**UCSC Extension – Digital Design with FPGA**

**Jae-Yang Park** ([jaeyangp@gmail.com](mailto:jaeyangp@gmail.com))

**Problem 2.6.1**

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

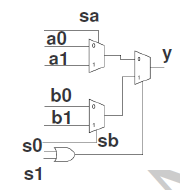
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| a | b | c | d | F  F = a b c d  LUTMASK = 8000 |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 0 |
| 1 | 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** |

1. Implement a 4 inputs XOR gate using a LUT4

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| a | b | c | d | F  F = a’b’c’d +a’b’cd’ +a’bc’d’ +a’bc’d’ +ab’c’d’ +ab’cd +abc’d+abcd’  LUTMASK = 6996 |
| 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 |

1. Implement the function below using a LUT4

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| a | b | c | d | F  F = abcd + a’b’c’d’ + ad’ + a’bc’ + b’d + a’bc’ + abc’d + bcd + a’cd’ = TAUTOLOGY  LUTMASK = FFFF |
| 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 |

  
**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 |

OR

ab+b’c

d

ab+b’c+d

OR

ab

b’c

ab+b’c

AND\*

c

b

b’c

AND

a

b

ab

ab+b’c+d

0

ab+b’c

d

d

0

1

1

1

0

0

ab+b’c

0

ab

b’c

b’c

1

1

1

b’c

0

0

0

b

c

1

c

ab

b

0

a

b

0

1

0

0

0

1

0

1

0

1

OR

0

1

0

1

0

1

OR

0

1

0

1

0

1

OR

0

1

0

1

0

1

OR

AND

a

b

ab

AND\*

c

b’c

OR

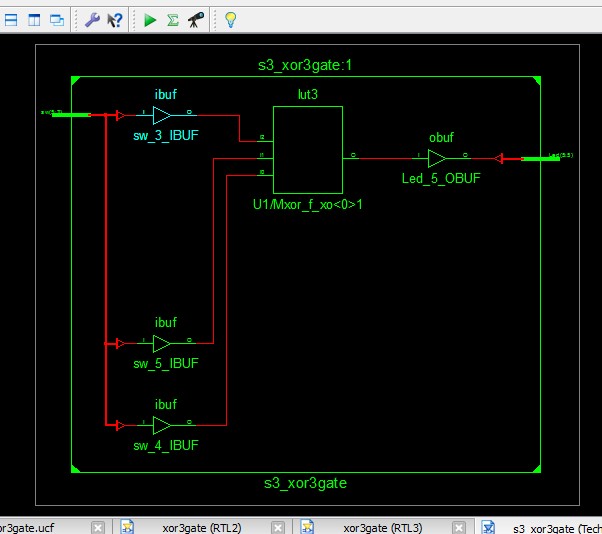
ab+b’c

OR

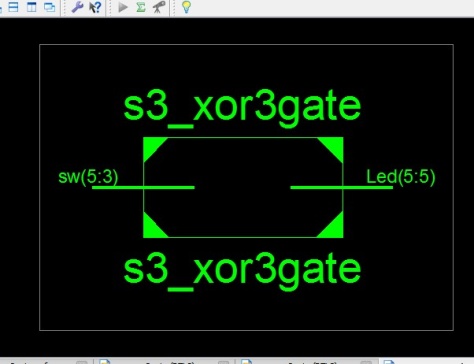
d

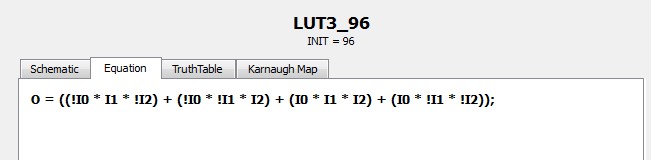
ab+b’c+d

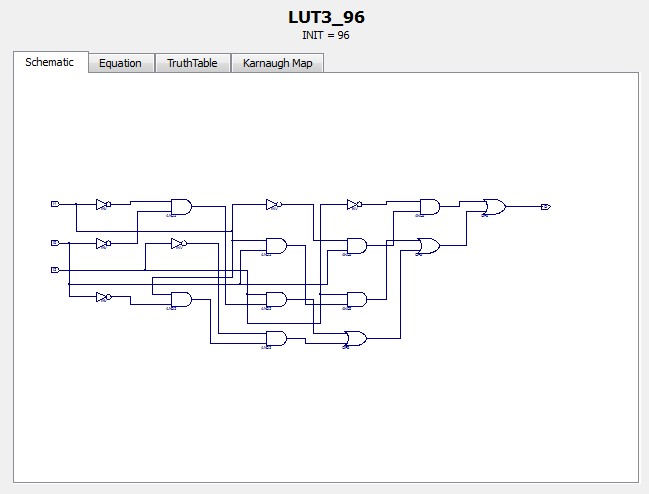
d

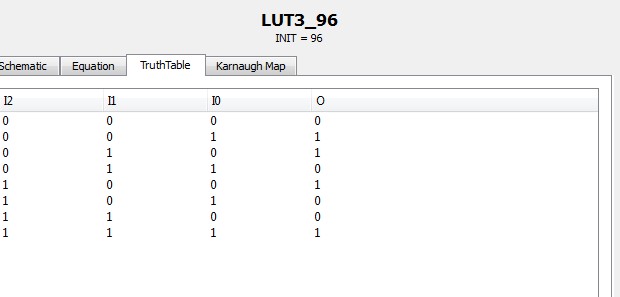
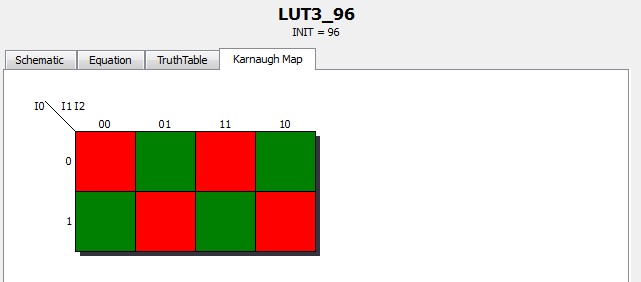


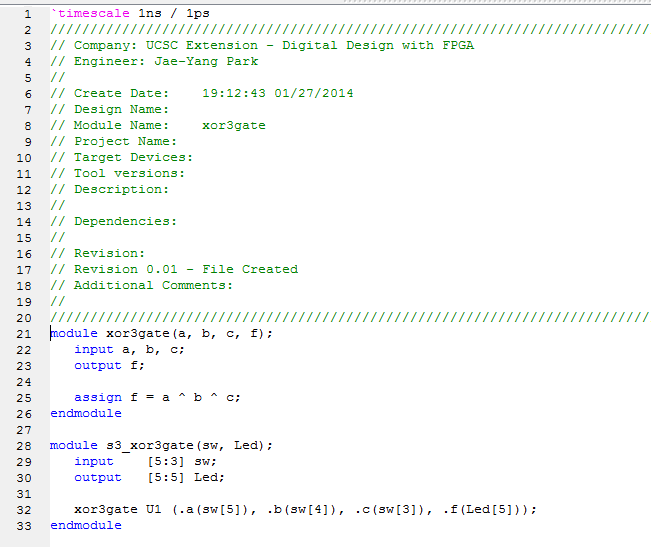
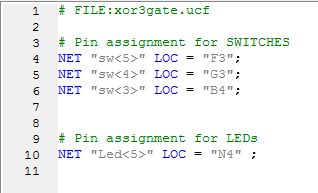
**Problem 3.7.1**





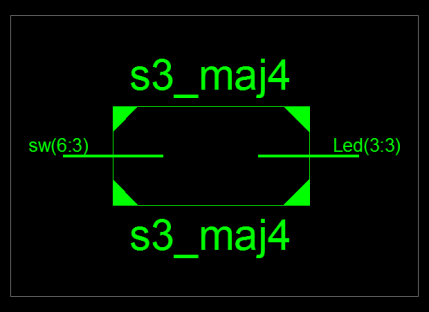
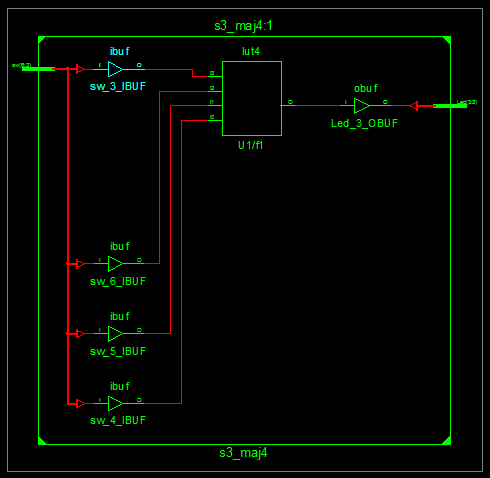


