# Representation for CSOP and CPOS forms:

| DE | $x$ | $y$ | $z$ | $F$ |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 |
| 2 | 0 | 1 | 0 | 0 |
| 3 | 0 | 1 | 1 | 1 |
| 4 | 1 | 0 | 0 | 1 |
| 5 | 1 | 0 | 1 | 1 |
| 6 | 1 | 1 | 0 | 0 |
| 7 | 1 | 1 | 1 | 0 |

$$F(x,y,z) = \overline{x}\cdot\overline{y}\cdot z + \bar{x}\cdot y\cdot z + x\cdot y\cdot \bar{z} + x\cdot y\cdot z \qquad CSOP$$

$$F(x,y,z) = m_1 + m_3 + m_4 + m_5 \qquad CSOP$$

$$F(x,y,z) = \sum m(1,3,4,5)$$

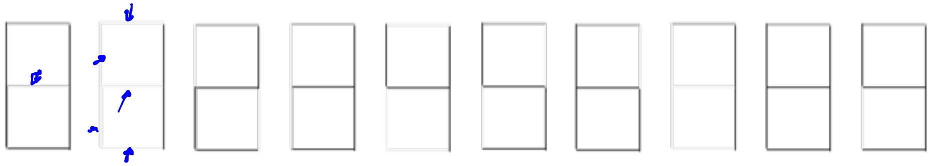
$$F(x,y,z) = (x+y+z)\cdot(x+\bar{y}+z)\cdot(\bar{x}+\bar{y}+z)\cdot(\bar{x}+\bar{y}+\bar{z}) \quad CPOS$$

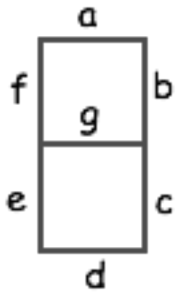
$$F(x,y,z) = M_0\cdot M_2\cdot M_6\cdot M_7 \qquad CPOS$$

Decimal equivalent

$$F(x,y,z) = \prod M(0,2,6,7)$$

# Example for Representation:






7_segment display

| DE | D | C | B | A | a | b | c | d | e | f | g |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 |
| 3 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 |
| 4 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| 5 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 |
| 6 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 |
| 7 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 8 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 9 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 |

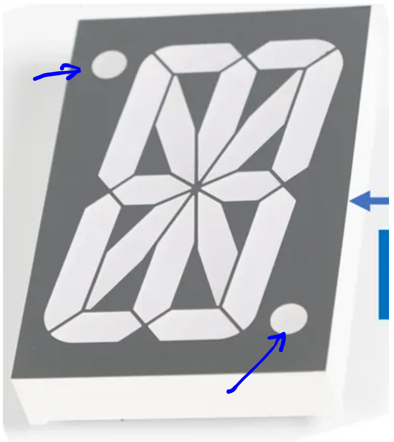
$$a(D,C,B,A) = \sum m(0,2,3,5,7,8,9)$$

$$= \prod M(1,4,6)$$

$$f(D,C,B,A) = \sum m(0,4,5,6,8,9)$$

$$= \prod M(1,2,3,7)$$

16 segment display



7 segment

## Converting SOP form to CSOP:

min term $x$ $\bar{x}$

- $f(x, y, z) = \bar{x} \cdot z + x \cdot \bar{y} = \left(\bar{x} \cdot z \cdot (y + \bar{y})\right) + \left(x \cdot \bar{y} \cdot (z + \bar{z})\right)$

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

- $f(a, b, c) = a + \bar{b}.c = a.\left(b + \bar{b}\right).(c + \bar{c}) + (a + \bar{a}).\bar{b}.c$

