Chapter 3 : Gate level minimization.

digital circuit cost minimize gate

The complexity of digital logic gates is directly proportional to the boolean expression from which the function is implemented.

complexity gates

$$F = x + x'y \quad \text{---} (1)$$

Minimization

$$= (x + x') \cdot (x + y) = x + y \quad \text{---} (2)$$

equivalent

less complex

algebra

We need to find the simplest boolean expression to describe the function.

- Minimize the function using algebra is awkward approach (hacks of specific rules to predict the next step).

So we need another way for minimization

Map Method (Minimization):

- Straight forward approach (طريقة مباشرة، تعتمد على الكل يفهم بيها الطريقة)
- Truth table is unique before Minimization and after minimization.

- Karnugh-map (K-map):
  - Diagram made of squares - each square represented one minterm.

*2 variable maps

2 variables $\Rightarrow 2^2$ minterms $\Rightarrow$ 4 minterms $\Rightarrow$ 4 squares.

$F_{(x,y)}$: $x'y'$ (m_0), $x'y$ (m_1), xy' (m_2), xy (m_3)

x \ y	0	1
0	$x'y'$ m_0	$x'y$ m_1
1	xy' m_2	xy m_3

$m_0, m_1 \Rightarrow$ جيران (بيختلفوا one variable)

$m_1, m_3 \Rightarrow$ جيران

$m_2, m_3 \Rightarrow$ جيران

$m_0, m_2 \Rightarrow$ جيران

$m_1, m_2 \Rightarrow$ مش جيران

ex) Minimize the following function.

$F_{(x,y)} = x'y + xy' + xy$ (m1, m2, m3)

$= x'y + x(y' + y)$

$= x'y + x$

$x + x'y = (x + x') \cdot (x + y)$

$= x + y$

الطريقة: (algebra)

K-map ⇓

x \ y	0	1
0	0	1
1	1	1

x صفره يس x & y

الربع بعطينا (y)

صفران

يعني نقطه zero

صفران → x

* ما بقدر اجمع ٣ "١١١" مع بعض.

بس يا بجمع العاده يا ٢ "١١" ، يا ٤ ، يا ٨ .

$2^0 \quad 2^1 \quad 2^2 \quad 2^3 \quad 2^4 \; \ldots \; 2^n$ ← مقدار الواحد الي بقدر اجمعه .

$F = x + y$ → minterm 8؟

Ex $F = x'y' + x'y + xy'$ (m0, m1, m2)

$= x'(y' + y) + xy'$

$x' + xy'$

$= (x' + x) \cdot (x' + y') = x' + y'$

x \ y	0	1
0	1	1
1	1	0

$\bar{x}$ (أفقي) , y'

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

• Ex. $F(x,y) = x'y + xy'$

m_1 , m_2

x \ y	0	1
0		1
1	1	

$x'y$, xy'

مش جيران

$F = x'y + xy'$

(يبقى كما هو، لا يوجد تبسيط له).

• Ex $F = A'B' + A'B + AB' + AB$

$A'(B' + B) + A(B' + B)$

$F = A' + A = 1$

جيران

A \ B	0	1
0	1	1
1	1	1

جيران جيران جيران

→ الجواب "1"

عملنا circle لها كلها

- بنز نطبق عليه كيف جيران ؟

بس بدوا بحطهم [2 variable 01 10 لو حطيت]

Three variable maps.

3 variables ⇒ 2^3 combination = 8 = 8 squares.

(جيران)

لازم الي جنب بعض جيران

$F_{(x,y,z)}$

x \ yz	00	01	11	10
0	$x'y'z'$ m_0	$x'y'z$ m_1	$x'yz$ m_3	$x'yz'$ m_2
1	$xy'z'$ m_4	$xy'z$ m_5	xyz m_7	xyz' m_6

$m_0 \rightarrow m_2$ جيران

$m_4 \rightarrow m_6$ جيران

(y تغيرت z ... ثابتة)

Minimize using k-map

$F_{(x,y,z)} = \Sigma(0,1,3,7)$

$= m_0 + m_1 + m_3 + m_7$

x \ yz	00	01	11	10
0	1	1	1	
1			1	

$x'y'$

$y \cdot z$

x	y	z	F
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

$F = x'y' + y \cdot z$

- Minimize using K-map

$F_{(x,y,z)} = \Sigma(0,1,2,3)$

x \ yz	00	01	11	10
0	1	1	1	1
1				

→ جيران بعض
بالـx ثابت على الصفر
y ، z تغيروا

$= \bar{X}$

- $F_{(A,B,C)} = \Sigma\, 0,1,2,3,4,6$

A \ BC	00	01	11	10
0	1	1	1	1
1	1			1

→ $\bar{A}$

جيران

صفين تغيرت

$\bar{C}$ له

0 1
1 1

$= \bar{A} + \bar{C}$ ✓

Ex $F_{(A,B,C)} = m_8 + m_5 + m_6 + m_7$

A \ BC	00	01	11	10
0			1	
1		1	1	1

→ AB

A·C B·C $= AC + BC + AB$

A \ CB	00	01	11	10
0	1	1		
1	1	1		

$F = \bar{C}$

- Four variable maps.

4 variables ⇒ 2^4 minterms.

