CH.3 | GATE LEVEL MINIMIZATION

Rawan Alfares

Boolean Function Minimization

* the truth table is unique
* However the algebric expression is not unique.
* Boolean function can be simplified by algebric manipulation.
  However, algebric manipulation depends on experience.
* Also, it doesn't guarantee that the simplified Boolean expression is minimal.
+ Minimize the number of litrals in the boolean expression

أقل استخدام transistors في كتابة الـ Circuit.

أقل عدد من المتغيرات
⇓
أقل تكلفة

Karnaugh Map

$n \rightarrow$ # of variables.

⇒ K-Map
⇒ diagram made up of squars. → squars $= 2^n$
⇒ each square represents a minterm.
⇐ كل مربع والآخر بيختلفوا فقط بـ one variable
⇒ Simplified SOP expression → AND-OR Circuit.
⇒ Simplified POS expression → OR-AND Circuit.

Two variable K-Map

$2^2 = 4$ squars.

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

EX. $F(x,y) = x'y + xy' + xy$

$x \backslash y$	0	1
0		1
1	1	1

$F(x,y) = \Sigma(1,2,3)$

⇒ لازم نجمع أكبر عدد من الواحات بحيث يكونوا 2^n. *

* يعني مثلاً ما ينفع أجمع 3 لإنه ما بقدر أمثله بصيغة 2^n.

X بنلغي المتغير المشترك

Y

so $F(x,y) = X + Y$.

ex. $F(x,y) = \bar{x}\bar{y} + \bar{x}y + x\bar{y}$.

x \ y	0	1
0	1	1
1	1	

$\bar{y}$ $\quad$ $\bar{x}$

$F = \Sigma(0,1,2)$

$F = \bar{x} + \bar{y}$ (2 litrals)

ex. $f(x,y) = x'y' + xy + \bar{x}y + xy'$

x \ y	0	1
0	1	1
1	1	1

بما إن كلهم موجودين فبالتالي الجواب = 1

Three variable K-Map

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

the variable that next to Another variable must differ with only one variable

$2^3 = 8$ squars

ex. $f = \bar{x}yz + xy'z' + xy'z + xyz$.

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

$x\bar{y}$ $\quad$ yz

$f = \Sigma(3,4,5,7)$

$f = yz + x\bar{y}$ (4 litrals).

ex. $f(x,y,z) = \Sigma(3,4,6,7)$.

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

$\rightarrow yz$

Adjacent variable $\rightarrow xz'$

$f = xz' + yz$ (4 litrals)

* one square → 3 variable
* two squares → 2 variables
* Four Squares → 1 variables
* Eight Squars → equals 1, "No variables".

⇒ Adjacent squares.

ex. $f(x,y,z) = \Sigma(0,1,2,3,4,6)$

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

$f = \bar{x} + z$

ex. $f(x,y,z) = \Sigma(0,1,2,4,6,7)$

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

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

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

$\bar{z}$

xy

ex. $f(x,y,z) = \Sigma(0,1,2,4,5,6)$.

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

$f = \bar{y} + \bar{z}$

$\bar{y}$

$\bar{z}$

Four variable K-Map

$= 2^4 = 16$ squares.

wx \ yz	00	01	11	10
00	m_0	m_1	m_3	m_2
01	m_4	m_5	m_7	m_6
11	m_{12}	m_{13}	m_{15}	m_{14}
10	m_8	m_9	m_{11}	m_{10}

ex. $F(A,B,C,D) = \Sigma(0,2,4,5,6,7,13,15)$.

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

$\rightarrow \bar{A}\bar{D}$

$\rightarrow BD$

$f = A\bar{D} + BD$

ex. $F(A,B,C,D) = \Sigma(0,1,2,3,8,9,10,11)$.