  $= a.b.c + a.b.\bar{c} + a.\bar{b}.c + a.\bar{b}.\bar{c} + \bar{a}.\bar{b}.c$

# Converting POS form to CPOS:

Axiom 2 $u + 0 = u$, $u.1 = u$

Axiom 3 $u.v = v.u$, $u + v = v + u$

Axiom 4 $u \cdot (v + w) = u \cdot v + u \cdot w$,

Axiom 5 $u + (v \cdot w) = (u + v) \cdot (u + w)$

$u \cdot \bar{u} = 0$, $u + \bar{u} = 1$

x+z+0.

$$f(x, y, z) = (x + z) \cdot (\bar{x} + \bar{y}) = (x + z + y \cdot \bar{y}) \cdot (\bar{x} + \bar{y} + z \cdot \bar{z})$$

$$= (x + y + z) \cdot (x + \bar{y} + z) \cdot (\bar{x} + \bar{y} + z) \cdot (\bar{x} + \bar{y} + \bar{z})$$

$$f(a, b, c, d) = a + d = a + d + b \cdot \bar{b} + c \cdot \bar{c} = \left(a + d + b \cdot \bar{b} + c\right) \cdot \left(a + d + b \cdot \bar{b} + \bar{c}\right)$$

$$= (a + d + b + c) \cdot \left(a + d + \bar{b} + c\right) \cdot \left(a + d + \bar{b} + \bar{c}\right) \cdot (a + d + b + \bar{c})$$

$$= (a + b + c + d) \cdot \left(a + \bar{b} + c + d\right) \cdot \left(a + \bar{b} + \bar{c} + d\right) \cdot (a + b + \bar{c} + d)$$

## Complements and Conversions:

- The complement of a logic function in its encoded form is very easy to do.
- To complement a function in either $\sum$ or $\prod$ form, replace the numerical codes in the list with all the codes that aren't there.
- Suppose we have three variables. Then the possible codes are the integers 0 through 7.
- Four variables would require codes from 0 to 15, and so on.
- The complement of $Z(C,B,A) = \sum m(1,3,5,6,7) = \bar{Z}(C,B,A) = \sum m(0,2,4)$

## Complements and Conversions:

- The complement of $X(C, B, A) = \prod(0,1,4)$ is $\bar{X}(G, H, J) = \prod(2, 3, 5, 6, 7)$
- Establishing these can be done by writing truth tables.

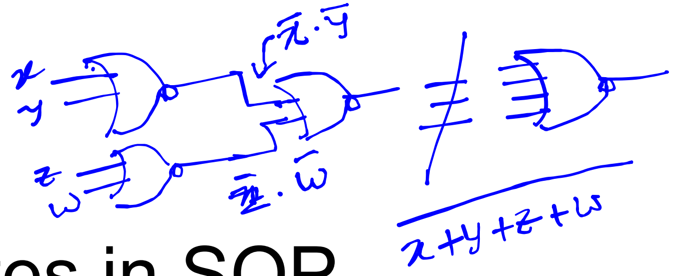
## Conversion from the product form to the sum form or vice versa:

- Change the product symbol to sum symbol, or sum symbnol to product symbol and replace the numerical codes with all the codes that aren't there.
- For example,

