

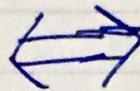
DeMorgan's Theorem

F 1

\equiv

F 2

$$\overline{A \cdot B}$$



$$\bar{A} + \bar{B}$$

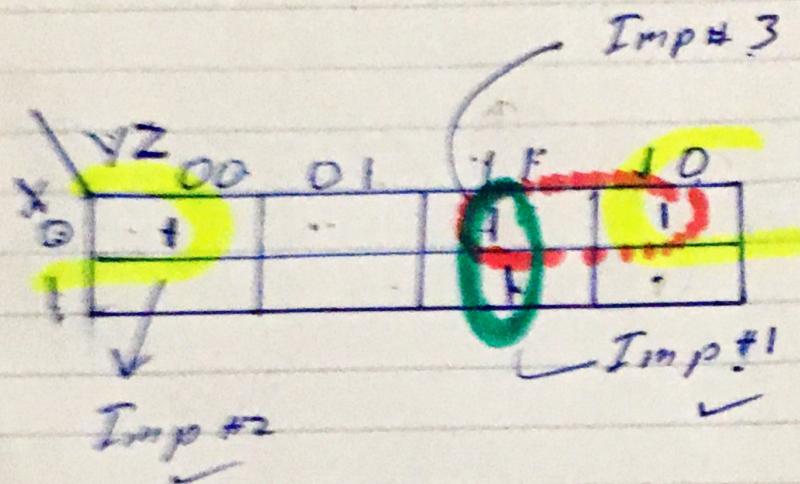
Inputs		Output
A	B	$A \cdot B'$
0	0	0
0	1	0
1	0	0
1	1	1

Input	$A \cdot B$	$\bar{A} \cdot \bar{B}$	$\bar{A} + \bar{B}$
00	0	1	1
01	0	0	1
10	0	0	1
11	1	0	0

Three Variable K-map

Minimize Boolean Expression below.

$$F = \bar{x}\bar{y}\bar{z} + \bar{x}yz + \bar{x}yz + xy\bar{z}$$



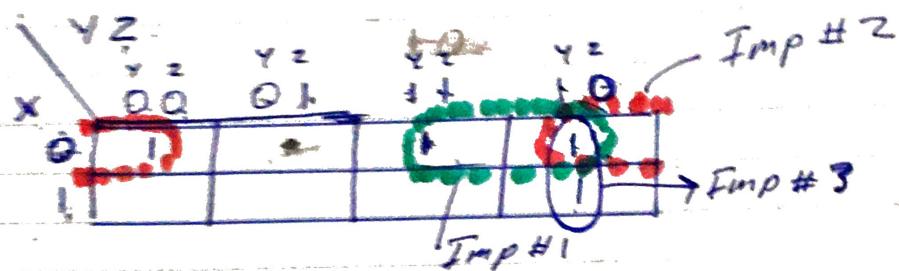
$$F = yz + x\bar{z} + \bar{x}y$$

Three Variable K-Map

Minimized form of logical Expression

$$F = \cancel{xyz}$$

$$F = \bar{x} \cdot \bar{y} \cdot \bar{z} + \bar{x} \cdot y \cdot \bar{z} + \bar{x} \cdot y \cdot z + x \cdot y \cdot \bar{z}$$