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

$\rightarrow \bar{B}$

$f = \bar{B}$

ex. $f(A,B,C,D) = \Sigma(0,1,2,3,9,11)$

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

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

$\rightarrow \bar{A}\bar{B}$

$\rightarrow \bar{B}D$

Adjacent squares

# of Squares	# of Variables
1	4
2	3
4	2
8	1
16	0, $f=1$

ex. $f(A,B,C,D) = \Sigma(0,2,4,5,6,7,8,12)$.

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

$\bar{A}\bar{D}$

$\bar{A}B$

$\bar{C}\bar{D}$

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

ex. $F(A,B,C,D) = \Sigma(0,1,2,6,8,9,10)$.

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

$\bar{A}C\bar{D}$

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

$\bar{B}\bar{C}$

❖ For the Boolean function

$$F = W'X'Y' + X'YZ' + W'XYZ' + WX'Y'$$

a) Express the function as a sum-of-minters

b) Find the minimal sum-of-products expression

a) $F = w'x'y'(z+z') + x'yz'(w'+w) + w'xyz' + wx'y'(z+z')$.

$F = w'x'y'z + w'x'y'z' + w'x'yz' + wx'yz' + w'xyz' + wx'y'z + wx'y'z'$

$= m_{0001} + m_{0000} + m_{0010} + m_{1010} + m_{0110} + m_{1001} + m_{1000}$

$= \Sigma(0, 1, 2, 6, 8, 9, 10)$.

b) $2^4 = 16$ squar.

wx \ yz	00	01	11	10
00	1	1		1
01				1
11				
10	1	1		1

$w'yz'$

$x'z'$

$x'y'$

$F(x,y,z,w) = w'yz' + x'z' + x'y'$

Prime Implicant'S

A product term obtained by combining the Max # of Adjacent Squares in the k-map.

* the largest possible groups of [ones].

* بدنا نجيب أكبر عدد ممكن من Adjacent squares.

* عدد الواحدات في المربع لازم يكون a power of 2. $2^0, 2^1, 2^2 \dots 2^n$

Essential prime Implicant

لـو اذا لاقينا [min term] واحد على الأقل لا يمكن ضمه لأكثر من مجموعة، فهذا المربع يعتبر [EPI]

* at least there is a single (1) which can't be combined in any other way

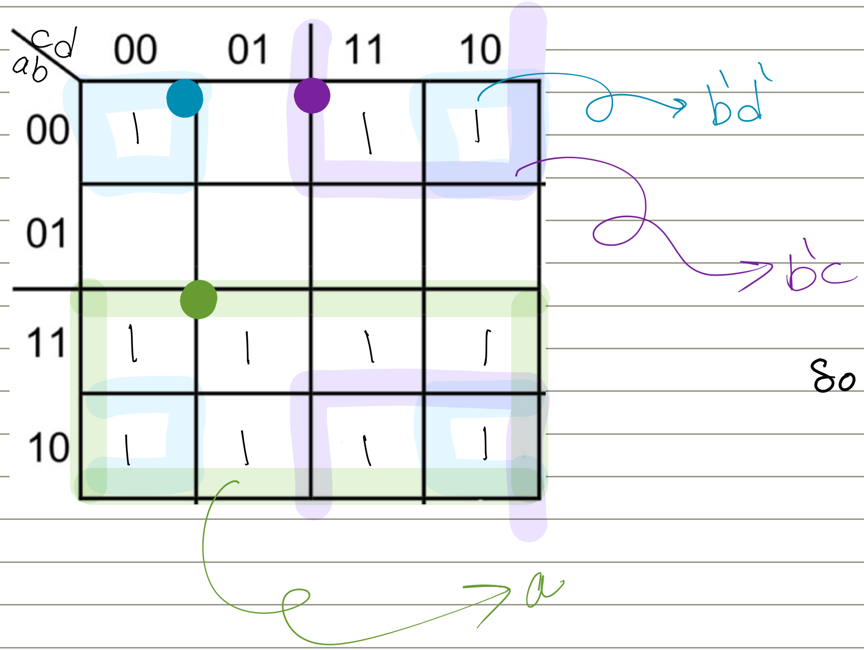
* PI & EPI, can be determined by inspecting K-Map.