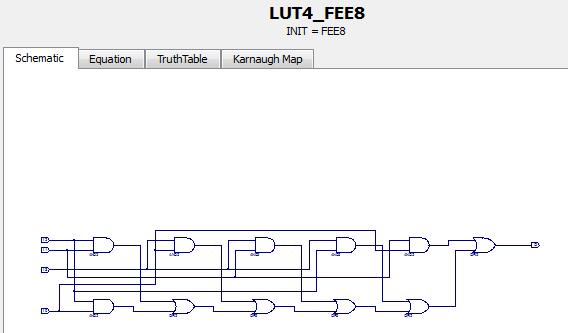


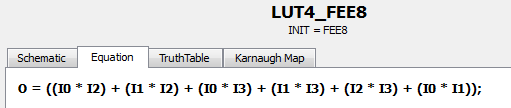


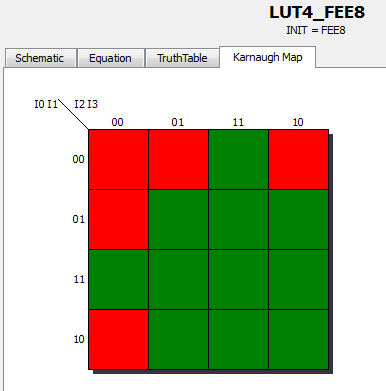
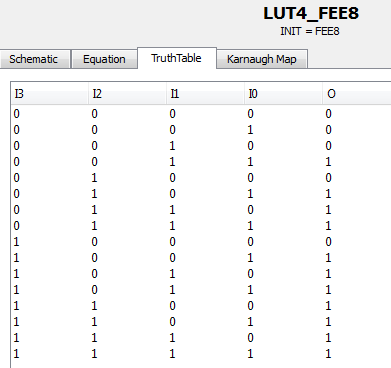
|  |  |  |  |
| --- | --- | --- | --- |
| **SW5** | **SW4** | **SW3** |  |
| OFF | OFF | OFF | 20140127_193751.jpg |
| OFF | OFF | ON | 20140127_193806.jpg |
| OFF | ON | OFF | 20140127_193815.jpg |
| OFF | ON | ON | 20140127_201627.jpg |
| ON | OFF | OFF | 20140127_193825.jpg |
| ON | OFF | ON | 20140127_201601.jpg |
| ON | ON | OFF | 20140127_193846.jpg |
| ON | ON | ON | 20140127_193838.jpg |

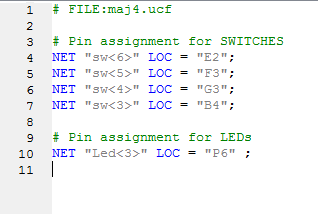
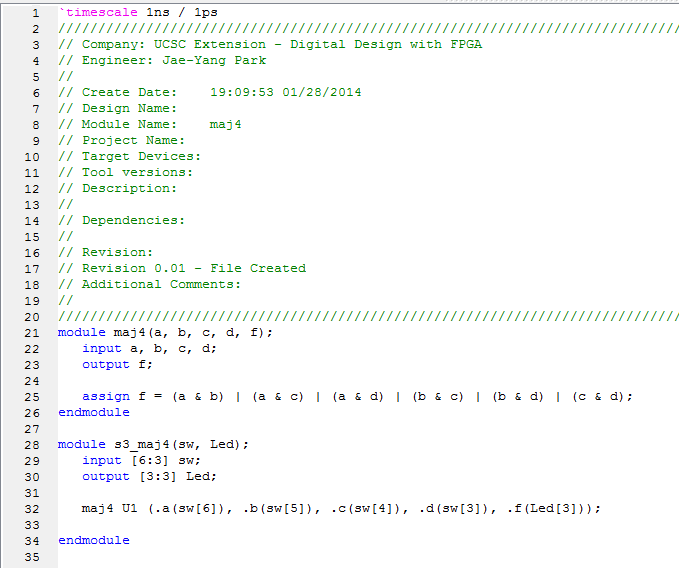
**Problem 3.7.1**











|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SW6** | **SW5** | **SW4** | **SW3** |  |
| OFF | OFF | OFF | OFF | 20140128_194523.jpg |
| OFF | OFF | OFF | ON | 20140128_194539.jpg |
| OFF | OFF | ON | OFF | 20140128_194549.jpg |
| OFF | OFF | ON | ON | 20140128_194600.jpg |
| OFF | ON | OFF | OFF | 20140128_194614.jpg |
| OFF | ON | OFF | ON | 20140128_194622.jpg |
| OFF | ON | ON | OFF | 20140128_194631.jpg |
| OFF | ON | ON | ON | 20140128_194649.jpg |
| ON | OFF | OFF | OFF | 20140128_194658.jpg |
| ON | OFF | OFF | ON | 20140128_194714.jpg |
| ON | OFF | ON | OFF | 20140128_194721.jpg |
| ON | OFF | ON | ON | 20140128_194729.jpg |
| ON | ON | OFF | OFF | 20140128_194738.jpg |
| ON | ON | OFF | ON | 20140128_194753.jpg |
| ON | ON | ON | OFF | 20140128_194803.jpg |
| ON | ON | ON | ON | 20140128_194814.jpg |