$$Z(C, B, A) = \sum(1, 3, 5, 6, 7) = \prod(0, 2, 4).$$

# Logic Synthesis:

Remember




- Implement function using NOT, AND and OR gates in SOP

$$f(D,C,B,A) = \sum m(0,4,5,6,8,9,10,11,12,13,14)$$

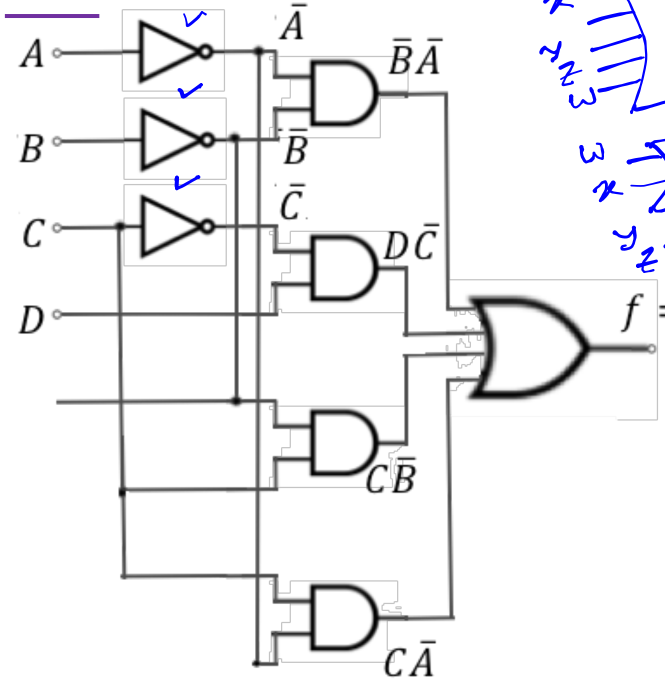
theorems & axioms.

$$f(D,C,B,A) = \bar{D}\bar{C}\bar{B}\bar{A} + \bar{D}C\bar{B}\bar{A} + \bar{D}C\bar{B}A + \bar{D}CB\bar{A} + D\bar{C}\bar{B}\bar{A} + D\bar{C}\bar{B}A + D\bar{C}B\bar{A} + D\bar{C}BA + DC\bar{B}\bar{A} + DC\bar{B}A + DCB\bar{A}$$

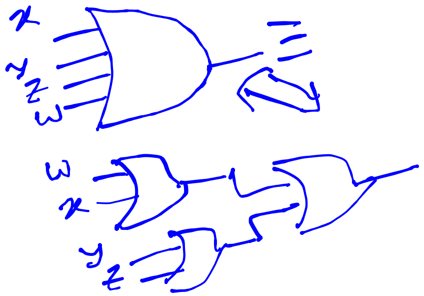
Karnaugh map.

$$= \bar{D}\bar{B}\bar{A} + \bar{D}C\bar{A} + D\bar{B}\bar{A} + C\bar{B}A + D\bar{C}A + DC\bar{A} + D\bar{C}\bar{A} + C\bar{B}\bar{A}$$