Sum of product.

Ex. Find all the PI & EPI for :-

1. $f(a,b,c,d) = \Sigma(0,2,3,8,9,10,11,12,13,14,15)$.

$2^4 = 16$.



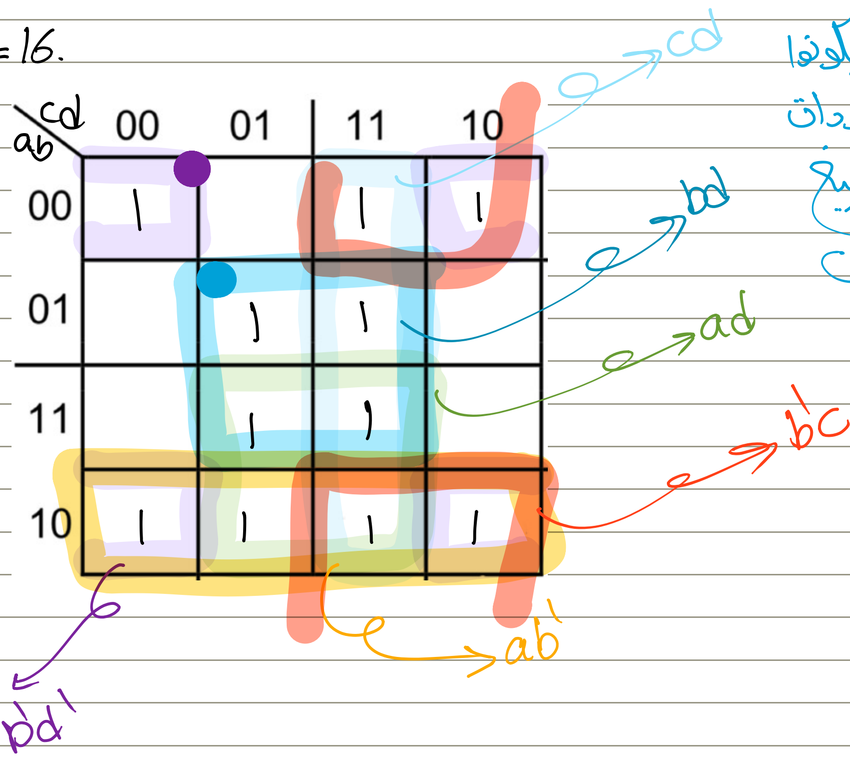
PI
↳ b'd', b'c, a

EPI
↳ b'd', b'c, a

So $F(a,b,c,d) = b'd' + b'c + a$.

Ex.2 ⇒ $f(a,b,c,d) = \Sigma(0,2,3,5,7,8,9,10,11,13,15)$.

$2^4 = 16$.



لازم يكونوا موجودات بكل صيغ الفنكشن

EPI
↳ bd, b'd'

PI
↳ 6 prim Implicants.
bd, b'd', ad, cd, b'c

✱ عند كتابة الصيغة المختصرة لـ f

1. ضروري نكتب EPI.
2. PI, تبقى احتمالات فيمكن كتابة أكثر من صيغة للفنكشن..
3. Choose a minimal subset of prime implicants that cover all remaining 1's
4. ممكن يطلب بالسؤال نكتب كل احتمالات الفنكشن.