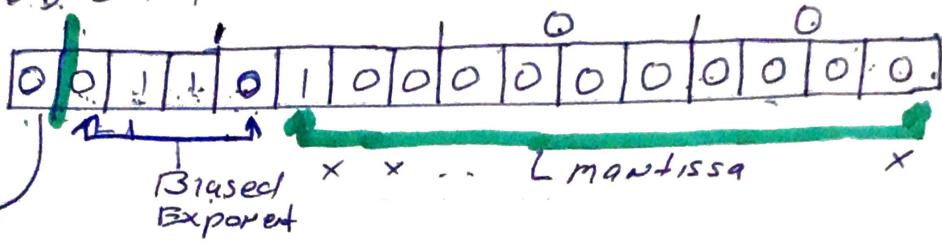
001 Adjacent cell \rightarrow 011

$$F = \bar{x} \cdot y + y \cdot \bar{z} + \bar{x} \cdot \bar{z}$$

HW# Problem 1.4

$$\text{Hex } 0x0 = 0000 \\ \alpha 3 = 0b 0011$$

$0x3400$ ✓
 $0b0110$



$$011_2 = 6 - 7 = -1$$

$$\boxed{\text{Exponent} = -1}$$

Step 2

1. $\boxed{X \ X \ X \ X \ X \ X \ X}$

$$1.10000000000 \times 2^{-1}$$

$$(1.1 \times 2^{-1})$$

$$0.11_2$$

$$1 \times 2^{-1} + 1 \times 2^{-2}$$

$$(1 \times 2^{-1} + 1 \times 2^{-2})$$

$$0.5 + 0.25$$

$$2^{-1} = \frac{1}{2}; 2^{-2} = \frac{1}{2^2} = \frac{1}{4}$$

$$(52.6 \times 10^{-1}) \rightarrow 5.26$$

$$56_{10}$$

$$5 \times 10^{-1} + 6 \times 10^{-2}$$

2. Implementation of SOP Expression with NAND GATES.

$$f = \bar{x} + \bar{y} + xy$$

$$\text{Theorem 1: } \overline{xy} = \bar{x} + \bar{y}$$

$$\text{Theorem 2: } \overline{x+y} = \bar{x} \cdot \bar{y}$$

$$\text{AND-OR Gate Level: } f = \bar{x} \cdot \bar{y} + xy$$

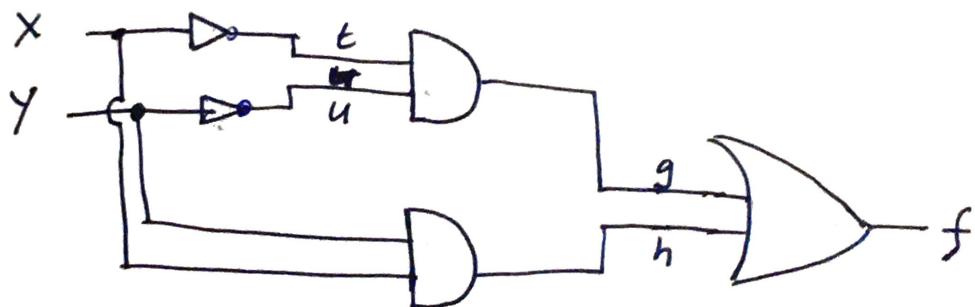
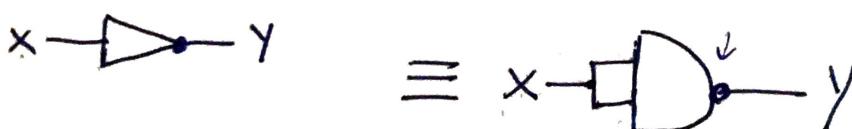


Fig 2.4

Step 1: Replace each NOT gate with its equivalent NAND gate by connecting the inputs of a two input NAND gate to the single input of the NOT gate.



See Fig 2.5

Step 2: Place two NOT gates on each end of the intermediate signals g and h . This will not change the output \Rightarrow e.g.: $\bar{g} = g$.
 A n AND-OR gate with two NOT gate added

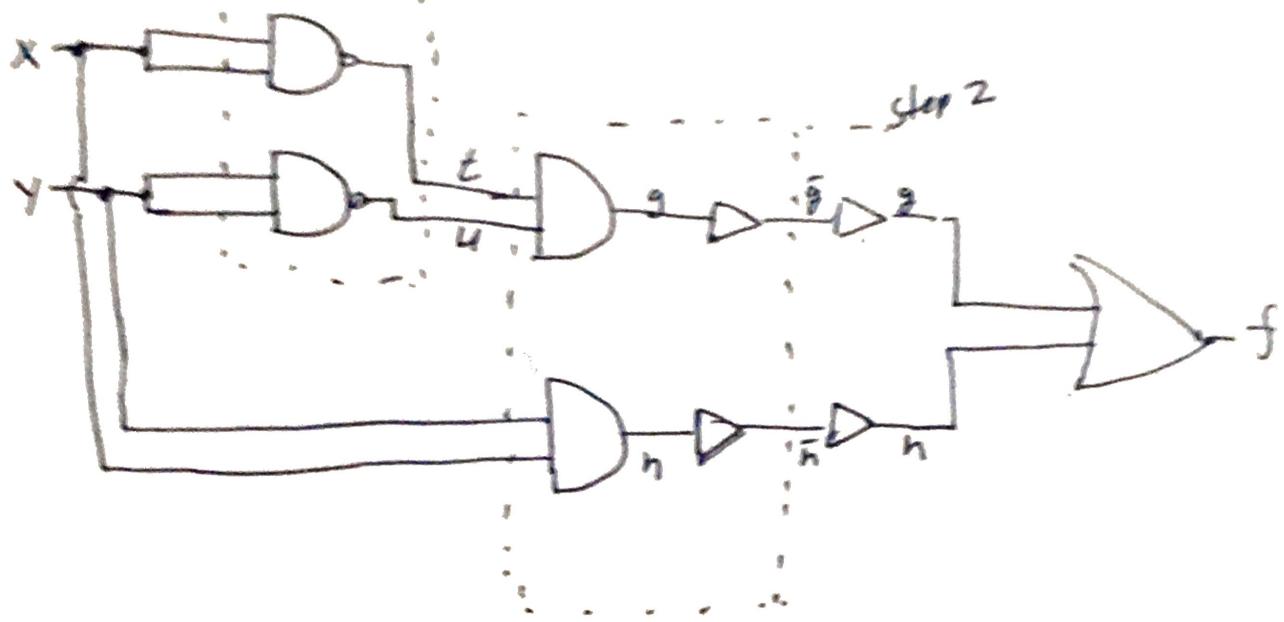


Fig 2.5

Step 3: Replace each AND-NOT gate with a NAND gate as shown below. AND gate is equivalent to an AND gate followed by a NOT gate
 AND-OR circuit with one bubble \Rightarrow Step 3

