Ch.2 |Boolean Algebra and logic gates

By: Rawan Alfares

Boolean Algebra

* A Set of two values : $B = \{0,1\}$
* Three basic operation : AND, OR, NOT.
  * denoted by: (.) (+) (-), (')
*

two inputs

AND

x y	x·y
0 0	0
0 1	0
1 0	0
1 1	1

OR

x y	x+y
0 0	0
0 1	1
1 0	1
1 1	1

Complement — one input.

x	x'
0	1
1	0

Postulates Of Boolean Algebra

1. Closure :- the result of any Boolean Operation is 0 or 1 .

2. 0 is The Identity element for (+) :-
   $0 + X = X + 0 = X$
   1 is The Identity element for (.) :-
   $1 . X = X . 1 = X$

3. Commutative law
   $X + Y = Y + X$
   $X . Y = Y . X$

4. Distributive law :-
   $X . (Y + Z) = XY + XZ$
   $X + YZ = (X + Y)(X + Z)$

5. Complement law :- " for any X, $\exists$ X' "
   $X + \bar{X} = 1$
   $X . \bar{X} = 0$

Boolean Functions

Consists of :-
1. variabels : x , y ...
2. Boolean Constants : 0's and 1's
3. Boolean Operators : AND , OR , NOT
4. parentheses

Operator precedence :-
1. parentheses . ( )
2. NOT (')
3. AND (.)
4. OR (+)

* number of rows : 2^n ($n \to$ number of variabels)

ex. The truth tabel for $f = xy' + x'z$

x	y	z	y'	xy'	x'	x'z	xy'+x'z
0	0	0	1	0	1	0	0
0	0	1	1	0	1	1	1
0	1	0	0	0	1	0	0
0	1	1	0	0	1	1	1
1	0	0	1	1	0	0	1
1	0	1	1	1	0	0	1
1	1	0	0	0	0	0	0
1	1	1	0	0	0	0	0

NOTEs:-

- $1 + y = 1$
- $x + \bar{x} = 1$
- $x \cdot \bar{x} = 0$
- $x + x = x$

ex. $(x+y)' = x'y'$
$(xy)' = x'+y'$
] demorgan law →

(x') convert to (x)
(x) convert to (x')
(+) convert to (·)
(·) convert to (+)

x	y	x'	y'	x+y	(x+y)'	x'y'
0	0	1	1	0	1	1
0	1	1	0	1	0	0
1	0	0	1	1	0	0
1	1	0	0	1	0	0

Identical.

what's the Complement of $f = x'yz' + xy'z$

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

ex. What's the complement of $g = (a'+bc)d' + e$

$g' = (a.(b'+c') + d).e'$

Dual Pairs :- we only change, [0's ⇆ 1's, + ⇄ ·] and keep variabels as they are.

Proving Some Expressions

ex. prove $x + xy = x$ absorbtion theorem

$$\begin{aligned} x + xy &= x.1 + xy \\ &= x.(1 + y) \\ &= x.1 \\ &= x \end{aligned}$$

ex. prove $x + x'y = x + y$

$$\begin{aligned} x + x'y &= (x + \bar{x}).(x + y) \\ &= 1.(x+y) \\ &= x + y. \end{aligned}$$

ex. prove that $xy + x'z + yz = xy + x'z$

$$\begin{aligned} &= xy + x'z + yz.1 \\ &= xy + x'z + yz.(x + \bar{x}) \\ &= xy + x'z + yzx + yz\bar{x} \\ &= xy + xyz + x'z + \bar{x}zy \\ &= xy(1+z) + x'z(1+y) \\ &= xy.1 + \bar{x}z.1 \\ &= xy + \bar{x}z \end{aligned}$$

ex. Simplify the following Boolean function to a minimum number of literals

$F(A,C) = (A+C)' + (A+C)(A'+C')$ 6 litral

$$\begin{aligned} &= x' + xy \\ &= x' + y \end{aligned}$$

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

$$\begin{aligned} &= (A+C)' + (A'+C') \\ &= A'.C' + A'+C' \\ &= A' + C' \end{aligned}$$

4 litral.