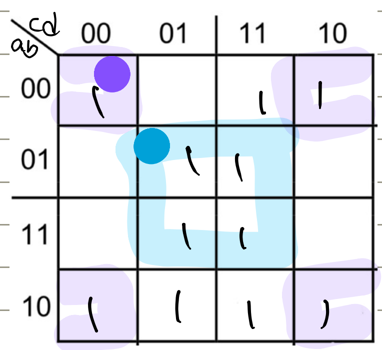
Four possible Solutions :-

$f = bd + b'd' + ab' + cd$

$f = bd + b'd' + ab' + b'c$

$f = bd + b'd' + ad + b'c$

$f = bd + b'd' + cd + ad$



product of sum

Ex. $F(a,b,c,d) = \Sigma(1,2,3,9,10,11,13,14,15)$.

$f'(a,b,c,d) = \Sigma(0,4,5,6,7,8,12)$.

ab \ cd	00	01	11	10
00	1			
01	1	1	1	1
11	1			
10	1			

$a'b$

$c'd'$

$f' = a'b + c'd'$

complement

$f = (a+b')\cdot(c+d)$

minimal pos = 4 litrals.

لما يطلب pos ويكون معطيك SOP بناخذ الcomplement ونطلع f' والجواب النهائي بنرجع نعمله complement.

Standard form simplification procedure

sop ← نوجد f

pos ← نوجد f'

① بنرسم K-Map

② نقارن pos & sop

الجواب يكون الي عنده أقل عدد من المتغيرات [litrals]..

Ex. express the boolean function f in Standard form Using the minimal number of litrals.

1. $F(a,b,c,d) = \Pi(3,4,6,7,11,12,13,14,15)$.

$F(a,b,c,d) = \Sigma(0,1,2,5,8,9,10)$.

بالسؤال أعطاك [max terms] لكن في K-map لازم نحط ب min terms, فلذلك بنحولها ثم نكمل..

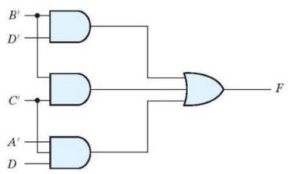
ab \ cd	00	01	11	10
00	1	1		1
01		1		
11				
10	1	1		1

$a'c'd$

$b'd'$

$b'c'$

$F = b'c' + b'd' + a'c'd$ 7 litrals.



$F'(a,b,c,d) = \Sigma(3,4,6,7,11,12,13,14,15)$.

ab \ cd	00	01	11	10
00			1	
01	1		1	1
11	1	1	1	1
10			1	

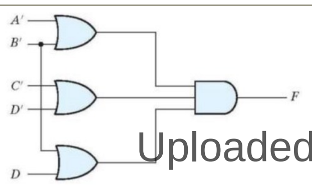
bd'

cd

ab

$F' = bd' + cd + ab$

$F = (b'+d)\cdot(c'+d')\cdot(a'+b')$ 6 litrals



Don't Cares

☐ the don't Care Values Can be selected to be either 0 or 1

Ex. Simplify the function $g(a,b,c,d) = \Sigma(1,3,7,11,15)$, Which has the don't Care Conditions $\Sigma_d(0,2,5)$.

ab \ cd	00	01	11	10
00	X	1	1	X
01		X	1	
11			1	
10			1	

first Solution :- $a'b' + cd$
Second solution :- $a'd + cd$

* Not all don't Cares need to be covered.

* نتعامل مع كل (X) على حدى، مرة بناخذهم 0 ومرة 1.

POS with don't cares

Ex. $g = \Sigma(1,3,7,11,15) + \Sigma_d(0,2,5)$

dont cares بنستطيع نغيرهم

ab \ cd	00	01	11	10
00	X			X
01	1	X		1
11	1	1		1
10	1	1		1

$g' = \Sigma(\cancel{0},\cancel{2},4,\cancel{5},6,8,9,10,12,13,14)$

$g' = \Sigma(4,6,8,9,10,12,13,14)$.

$g' = d' + ac'$

first Sol. → $g = d \cdot (a' + c)$

↳ 3 litrals.

However SOP 4 litrals.

STUDENTS-HUB.com Uploaded By: Rawan Fares

5&6 variables k-Map

$2^5 = 32$, So, each Squar is Adjacent to 5 other Squares.