Reduces to $f = \bar{B}\bar{A} + D\bar{C} + C\bar{B} + C\bar{A}$

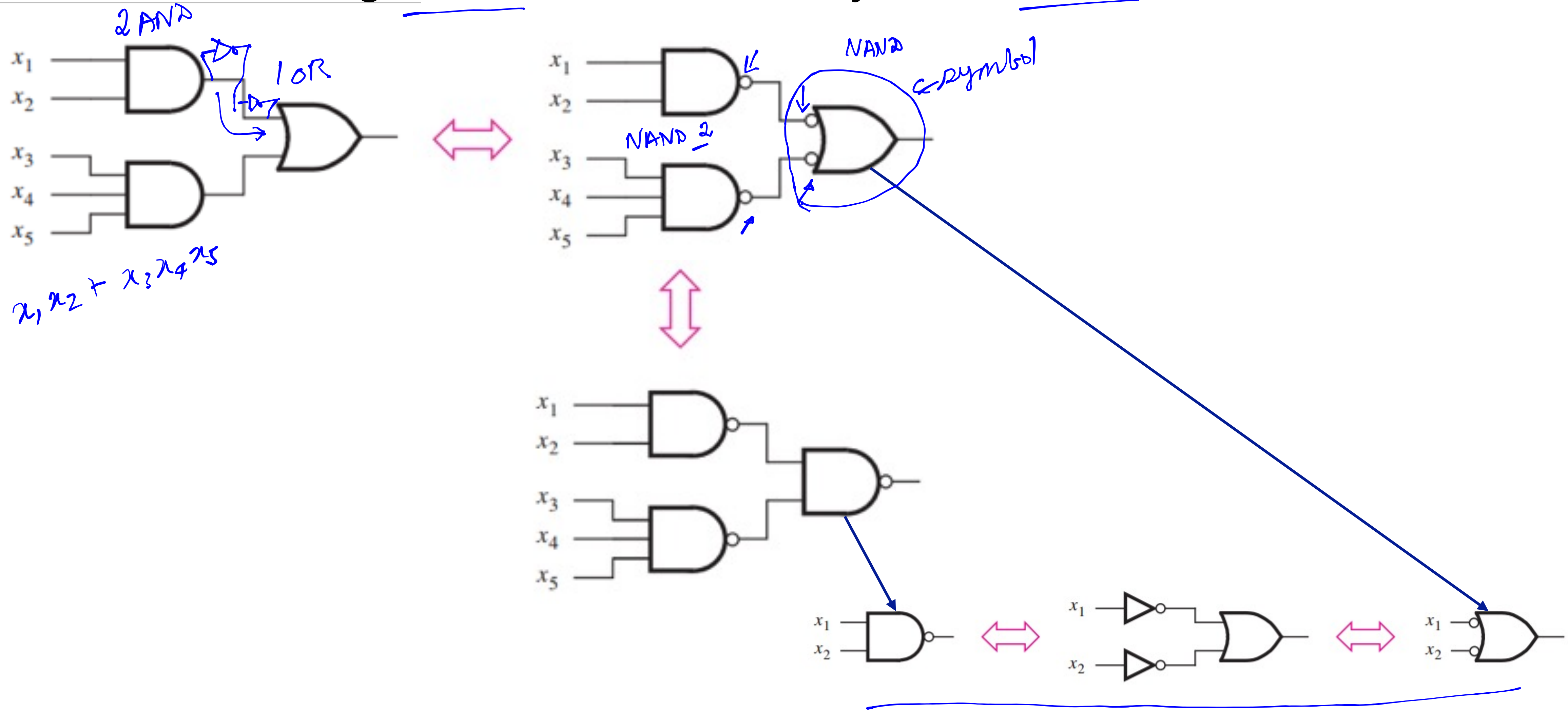



$f = \bar{B}\bar{A} + D\bar{C} + C\bar{B} + C\bar{A}$



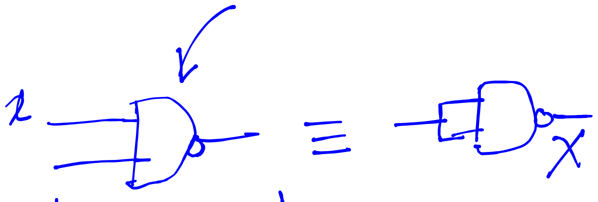
# Logic Synthesis:

## Realization Using NAND Gates Directly From SOP



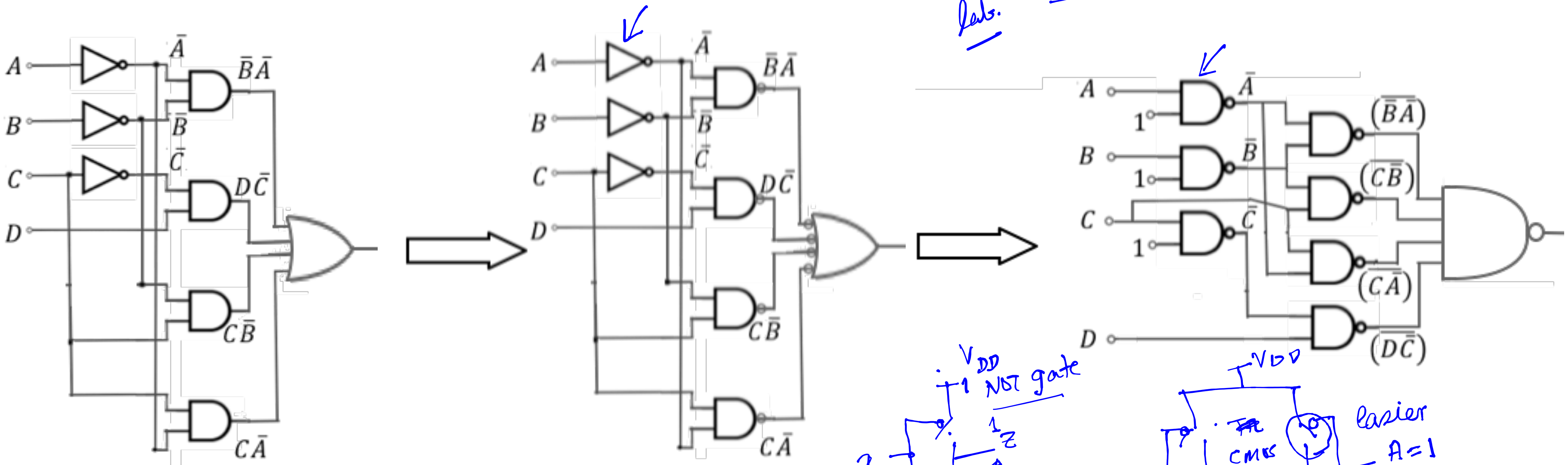

(a) $\overline{x_1 x_2} = \bar{x}_1 + \bar{x}_2$

$$f = \bar{B}\bar{A} + D\bar{C} + C\bar{B} + C\bar{A}$$



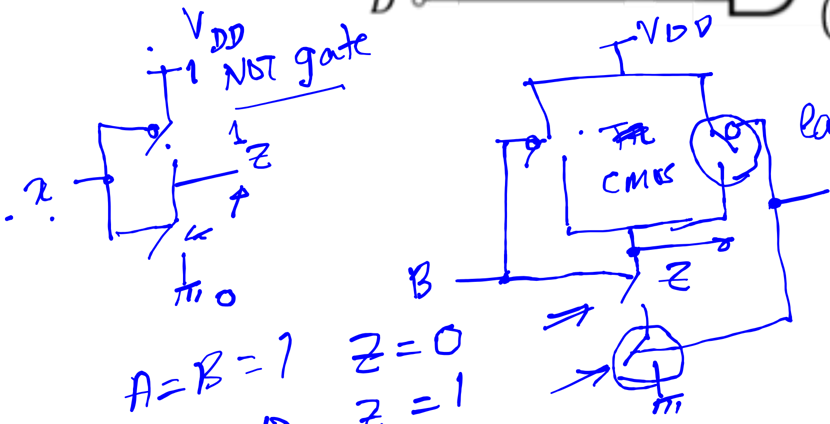

preferred

less.