2 litral.

ex. $A'B' + B'C + AB'C' + AB$

$= B'(A' + (C + AC')) + AB$ distributive law 9 litrals, 7 litrals

$= B'((A' + A) + C) + AB$ simplification law 6 litrals

$= B'(1 + C) + AB$ Complement law 4 litrals

$= B'.1 + AB$ null law 3 litrals

$= B' + AB$ Identity law 3 litrals

$= A + B'$ 2 litrals

ex. $ab + a'cd + a'bd + a'cd' + abcd$

$= ab + abcd + a'cd + a'bd + a'cd'$ 15 litrals

$= ab(1 + cd) + a'c(d + d') + a'bd$

$= ab + a'c + a'bd$

$= b(a + a'd) + \bar{a}c$

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

ex. $(A' + B' + C')(A + C')(B + C')(B' + C)$ Product of Sum

by duality:- they are not equivilant, so at the end we have to reuse duality.

$= (A'B'C') + (AC') + (BC') + (B'C)$

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

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

$= C'(A + 1) + B'C$

$= C' + B'C$

$= C' + B'$ by dual

$= C' \cdot B'$

NOTE:

the Complement of a function is by taking the dual then complementing each litral

ex. Complement $f = x'yz' + xy'z'$

1. dual: $(x' + y + z') \cdot (x + y' + z')$

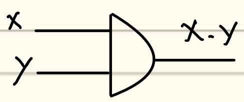
2. Complement $(x + y' + z) \cdot (x' + y + z$

ex. $g = (\bar{a} + bc)d' + e$

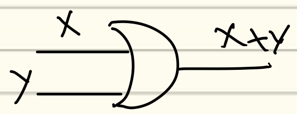
1. $((\bar{a} \cdot (b+c)) + d') \cdot e$

2. $g' = ((a \cdot (b' + c')) + d) \cdot e'$

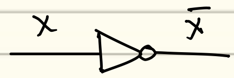
* 0 → low voltage
* 1 → high voltage



AND gate



OR gate



NOT gate (invertor).

Canonical and Standard Form

* بدي أستخدم لغة أفهمها أنا والجهاز | Canonical

minterms ⟷ max terms

they are complement to each other.

- must have the all variabels in same order for all terms. } إحدى سلبياتها إنها تحوي على أعداد كبيرة من variabels *

ex.

variabels			minterms (always 1)		max terms (always 0)	
x	y	z	term	design	term	design
0	0	0	$x'y'z'$	m_0	$x+y+z$	M_0
0	0	1	$x'y'z$	m_1	$x+y+z'$	M_1
0	1	0	$x'yz'$	m_2	$x+y'+z$	M_2
0	1	1	$x'yz$	m_3	$x+y'+z'$	M_3
1	0	0	$xy'z'$	m_4	$x'+y+z$	M_4
1	0	1	$xy'z$	m_5	$x'+y+z'$	M_5
1	1	0	xyz'	m_6	$x'+y'+z$	M_6
1	1	1	xyz	m_7	$x'+y'+z'$	M_7

inside: product
outside: sum

in: sum
out: product

ex. $f(x,y,z) = x'y'z' + x'y'z + xyz$ — Sum of product.

$0'.0'.0' + 0'.0'.1 + 1.1.1$

$= m_0 + m_1 + m_7$

$= \Sigma\, 0,1,7$

ex. $F(A,B,C) = \Sigma\, 0,2,4,6$

$= m_0 + m_2 + m_4 + m_6$

$= 000 + 010 + 100 + 110$

$= A'B'C' + A'BC' + AB'C' + ABC'$

ex. $F(A,B,C,D) = \Sigma\, 0,2,10,15$

$= m_0 + m_2 + m_{10} + m_{15}$

$= 0000 + 0010 + 1010 + 1111$

$= A'B'C'D' + A'B'CD' + AB'CD' + ABCD.$

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

$(0+0+0) . (0+1+0) . (1+1+1)$

$= M_0 . M_2 . M_7$

$= \Pi\,(0,2,7).$

ex. $F(A,B,C) = \Pi(0,2,5)$

$$= M_0 \cdot M_2 \cdot M_5$$
$$= (0+0+0) \cdot (0+1+0) \cdot (1+0+1)$$
$$= (A+B+C) \cdot (A+B'+C) \cdot (A'+B+C').$$

ex. $F(A,B,C,D) = (A+B+C+D) \cdot (\bar{A}+\bar{B}+C+D) \cdot (\bar{A}+B+\bar{C}+D).$