$F(A,B,C,D)$

AB \ CD	00	01	11	10
00	$\bar{A}\bar{B}\bar{C}\bar{D}$ m_0	$\bar{A}\bar{B}\bar{C}D$ m_1	$\bar{A}\bar{B}CD$ m_3	$\bar{A}\bar{B}C\bar{D}$ m_2
01	$\bar{A}B\bar{C}\bar{D}$ m_4	$\bar{A}BCD$ m_5	$\bar{A}BCD$ m_7	$\bar{A}BC\bar{D}$ m_6
11	$AB\bar{C}\bar{D}$ m_{12}	$AB\bar{C}D$ m_{13}	$ABCD$ m_{15}	$ABC\bar{D}$ m_{14}
10	$A\bar{B}\bar{C}\bar{D}$ m_8	$A\bar{B}\bar{C}D$ m_9	$A\bar{B}CD$ m_{11}	$A\bar{B}C\bar{D}$ m_{10}

~~m_{12} m_{13} m_4~~
~~m_8 m_9 m_{11} m_{10}~~

$F_{(A,B,C,D)} = \Sigma\, 0, 2, 5, 7, 8, 10, 13, 15.$

AB \ CD	00	01	11	10
00	1			1
01		1	1	
11		1	1	1
10	1			1

جيران

BD

(الاربعة في وسط)

$B'D'$

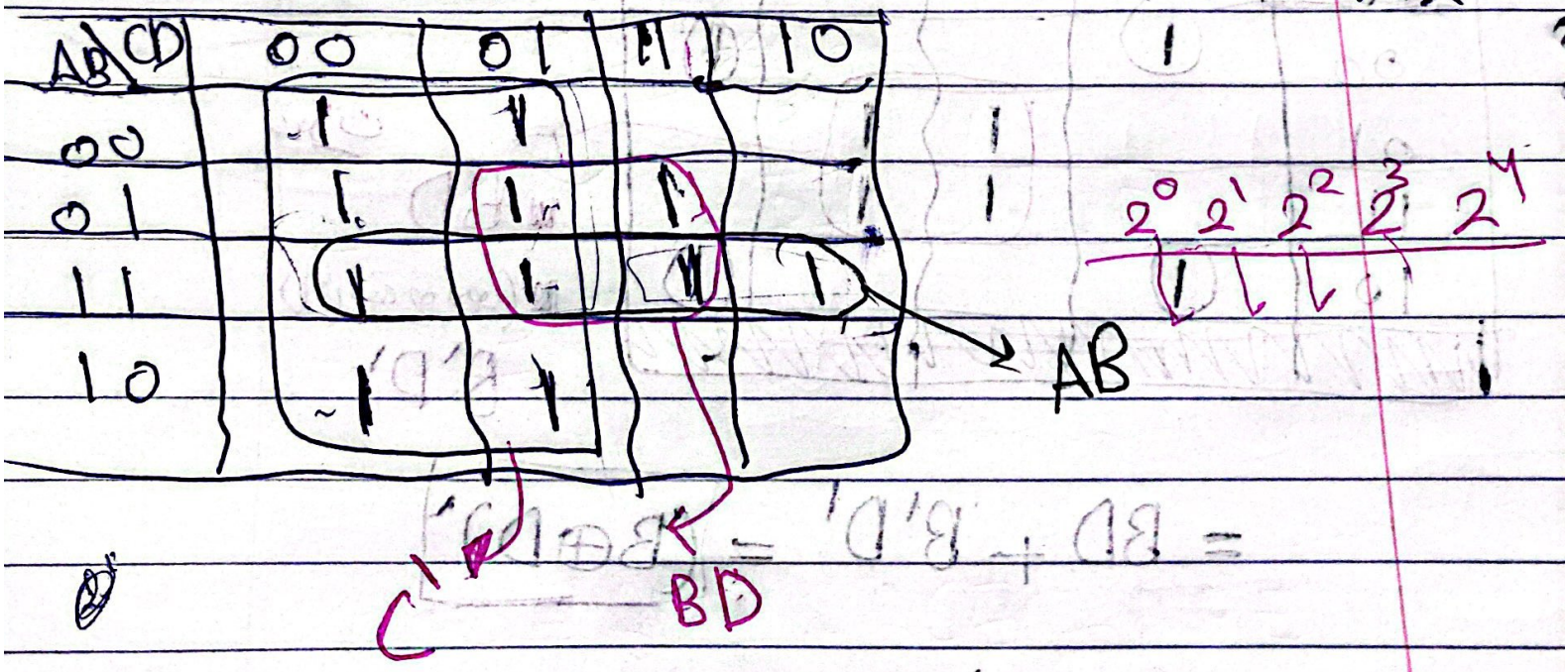
$= BD + B'D' = \overline{(B \oplus D)}$

Note ⇒ $F = \overline{(A \oplus B)} = A'B' + AB$

$$F = m_1 + m_2 + m_4 + m_7$$
$$= \Sigma 1, 2, 4, 7$$
$$= A \oplus B \oplus C$$

A	B	C	F
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

$$F_{(A,B,C,D)} = \Sigma 0, 1, 4, 5, 7, 8, 9, 13, 14, 15, 12$$



$$2^0\ 2^1\ 2^2\ 2^3\ 2^4$$

$$= AB + BD + \bar{C}$$

5 variable maps.

$F_{(A,B,C,D,E)} = 2^5 = 32$ minterms.

Note: $F_{(A,B,C)} = \Sigma\, 0, 2, 5, 7$

A	B	C	F
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

بنقسم ال truth table حسب A.

A = 0

B	C	F
0	0	1
0	1	0
1	0	1
1	1	0

A = 1

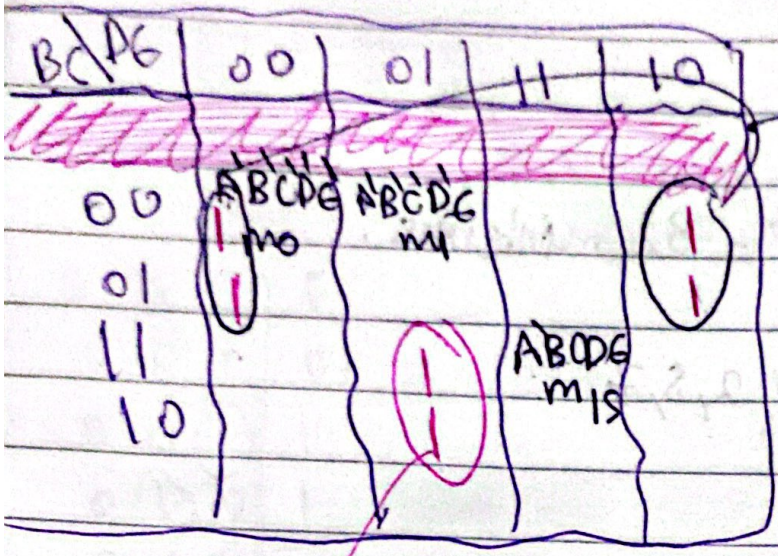
B	C	F
0	0	0
0	1	1
1	0	0
1	1	1

أو بقسم ال truth table لأربع أقسام A, B.

