;
   output [3:3] Led;

   maj4 U1 (.a(sw[6]), .b(sw[5]), .c(sw[4]), .d(sw[3]), .f(Led[3]));

endmodule

```



```
# FILE:maj4.ucf

# Pin assignment for SWITCHES
NET "sw<6>" LOC = "E2";
NET "sw<5>" LOC = "F3";
NET "sw<4>" LOC = "G3";
NET "sw<3>" LOC = "B4";

# Pin assignment for LEDs
NET "Led<3>" LOC = "P6" ;

```

SW6	SW5	SW4	SW3	
OFF	OFF	OFF	OFF	
OFF	OFF	OFF	ON	

OFF	OFF	ON	OFF	
OFF	OFF	ON	ON	
OFF	ON	OFF	OFF	
OFF	ON	OFF	ON	
OFF	ON	ON	OFF	
OFF	ON	ON	ON	
ON	OFF	OFF	OFF	
ON	OFF	OFF	ON	
ON	OFF	ON	OFF	
ON	OFF	ON	ON	

ON	ON	OFF	OFF	
ON	ON	OFF	ON	
ON	ON	ON	OFF	
ON	ON	ON	ON