| A | B | Z |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

NAND




easier

A = 1

NAND

A = B = 1 Z = 0

A or B = 0 Z = 1

# Logic Synthesis

$\bar{x}+\bar{y}$

$\overline{(\bar{x}+\bar{y})\cdot(\bar{z}+\bar{w})}$

$xy+zw \neq \overline{x\cdot y\cdot z\cdot w}$

$xy+zw$

$\bar{z}+\bar{w}$

Silicon

Wafer

Chip design

chip

mm²

- Implement function using only NAND gates

draw with basic gates

realize using only 2 input

$\bar{\bar{x}} = x$

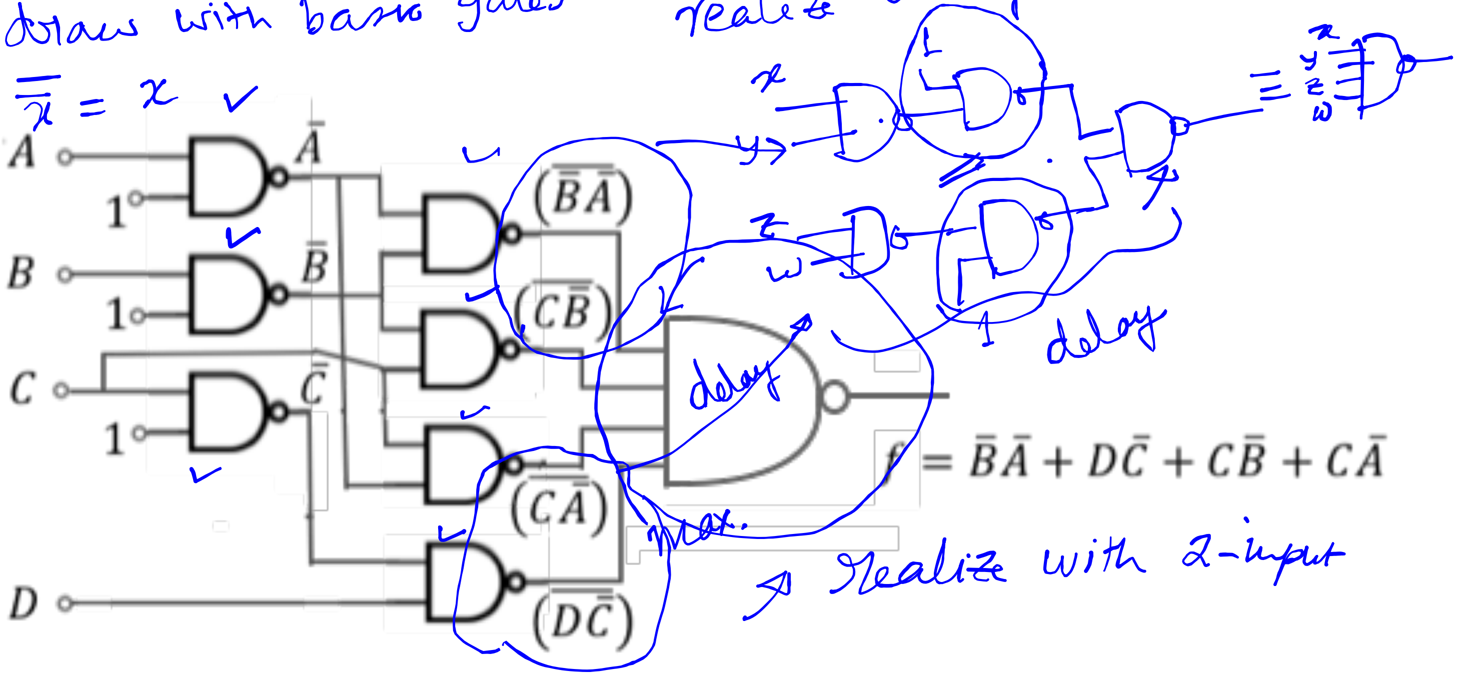
$$f = \bar{B}\bar{A} + D\bar{C} + C\bar{B} + C\bar{A}$$

$$= \overline{\overline{\bar{B}\bar{A} + D\bar{C} + C\bar{B} + C\bar{A}}}$$

$$= \overline{(\overline{\bar{B}\bar{A}})(\overline{D\bar{C}})(\overline{C\bar{B}})(\overline{C\bar{A}})}$$