$F_{(A,B,C,D,E)} = \Sigma\, 0, 2, 4, 6, 9, 13, 21, 23, 25, 29, 31$

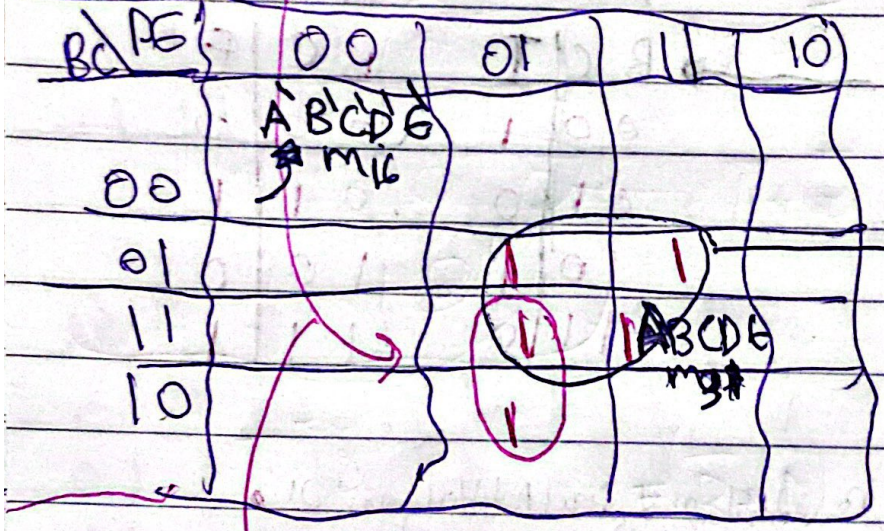
32 → 16 (A=0), 16 (A=1)

A = 0



* كل اصيص بنطبقوا فوق
بعض جيران

A = 1



BD'E

$$A'B'E' = BD'E + ACE + A'B'E'$$

6 variable maps

(64 squares)

$F_{(A,B,C,D,E,G)} = \Sigma 0, 16, 32, 48 = \bar{C}\bar{D}\bar{E}\bar{G}$

AB = 00 AB = 01 AB = 11 AB = 10

AB = 00

CD \ EG	00	01	11	10
00	1			
01				
11				
10				

AB = 11

CD \ EG	00	01	11	10
00	1			
01				
11				
10				

AB = 01

CD \ EG	00	01	11	10
00	1			
01				
11				
10				

AB = 10

CD \ EG	00	01	11	10
00	1			
01				
11				
10				

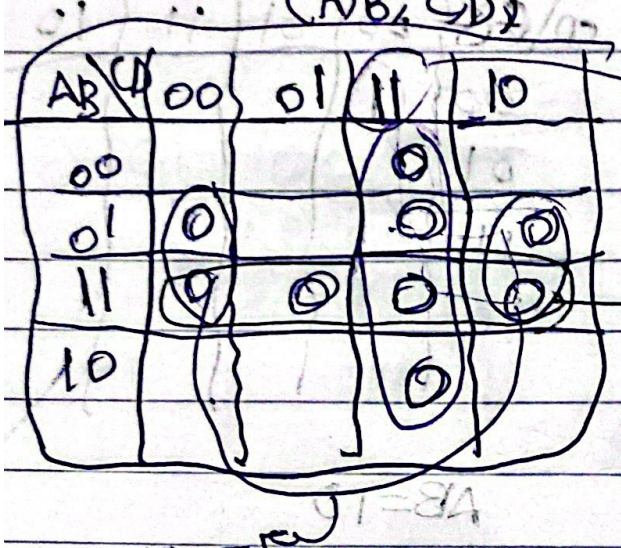
☑ الأربع حيارات

* Q-M method, Tabulation, Madawsky.

Product of Sum

simplify the following function.

$$F_{(A,B,C,D)} = \Pi(3,4,6,7,11,12,13,14,15)$$

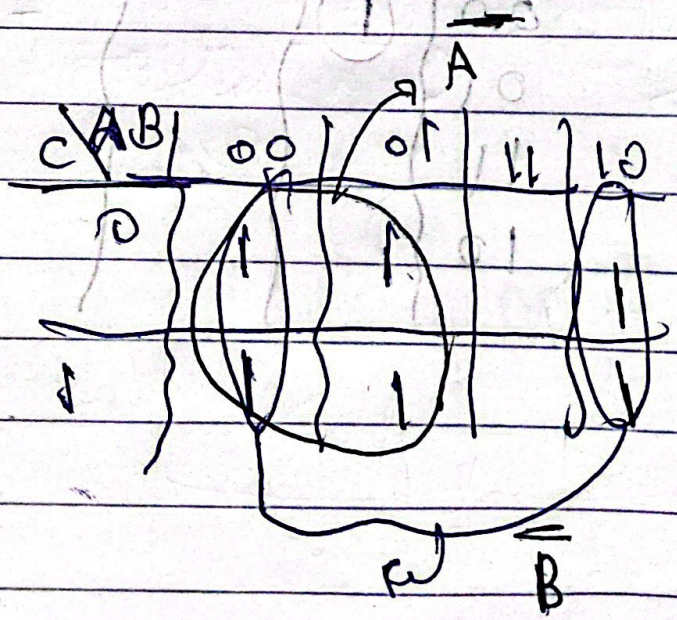


$\bar{C}+D$

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

$\bar{B}+D$

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



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

$CD + A\bar{D}$

- Don't care condition (X)

function that have unspecified output for specific inputs.

ex. simplify (K-map) the boolean function.