$$= (0+0+0+0) \cdot (1+1+0+0) \cdot (1+0+1+0).$$
$$= M_0 \cdot M_{12} \cdot M_{10}$$
$$= \Pi(0, 10, 12).$$

ex. $F(A,B,C) = \Sigma(0,7)$

$$= m_0 + m_7$$
$$= A'B'C' + ABC$$

ex. $F(A,B,C) = (A+B+C) \cdot (\bar{A}+\bar{B}+\bar{C})$

$$= M_0 \cdot M_7$$
$$= \Pi(0,7)$$

ex. write $F(A,B,C) = \Sigma(0,2,4)$

1. mathmatical expression.
2. truth tabel.
3. product of Max term.

1.
$$= m_0 + m_2 + m_4$$
$$= m_{000} + m_{010} + m_{100}$$
$$= (A'B'C') + (A'BC') + (AB'C')$$

2	A	B	C	min term	max term
m_0	0	0	0	1	1
m_1	0	0	1	0	0
m_2	0	1	0	1	1
m_3	0	1	1	0	0
m_4	1	0	0	1	1
m_5	1	0	1	0	0
m_6	1	1	0	0	0
m_7	1	1	1	0	0

3. $F(A,B,C) = \Pi(1,3,5,6,7)$

F for product $= \Pi(1,3,5,6,7).$

complement

$$F(A,B,C) = \Sigma(0,2,4) = \Pi(1,3,5,6,7).$$
$$F'(A,B,C) = \Pi(0,2,4) = \Sigma(1,3,5,6,7)$$
$$F \cdot F' = 0$$

ex. express the Boolean function $F(A,B,C) = A + B'C$ as A Sum of minterms.

missing two variabels

$A = A(B+B') = AB + AB'$

$AB(C+C') = ABC + ABC'$

$AB'(C+C') = AB'C + AB'C'$

Second term : missing one variable

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

$F(A,B,C) = ABC + ABC' + AB'C + AB'C' + A'B'C$

$= m_7 + m_6 + m_5 + m_4 + m_1$

$= \Sigma(1,4,5,6,7) = \Pi(0,2,3)$

Complement $F(A,B,C) = \Pi(1,4,5,6,7) = \Sigma(0,2,3)$.

Express the following in the Sum of min terms :-

1) $f(a,b,c,d) = \Sigma(2,3,6,10,11)$

0010 0011 0110 1010 1011

$= m_2 + m_3 + m_6 + m_{10} + m_{11}$

$= a'b'cd' + a'b'cd + a'bcd' + ab'cd' + ab'cd$

2) $g(a,b,c,d) = \Sigma(0,1,12,15)$

0000 0001 1100 1111

$= m_0 + m_1 + m_{12} + m_{15}$

$= a'b'c'd' + a'b'c'd + abc'd' + abcd$

*Express the following function as a Product of max terms :-

$f(x,y,z) = xy + x'z$ → مش كل المتغيرات موجودة فبدنا نحطهم بكل تيرم .

① Convert function into OR terms by using the distributive law.

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

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

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

② each term missing one variabel :

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

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

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

$= M_4 . M_5 . M_0 . M_2$

$= \Pi(0, 2, 4, 5)$

* express $f(a,b,c,d) = \Pi(1,3,11)$ in the product of max terms form

$= M_1 . M_3 . M_{11}$

$= (0+0+0+1) . (0+0+1+1) . (1+0+1+1)$

$= (a+b+c+d') . (a+b+c'+d') . (a'+b+c'+d')$

* $g(a,b,c,d) = \Pi(0, 5, 13)$

1100

$= M_0 . M_5 . M_{13}$

$= (0+0+0+0) . (0+1+0+1) . (1+1+0+1) .$

$= (a+b+c+d) . (a+b'+c+d') . (a'+b'+c+d') .$

To convert from one canonical to another, interchange the symbols Σ and Π and list those number missing from the original form.

The complement of a function expressed by a Sum of Minterms is the product of Max terms → interchang Σ and Π, and keep the same list of indices.

ex. write the complement of the following function using Sum of minterms.