4 in the Same layer — 1 in other layer.

اختار a و اعتبرها most significant

when a = 0, when a = 1 — two layers

نتخيل *

$bc \backslash de$	00	01	11	10
$a = 0$				
00	m_0	m_1	m_3	m_2
01	m_4	m_5	m_7	m_6
11	m_{12}	m_{13}	m_{15}	m_{14}
10	m_8	m_9	m_{11}	m_{10}

$bc \backslash de$	00	01	11	10
$a = 1$				
00	m_{16}	m_{17}	m_{19}	m_{18}
01	m_{20}	m_{21}	m_{23}	m_{22}
11	m_{28}	m_{29}	m_{31}	m_{30}
10	m_{24}	m_{25}	m_{27}	m_{26}

Ex. given $f(a,b,c,d,e) = \Sigma(0,1,8,9,16,17,22,23,24,25)$

$bc \backslash de$	00	01	11	10
$a = 0$				
00	1	1		
01				
11				
10	1	1		

$bc \backslash de$	00	01	11	10
$a = 1$				
00	1	1		
01			1	1
11				
10	1	1		

$ab'cd$

$c'd'$

$f = ab'cd + c'd'$ [6 literals]

Ex. $g(a,b,c,d,e) = \Sigma(3,6,7,11,24,25,27,28,29) + \Sigma_d(2,8,9,12,13,26)$

$bc \backslash de$	00	01	11	10
$a = 0$				
00			1	X
01			1	1
11	X	X		
10	X	X	1	

$bc \backslash de$	00	01	11	10
$a = 1$				
00				
01				
11	1	1		
10	1	1	1	X

bd'

$g = bd' + bc'e + a'b'd$

not covered.

$bc'e$

$a'b'd$

6 squares Kmap.

cd \ ef	ab = 00				ab = 01				ab = 11				ab = 10			
	00	01	11	10	00	01	11	10	00	01	11	10	00	01	11	10
00	m_0	m_1	m_3	m_2	m_{16}	m_{17}	m_{19}	m_{18}	m_{48}	m_{49}	m_{51}	m_{50}	m_{32}	m_{33}	m_{35}	m_{34}
01	m_4	m_5	m_7	m_6	m_{20}	m_{21}	m_{23}	m_{22}	m_{52}	m_{53}	m_{55}	m_{54}	m_{36}	m_{37}	m_{39}	m_{38}
11	m_{12}	m_{13}	m_{15}	m_{14}	m_{28}	m_{29}	m_{31}	m_{30}	m_{60}	m_{61}	m_{63}	m_{62}	m_{44}	m_{45}	m_{47}	m_{46}
10	m_8	m_9	m_{11}	m_{10}	m_{24}	m_{25}	m_{27}	m_{26}	m_{56}	m_{57}	m_{59}	m_{58}	m_{40}	m_{41}	m_{43}	m_{42}

6 Adjacent
- 4 in the Same layer
- 2 in other layers.

Ex:

$h(a, b, c, d, e, f) = \sigma(2, 10, 11, 18, 21, 23, 29, 31, 34, 41, 50, 53, 55, 61, 63)$

cd \ ef	ab = 00				ab = 01				ab = 11				ab = 10			
	00	01	11	10	00	01	11	10	00	01	11	10	00	01	11	10
00				1				1				1				1
01						1	1			1	1					
11						1	1			1	1					
10			1	1										1		