$F(A,B,C,D) = \Sigma 1, 3, 7, 11, 15$ that has don't care

$d(A,B,C,D) = \Sigma 0, 2, 5$

A	B	C	D	F
0	0	0	0	X
0	0	0	1	1
0	0	1	0	X
0	0	1	1	1
0	1	0	0	0
0	1	0	1	X
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1

AB \ CD	00	01	11	10
00	X	1	1	X
01		X	1	
11			1	
10			1	

$F = CD + A\bar{D}$

$F = CD + \bar{A}\bar{B}$

essential term

not essential term

Prime Implicants (PIs)

$F_{(A,B,C,D)} = \sum 0, 2, 5, 7, 8, 10, 13, 15$

AB \ CD	00	01	11	10
00	1			1
01		1	1	
11		1	1	
10	1			1

5, 7, 13, 15 → $BD = m_5 + m_7 + m_{13} + m_{15}$

term

Prime Implicant (essential)

اذا فيه واحد من الواحدات لا يغطيه الا هذه الطريقة ← essential

$\bar{B}\bar{D}$

essential

$= BD + \bar{B}\bar{D}$

$F_{(A,B,C,D)} = \sum 0, 2, 5, 6, 7, 8, 10, 13, 14, 15$

AB \ CD	00	01	11	10
00	1			1
01		1	1	1
10		1	1	1
11	1			1

$\bar{B}\bar{D}$ / E

هاد الشي غلط

$B\bar{C}$

BD / E

$= \overline{BD} + \overline{B}\overline{D} + C\overline{D}$

$= BD + \overline{B}\overline{D} + BC$

e ← not e

$F_{(A,B,C,D)} = \Sigma\, 0, 2, 5, 7, 8, 10, 13, 15, 6, 14, 9, 11$

(e) BD

AB \ CD	00	01	11	10
00	1			1
01		1	1	1
11		1	1	1
10	1	1	1	1

(e) $\overline{B}\overline{D}$

$F = BD + \overline{B}\overline{D} + C\overline{D} + A\overline{B}$

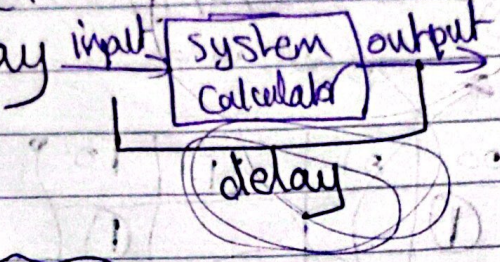
$F = BD + \overline{B}\overline{D} + C\overline{D} + AD$

$F = BD + \overline{B}\overline{D} + BC + A\overline{B}$

$F = BD + \overline{B}\overline{D} + BC + AD$

NAND & NOR Implementation (ابناء السيركت)

All gates consist of transistors.
NAND/NOR has fewer # of transistors.
1) Design using NAND/NOR cheaper.
2) size.
3) delay

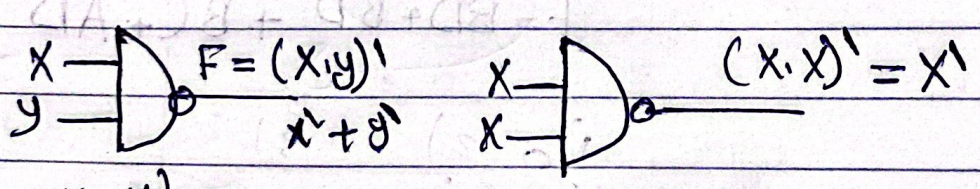


* NAND/NOR (universal gates)