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

$f'(x,y,z) = \Pi(0,2,3,4,6) = \Sigma(1,5,7)$

$f'(x,y,z) = x'y'z + xy'z + xyz$

ex. $f(x,y,z,w) = \Sigma(0,1,2,4,5,7)$

Since the system has 4 input variabels, then the number of minterm and maxterms $= 2^4 = 16$, $0 \rightarrow 15$

$$f(x,y,z,w) = \Sigma(0,1,2,4,5,7)$$
$$= \Pi(3,6,8,9,10,11,12,13,14,15)$$

$$f'(x,y,z,w) = \Pi(0,1,2,4,5,7)$$
$$= \Sigma(3,6,8,9,10,11,12,13,14,15).$$

* the number of Minterms and Maxterms $= 2^n$ → n → number of variabels
* Minterms → always 1 entries
* Maxterms → always 0 entries
* the complement of a sum of minterms is product of Maxterms to the same indicies and vice versa.

* for Boolean function, given the list Minterms indicies one can determine the list of Maxterms indices and vice versa.

* باختصار، إذا أعطاك معلومة وحدة بتقدر تطلع منها باقي المعلومات،

* complement ← بنحوّل $\Pi \rightarrow \Sigma$ ، $\Sigma \rightarrow \Pi$ ونخلي الأرقام نفسها

* لكن إذا بدنا نكتب Σ بطريقة Π (يعني جواب مكافئ) بناخذ الأرقام اللي ضايلات، والعكس كذلك...

Operation on Functions

And = intersection التقاطع
OR = union اتحاد

ex. $F(A,B,C) = \sum m(1,3,6,7)$, $F'(A,B,C) = \sum(0,2,4,5)$.

$G(A,B,C) = \sum m(0,1,2,4,6,7)$, $G'(A,B,C) = \sum(3,5)$.

Find $F.G = \sum m(1,6,7)$

$F+G = \sum m(0,1,2,3,4,6,7)$

$F'.G = \sum m(0,2,4)$.

Equality functions

1. Same sum of minterms
2. Same product of Max terms

ex. $F_1 = a'b' + ac + bc'$
$F_2 = a'c' + ab + b'c$

F_1

a	b	c	
0	0	0/1	0, 1
1	0/1	1	5, 7
1/0	1	0	2, 6

$\Rightarrow F_1 = \sum(0,1,2,5,6,7)$
$= \prod(3,4)$

F_2

a	b	c	
0	1/0	0	0, 2
1	1	0/1	6, 7
0/1	0	1	1, 5

$\rightarrow F_2 = \sum(0,2,6,7,1,5)$
$= \prod(3,4)$

yes, they are equivilant.

ex. $F_1(x,y,z) = \sum m(1,2,4,5,6,7)$ = Combinations $= 2^3 = 8$

$F_2(a,b) = \prod(0,3)$ $\rightarrow$ Combinations $= 2^2 = 4$

$F_2(a,b) = \sum(1,2)$

$F_1(x,y,z) = \prod(0,3)$

thus, they are not equal.

Standard form

another way to express Boolean functions

Sum of product SOP
Product of sum POS

1. SOP

Boolean expression is the ORing (sum) of ANDterms (product).

ex. $f_1 = xy' + xz$
$f_2 = y + xy'z$

↳ similar to min term

2. POS

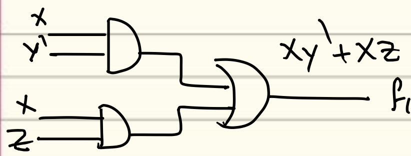
Boolean expression is the ANDing (product) of OR terms (sums).

ex. $f_3 = (x+z)(x'+y')$
$f_4 = x(x+y+z)$

↳ similar to max term.