NAND

NAND




realize with 2-input

# Logic Synthesis

AND OR

- Implement function using NOT, OR and AND gates in POS

$$f(D, C, B, A) = \prod M(1,2,3,7)$$

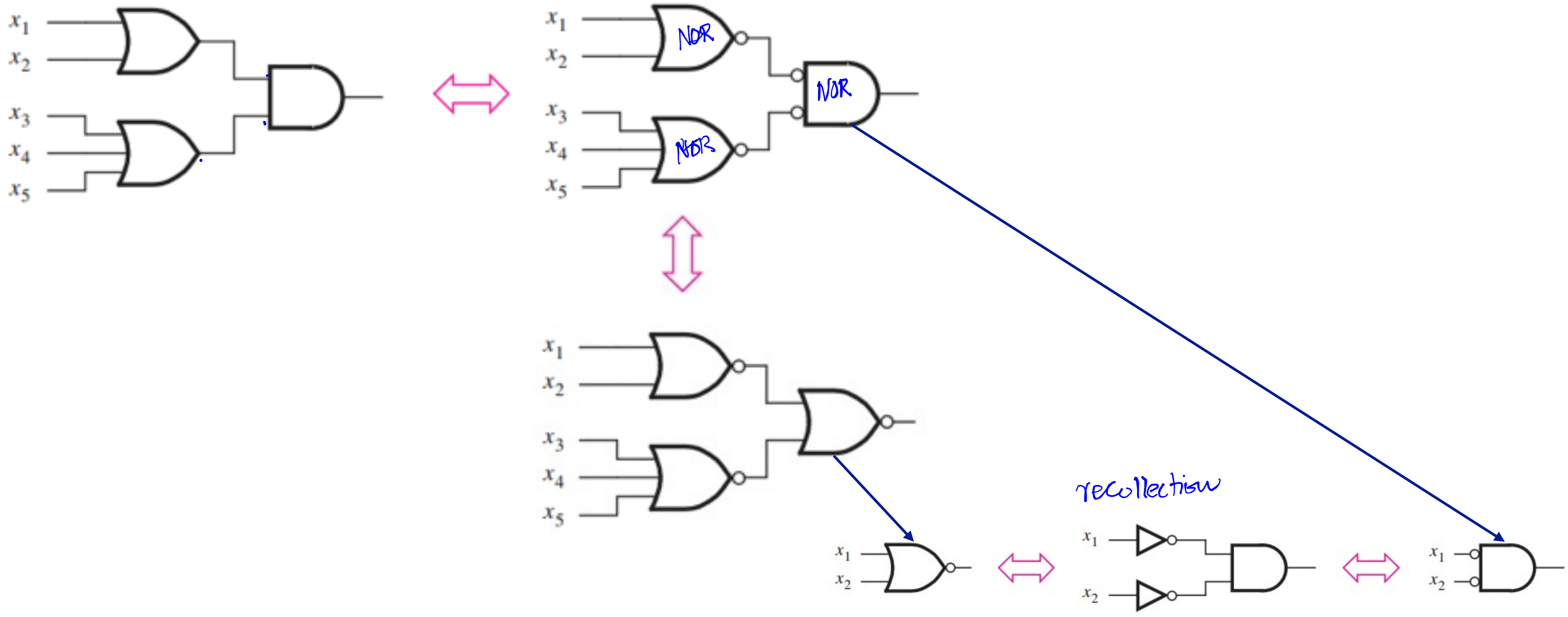
$$f = (\bar{D} + \bar{C} + \bar{B} + A)(\bar{D} + \bar{C} + B + \bar{A})(\bar{D} + \bar{C} + B + A)(\bar{D} + C + B + A)$$

This reduces to $f = (A + B)(C + \bar{B})(\bar{A} + C + D)$

Take over from here and carry out a process (like what we did for implementation using NOT, AND and OR gates) to implement using NOT, OR and AND gates

# Logic Synthesis:

## Realization Using only NOR Gates Directly From POS




(b) $\overline{x_1 + x_2} = \bar{x}_1\bar{x}_2$

# Logic Synthesis

- Implement function using only NOR gates

$$f = (A + B)(C + \bar{B})(\bar{A} + C + D)$$

$$f = \overline{\overline{(A + B)(C + \bar{B})(\bar{A} + C + D)}}$$

Take over from here and carry out a process (like what we did for implementation using only NAND gates) to implement using only NOR gates.

# Homeworks:

different from Tutorial problems.