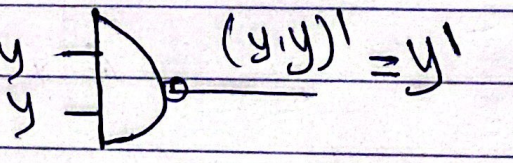
AND OR Not

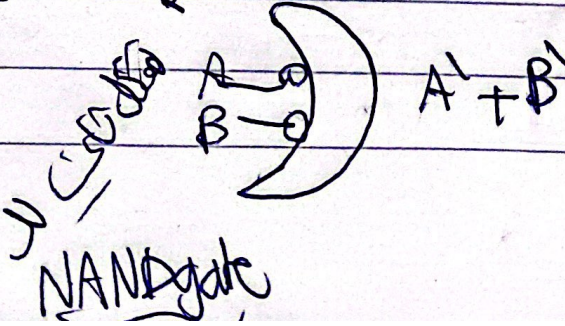
AND OR Nob

- NAND

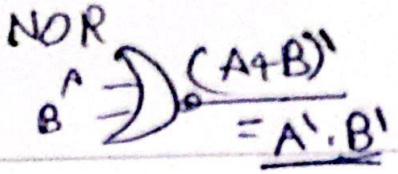


X	y	F
0	0	1
0	1	1
1	0	1
1	1	0

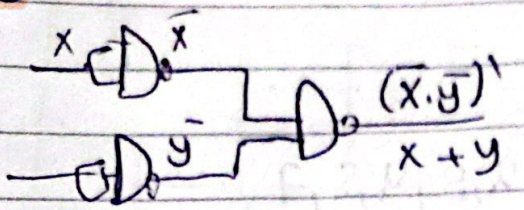


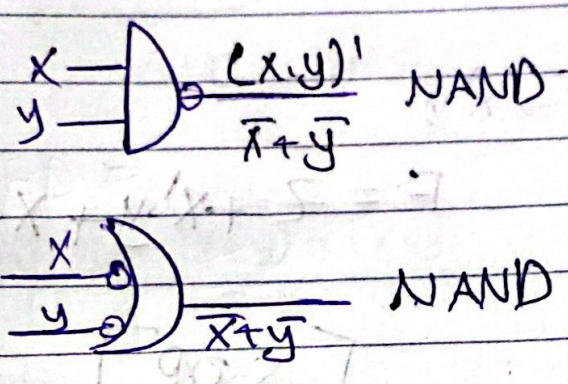


NOR



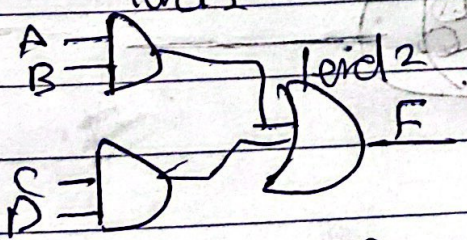
OR





Two level Implementation

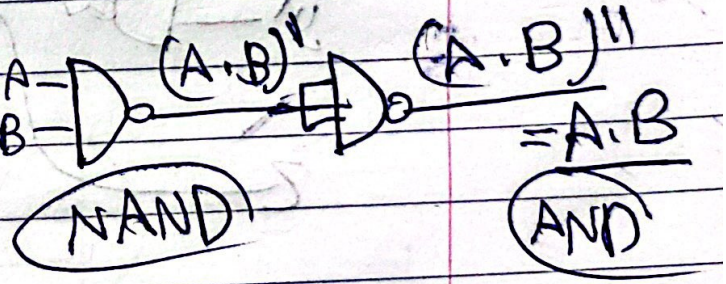
F = AB + CD



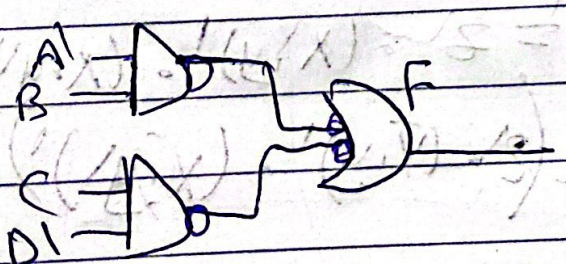
AND – OR

① NAND/Inverter

NAND → AND

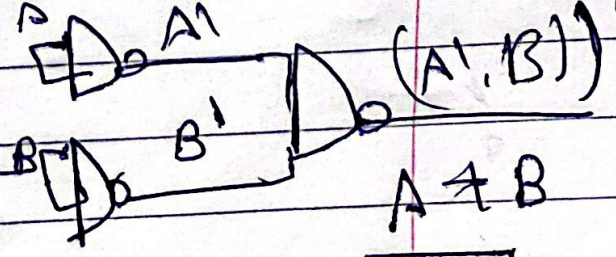


F = AB' + CD'



NAND – NAND

NAND → OR



1) AND_OR
2) NAND_NAND

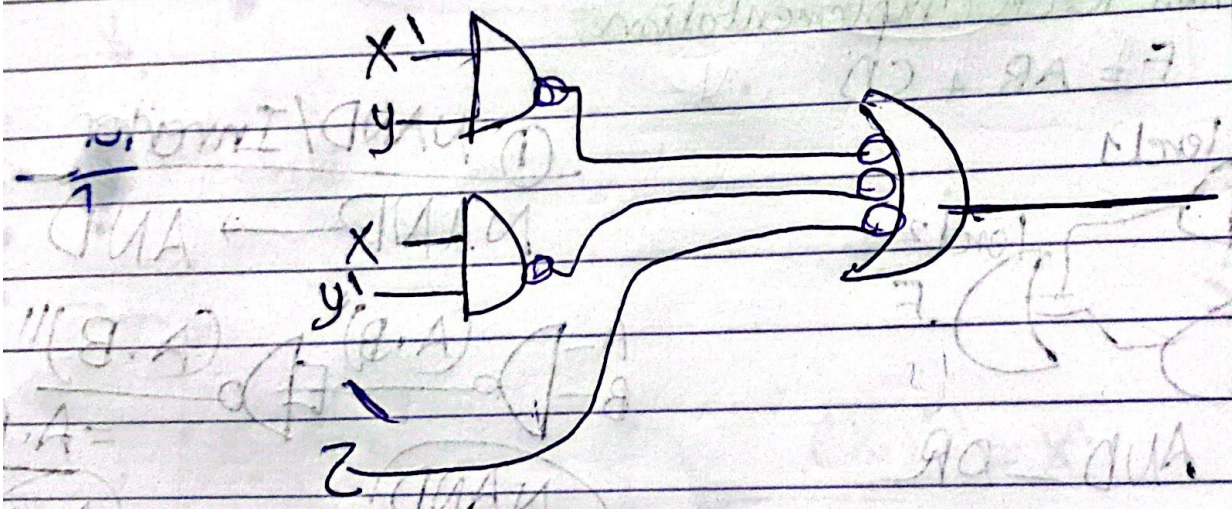
$F(x,y,z) = \Sigma 1,2,3,4,5,7$

yz \ x	00	01	11	10
0		1	1	1 → $x'y$
1	1	1	1	

x,y'

$F = z + x'y + xy'$

[SOP]