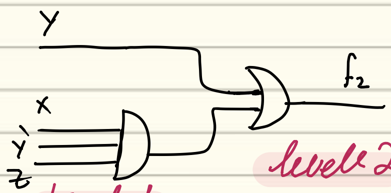
Two level Gate Implementation

$f_1 = xy' + xz$



level 1 AND
level 2 OR

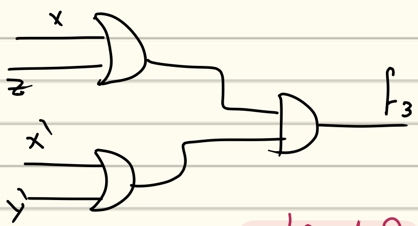
$f_2 = y + xy'z$



level 1 AND
level 2 OR

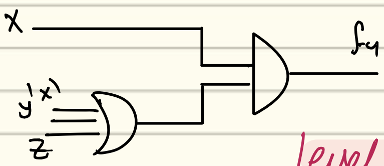
AND - OR implementation
SOP

$f_3 = (x+z)(x'+y')$



level 1 OR
level 2 and

$f_4 = x(x'+y'+z)$



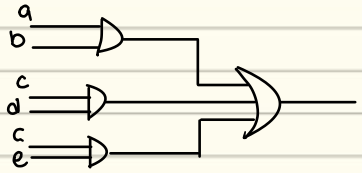
level 1 OR
level 2 and

OR - AND implementation
POS

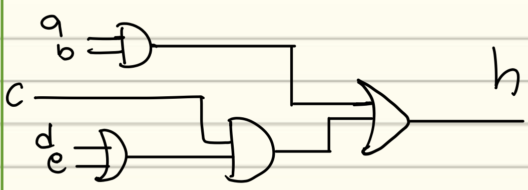
ex. h = ab + cd + ce (6 litral) → standard form SOP
(a)

h = ab + c(d + e) (5 litral) → non-standard form

(a) standard form



(b) non_standard form

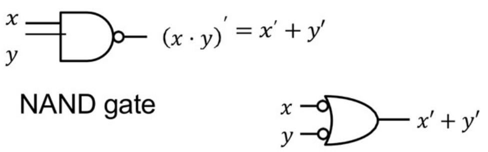


Additional Logic Gates

→ low Cost implementation
→ usfel in implementing Boolean function.

NAND Gate

→ NOT AND
opposite of AND



NAND gate

Another symbol for NAND

x y	NAND
0 0	1
0 1	1
1 0	1
1 1	0

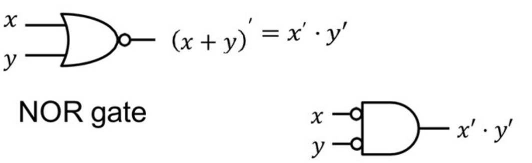
universal gate :- NAND / NOR

يعني بنقدر نبني أكثر من gate منها
سواء كان
→ invertor
AND
→ OR

NOR Gate

→ NOT OR
opposite of OR.

* NAND/NOR
Not assotiative.



NOR gate

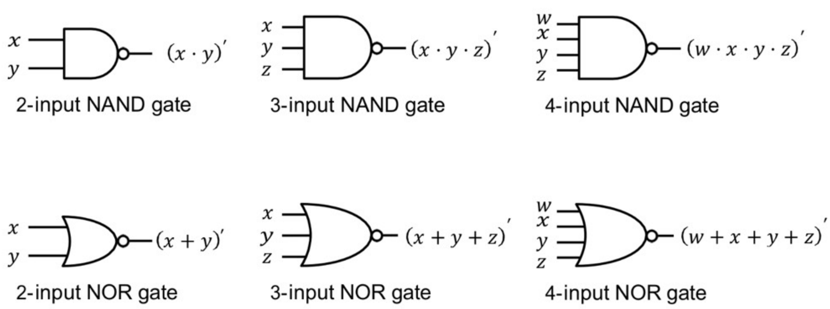
Another symbol for NOR

x y	NOR
0 0	1
0 1	0
1 0	0
1 1	0

* (x NAND y) NAND z ≠ x NAND (y NAND z).

* (x NOR y) NOR z ≠ x NOR (y NOR z)

NAND/NOR gates can have multiple inputs, similar to AND/OR gates

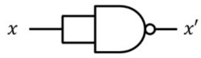


NAND

using NAND gate to build inverter, AND, OR } gates

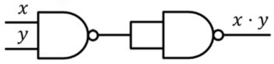
- A single-input NAND gate is an inverter

  x NAND $x = (x \cdot x)' = x'$



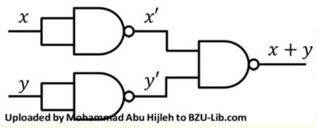
- AND is equivalent to NAND with **inverted output**

  $(x \text{ NAND } y)' = ((x \cdot y)')' = x \cdot y$ (AND)



- OR is equivalent to NAND with **inverted inputs**

  $(x' \text{ NAND } y') = (x' \cdot y')' = x + y$ (OR)