$c'd'ef'$

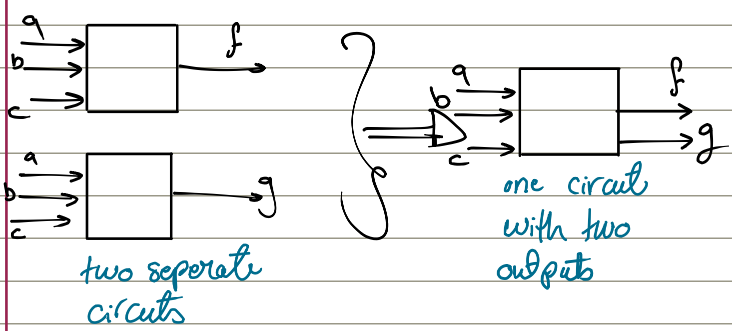
bdf

$a'b'cd'e$

$ab'cd'e'f$

Multiple outputs

- having two functions, with same input, but different output.
- the idea is to minimize both functions, by sharing (terms) (gates) between them.



EX. $f(a,b,c) = \Sigma(0,2,6,7)$.
$g(a,b,c) = \Sigma(1,3,6,7)$.

Q. minimize each circute seperatly.
Q. minimize both functions as one circute.

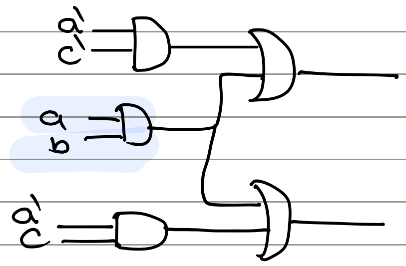
a \ bc	00	01	11	10
0	1			1
1			1	1

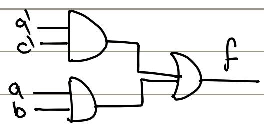
$f = a'c' + ab$

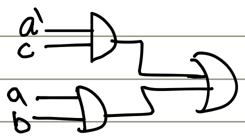
a \ bc	00	01	11	10
0		1	1	
1			1	1

$g = a'c + ab$

$f = a'c' + ab$
$g = a'c + ab$







$f(a,b,c,d) = \sigma(3,5,7,10,11,14,15)$, $g(a,b,c,d) = \sigma(1,3,5,7,10,14)$

1. draw K-Map.
2. extract the Common between the two functions.

f

ab \ cd	00	01	11	10
00			1	
01		1	1	
11			1	1
10			1	1

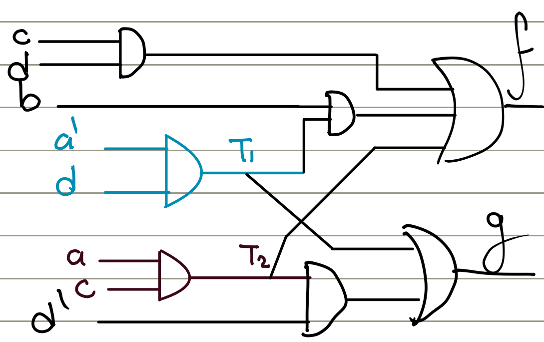
g

ab \ cd	00	01	11	10
00		1	1	
01		1	1	
11				1
10				1

$f = a'bd + cd + ac$

$g = acd' + a'd$

$T_1 = a'd$, $T_2 = ac$
$f = T_1 b + cd + T_2$
$g = T_1 + T_2 d'$



NAND and NOR Gate

تم تغطيته في تشابتر 2

EX. Implement the boolean function so

$f(x,y,z) = \Sigma(1,2,3,5,7)$ using only NOR gate.

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

$f = z + x'y$

❖ Example: Implement the Boolean function
$f(x, y, z) = \sigma(1, 2, 3, 5, 7)$ using only **NOR** gates

NOR = POS

$f'(x,y,z) = \Sigma(0,4,6)$

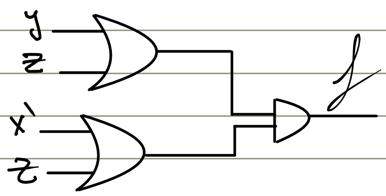
$x \backslash yz$	00	01	11	10
0	1			
1	1			1