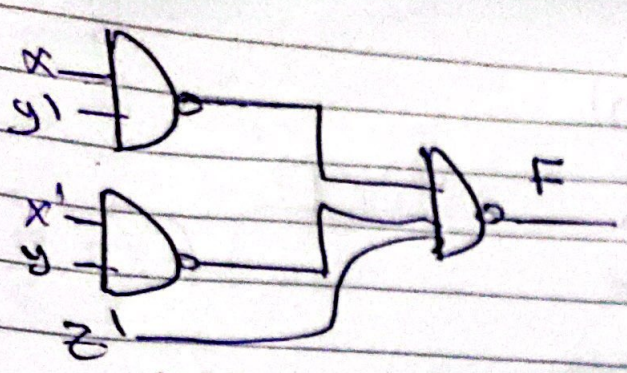


طريقة NAND-NOR

$F' = (z + x'y + xy')'$

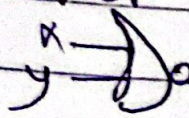
$F' = z' \cdot (x'y)' \cdot (xy')$

$F = (z' \cdot (x'y)' \cdot (xy'))'$



NAND _ NAND

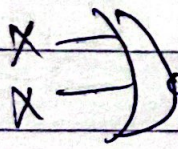
Nor



$F = \overline{(x+y)} = (x+y)' = x' \cdot y'$

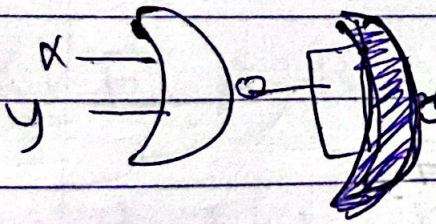
x	y	F
0	0	1
0	1	0
1	0	0
1	1	0

Not / Nor

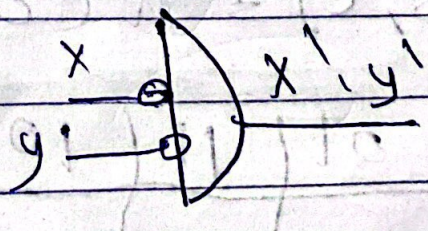


$(x+x)' = x'$

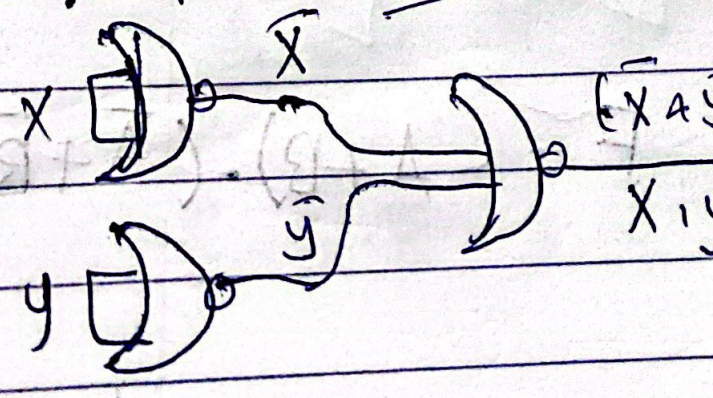
Inverter ✓



✓

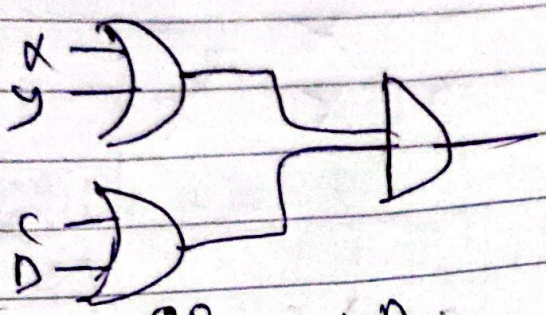


AND

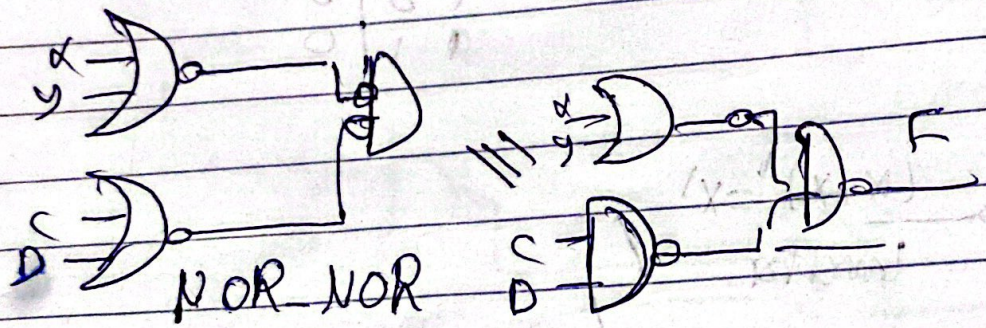


$F = (X+y) \cdot (C+D)$

OR — AND



OR - AND

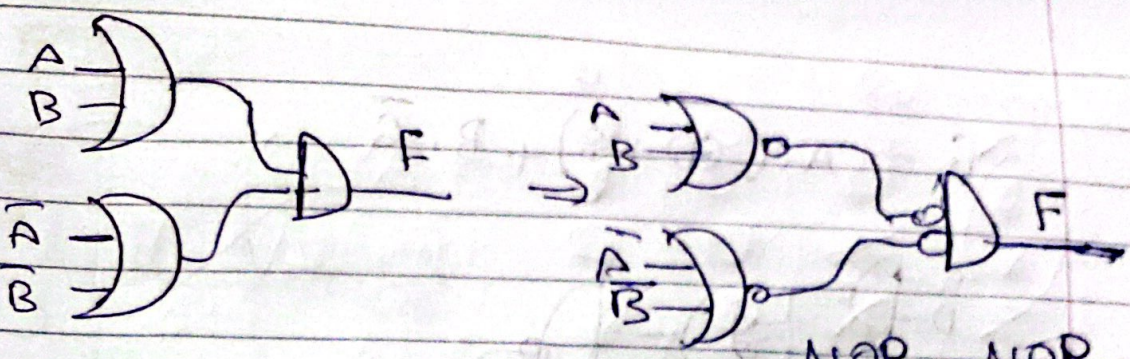


NOR NOR

$F(A,B,C) = \Pi(0,1,6,7)$

A \ BC	00	01	11	10
0	0	0		
1			0	0

$F = (A+B) \cdot (\bar{A}+\bar{B})$

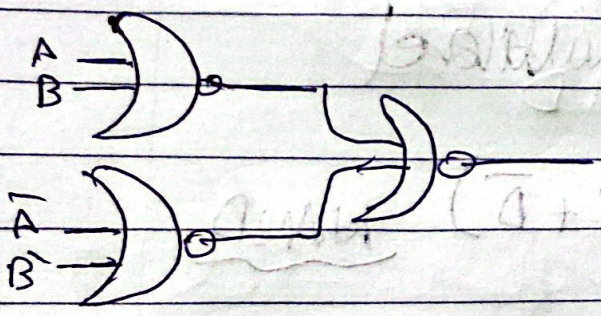


NOR - NOR

$$F' = [(A+B).(\bar{A}+\bar{B})]'$$

$$= (A+B)' + (\bar{A}+B')'$$

$$F'' = ((A+B)' + (\bar{A}+B)'')'$$



OR - AND