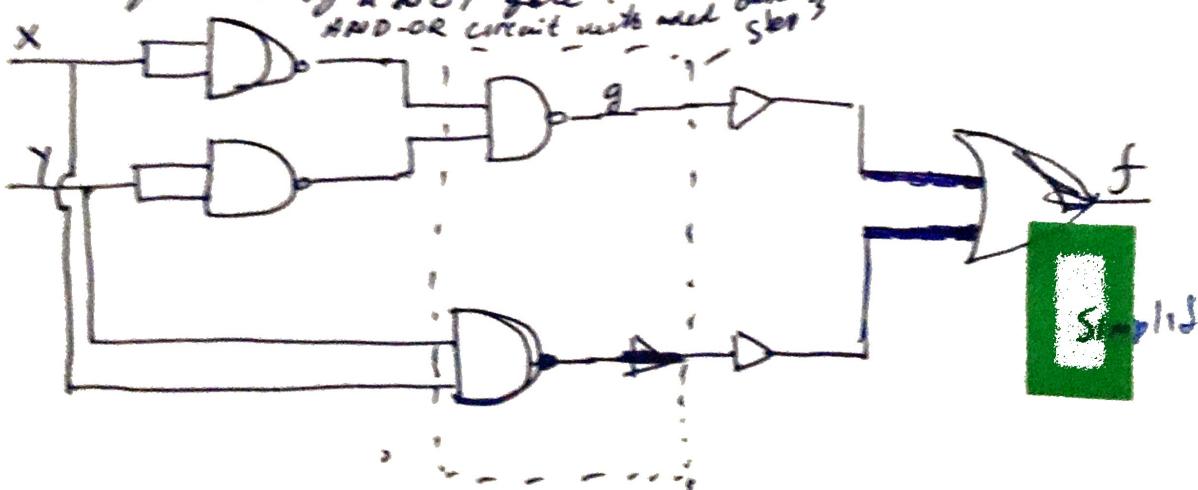


Fig 2.6

(The only gate not converted to NAND gate is the OR gate with inverted inputs.)

Recall DeMorgan theorem 1: $\overline{X \cdot Y} \equiv \overline{X} + \overline{Y}$

Step 4: Replicate OR gate with the inverted inputs with NAND gate as shown below

Add NAND gate only implementation of $f = \overline{x} \cdot \overline{y} + x \cdot y$

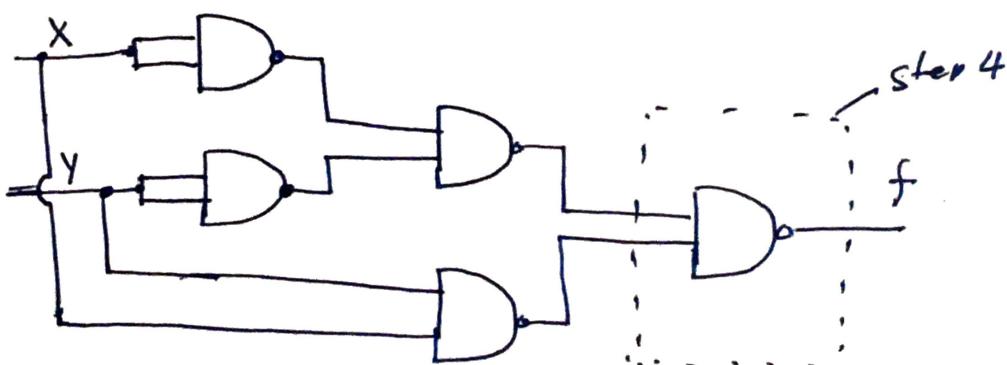
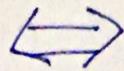
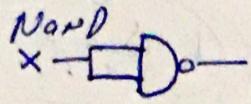


Fig 2.7

<u>Dec</u>	<u>Hex</u>	<u>Binary</u>
0	0	0b 0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
10	A	1010
11	B	1011
12	C	
13	D	
14	E	
15	F	

Configured



Inverter



X	X	Y
0	0	1
1	1	0

X	Y
0	1
1	0