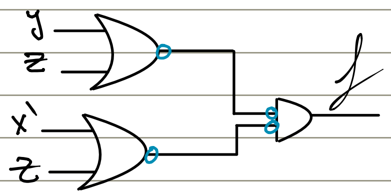
$f' = y'z' + xz'$

$f = (y+z).(x'+z).$

ممكن نحل ال Kmap ل function f.
ونجمع الأصفار بدل الواحدات
كطريقة أسرع

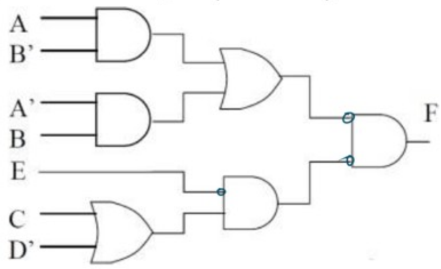
أخذنا f' ⟸ لأنه
المطلوب إنه نرسم باستخدام
NOR gate



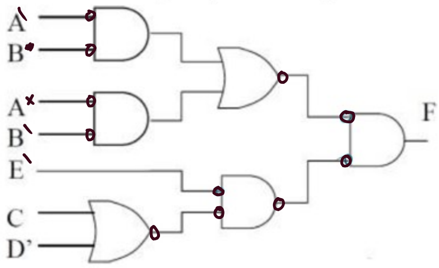


❖ Example: Implement the Boolean function

$f(A, B, C, D, E) = (AB' + A'B)E(C + D')$ using only **NOR** gates



always start from out put to input



parity :- extra bit added to the massage to check if the recive massage is correct.

only to check single error.

parity < odd → use even function
even → use odd function.

يكون محدد بالسؤال

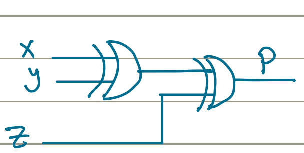
Ex. message x, y, z

x	y	z	P (even) parity
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

عدد الواحدات زوجي فلذلك لا يوجد error ←

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	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 → عدد الواحدات فردي
0 → عدد الواحدات زوجي

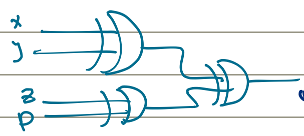


بتفحص ال parity generation

XOR → تستخدم في الاتصالات

∗ 0 → there is no error.
1 → there is error

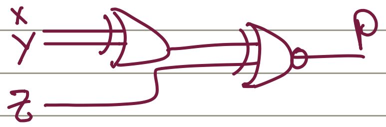
بتفحص ال error
no error ← 0
error ← 1



"Checker"

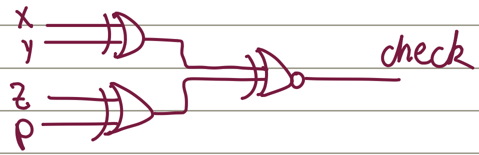
for odd parity.

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



to check errors :-

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



Odd parity :- Count of 1's in the (n+1)-bit Code is odd.

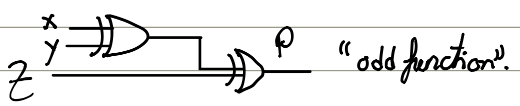
- use an even function to generate the odd parity bit.
- use an even function to check the (n+1) - bit Code.

Even Parity :- Count of 1's in the (n+1)'s bit Code is even.

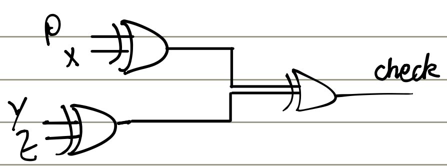
- Use an odd function to generate the even parity bit.
- Use an odd function to check the (n+1) bit Code.

Q. Design even parity generator & checker for 3-bit Codes.

even parity → odd function.
// check for 3 bit Code.



Parity generator.



Parity checker

example:-
xyz = 010
generator → p = 1
Pxyz = 1010
checker → 0