Ex. $F(A,B,C) = \Pi(0,1,6,7)$

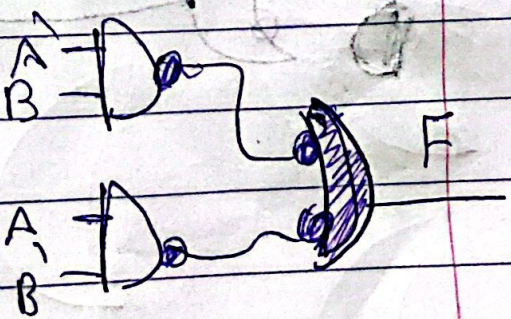
NAND - NAND

A \ BC	00	01	11	10
0			1	1
1	1	1		

$\Sigma(2,3,4,5)$

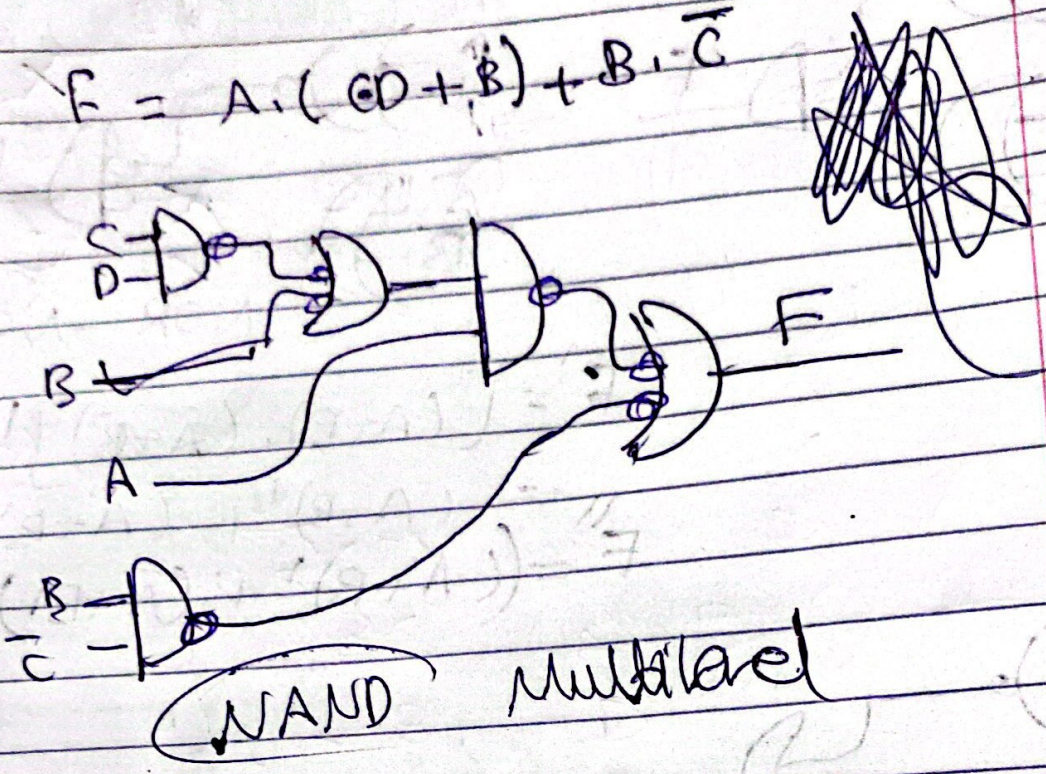
AND - OR

$$F = \bar{A}.B + A.B'$$



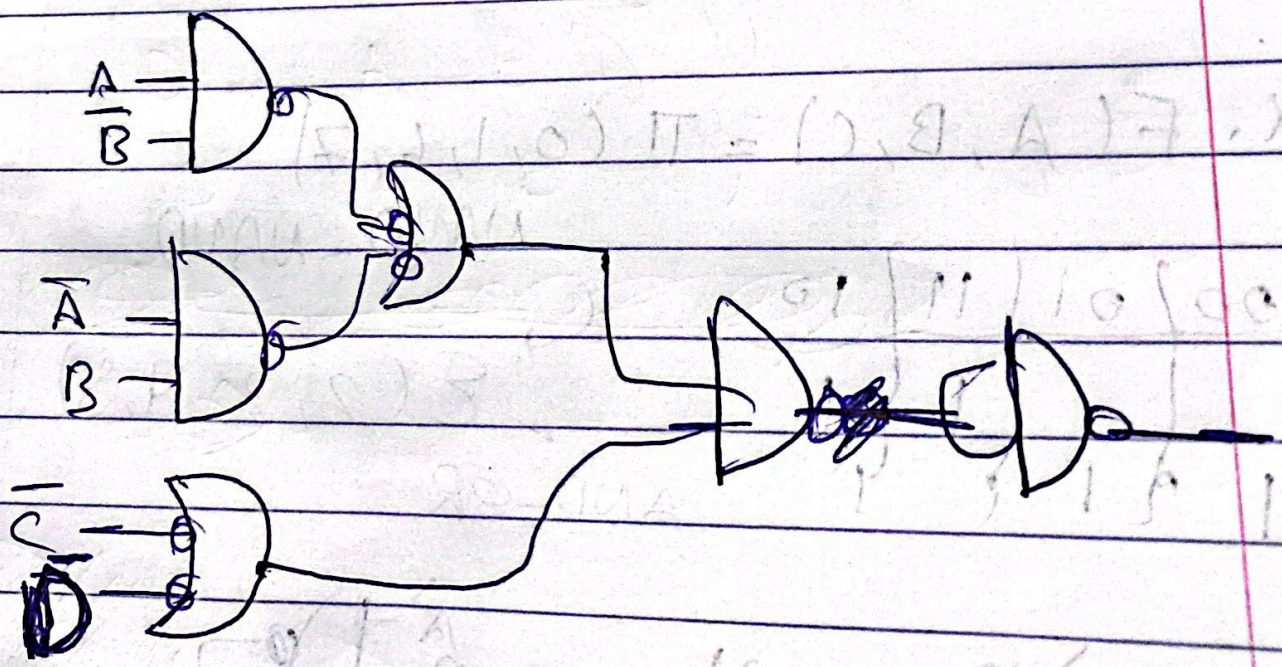
NAND - NAND

$F = A \cdot (\overline{C}D + \overline{B}) + B \cdot \overline{C}$



NAND Multilevel

$F = (A\overline{B} + \overline{A}B) \cdot (C + \overline{D})$ NAND



$F(A,B,C) = \Pi(0, 1, 3, 7)$ // NOR-NOR

A \ BC	00	01	11	10
0	0	0	0	
1			0	

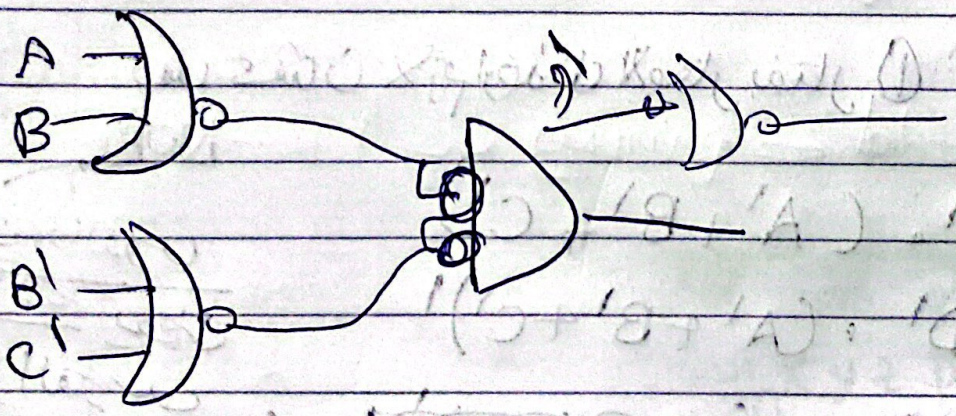
~~$AB + B'C'$~~

$(A+B) \cdot (B'+C')$

$F = (A+B) \cdot (B'+C')$

$F' = (A+B)' + (B'+C')'$

$F'' = [(A+B)' + (B'+C')']'$



$F = \sum_{(A,B,C,D)}(2, 4, 6, 10, 12)$

$d = \sum_{(A,B,C,D)}(0, 8, 9, 13)$

NOR-NOR

(+) . (+) ( )

NOR – NOR

AB \ CD	00	01	11	10
00	X	0	0	1
01	1	0	0	1
11	1	X	0	0
10	X	X	0	1

→ $A'CD'$

→ $(A'+B'+C')$

$AB'CD'$

$BC'D'$

1) D'

2) ~~$A'CD'$ $(A'+C+D')$~~

3) ~~$A'BC$ $(A'+B+C)$~~