Practice Problems

- Show that NAND and NOR are universal gates with the help of axioms, postulates and DeMorgan's law
- Implement the three functions given below using (i) Basic gates, (ii) only NOR gates and (iiI) only NAND gates. Use minimum number of gates

| A | B | C | F1 | F2 | F3 |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 1 | 1 | 1 |
| 0 | 1 | 0 | 1 | 1 | 0 |
| 0 | 1 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 | 0 | 0 |
| 1 | 1 | 1 | 0 | 0 | 1 |

# Tips:

- To implement any function using NOR gates, we need minimized POS. There are three ways to obtain minimized POS:

  ❖ Using postulated and axioms, obtain minimized SOP and convert it to POS using postulate 4.

  $F \rightarrow SOP \leftarrow$

  $\bar{F} \rightarrow POS$

  ❖ Instead of option 1, you may use complement version of given SOP if it has fewer number of terms. Using axioms and postulate, simplify this expression. Then, use DeMorgan's law to obtain minimized POS

  ❖ Convert the given expression to POS using Postulate 4, if it is in SOP form. Simplify it using axioms and postulates. (Simplifying expression in POS form is bit harder compared to SOP form)

  Karnaugh map

# Logical Statements -> Boolean Expressions -> Logical Networks

Abhishek will attend the DC class (A) if his friend Manav is attending the class (M) and Sheela is busy (S) or the topic being covered in class is important from examination point of view (T) and Manav is not attending the class (~M) or there is a quiz (Q). Write the expression for output which goes high when Abhishek attend the class.

machine

A ; M ; S Sheela busy T important for examination Q Quiz

Abhishek

Manav attends → 1

Early example

Right
Left
Flash

Abhishek attend if 1. Manav attend & Sheela is busy → M·S
or
2. Important for exam & Manav not attending → $T \cdot \bar{M}$
or
3. There is a Quiz → Q

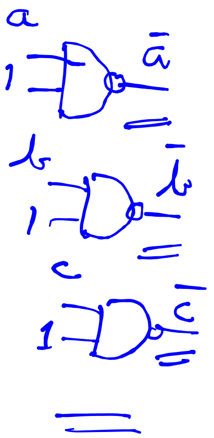
$A = M \cdot S + T \cdot \bar{M} + Q$

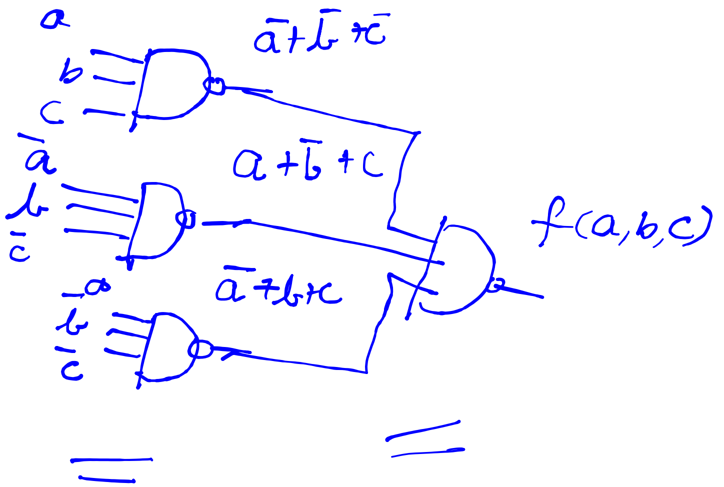
$A = M.S + T.\bar{M} + Q$

$$f(a,b,c) = abc + \bar{a}b\bar{c} + a\bar{b}\bar{c}$$

NAND gate

$$\overline{\overline{abc + \bar{a}b\bar{c} + a\bar{b}\bar{c}}} = \overline{\underbrace{(\bar{a}+\bar{b}+\bar{c})}_{X}\cdot\underbrace{(a+\bar{b}+c)}_{Y}\cdot\underbrace{(\bar{a}+b+c)}_{Z}} = \overline{X\cdot Y\cdot Z}$$




$$\overline{a\cdot b\cdot c\cdot d\cdots\cdot z}$$

$$\bar{a}+\bar{b}+\bar{c}+\bar{d}+\cdots+\bar{z}$$