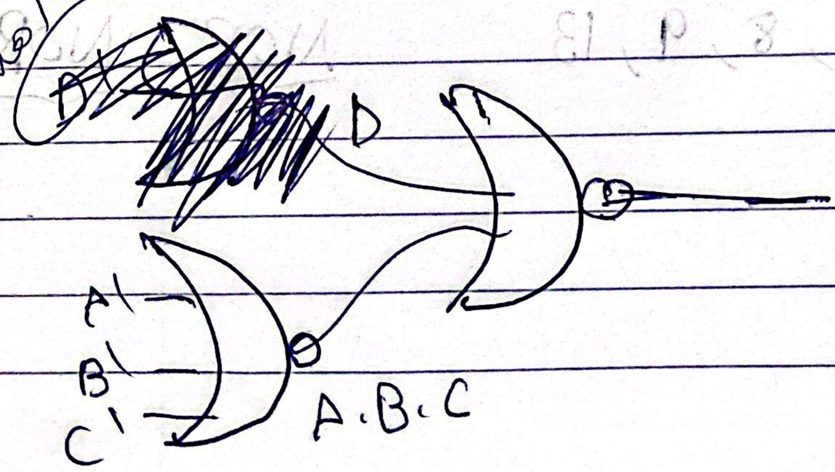
4) ~~$(A+B+C+D)$~~

يس مش لازم اعمل كل العطار بيختار ①

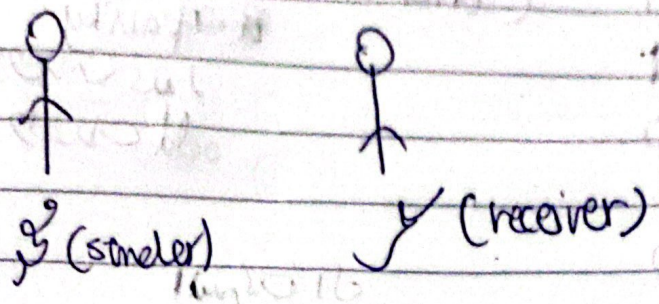
$F = D' \cdot (A'+B'+C')$

$F' = (D' \cdot (A'+B'+C'))'$

$F'' = (D + (A'+B'+C')')'$



Parity Generator



data
Image
Video
Audio

receiver checks if message is correct
ok, if not correct (sender please again) correction

(single error)

parity : extra bit added to the message to check if the received message is correct?

Ex message x, y, z → sender → 3 bit e.g. (message) → (circuit) sender add extra bit, (P)

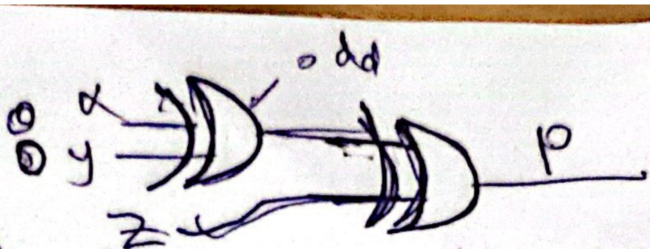
x	y	z	P (even)
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

even x y z p

parity
odd
even

even [illegible]

Receiver



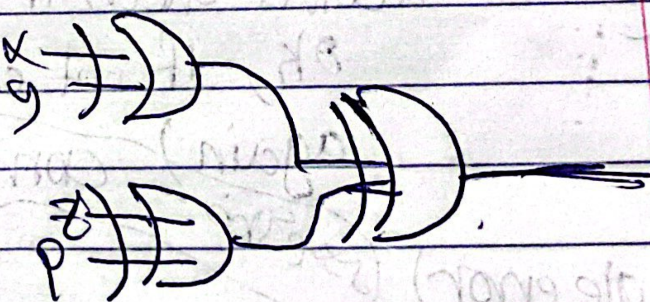
x	y	z	p	check
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	1
1	1	1	1	0

(no error)

(error)

even عند ال parity بصير عندي ال check odd

السركت ال check عند



odd. ← parity كان اذا ← واجب