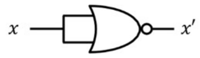


NOR

using NOR gate to build inverter, AND, OR } gate

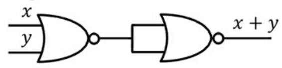
- A single-input NOR gate is an inverter

  x NOR $x = (x + x)' = x'$



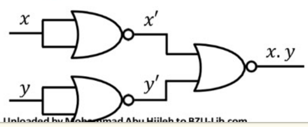
- OR is equivalent to NOR with **inverted output**

  $(x \text{ NOR } y)' = ((x + y)')' = x + y$ (OR)



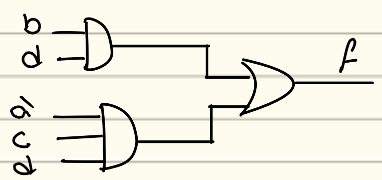
- AND is equivalent to NOR with **inverted inputs**

  $(x' \text{ NOR } y') = (x' + y')' = x \cdot y$ (AND)

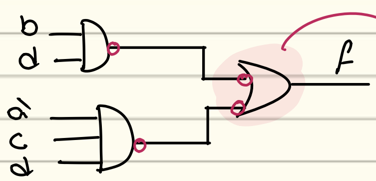


* the multiple Input NAND/NOR Gates are a single gate, NOT a Combination of 2-input gate.

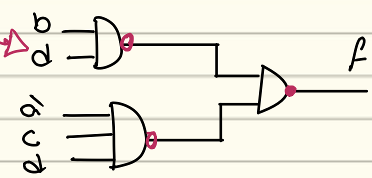
ex. $f = bd + a'cd'$



AND-OR Implementation



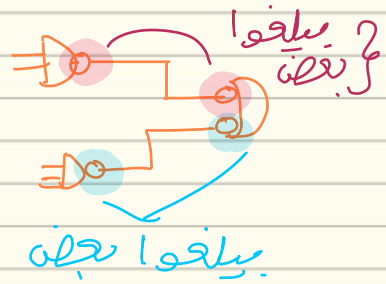
NAND-NOR Implementation



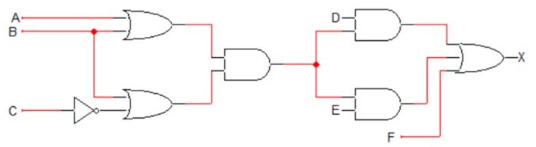
NAND-NAND Implementation

* two successive bubbels onsame line Cancel each other

إذا كان فيه أكثر من (NOT) على نفس السلك بيلغوا بعض

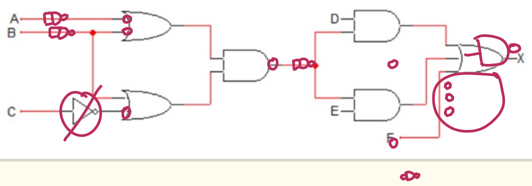


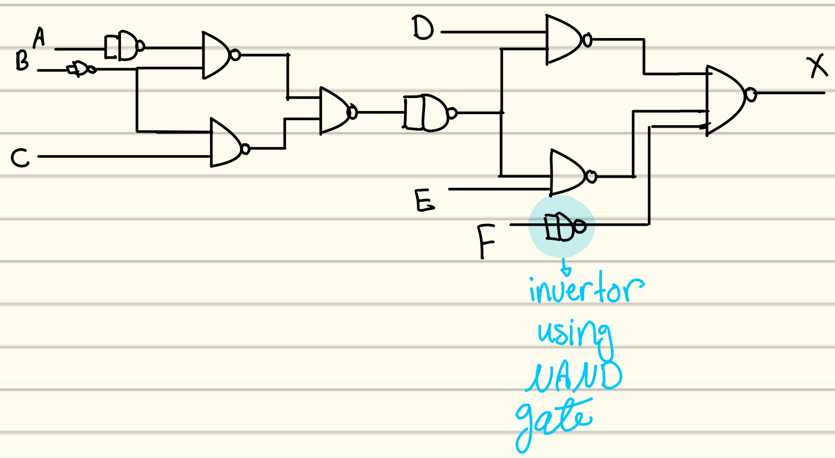
❖ Example: Implement the given circuit using only **NAND** gates



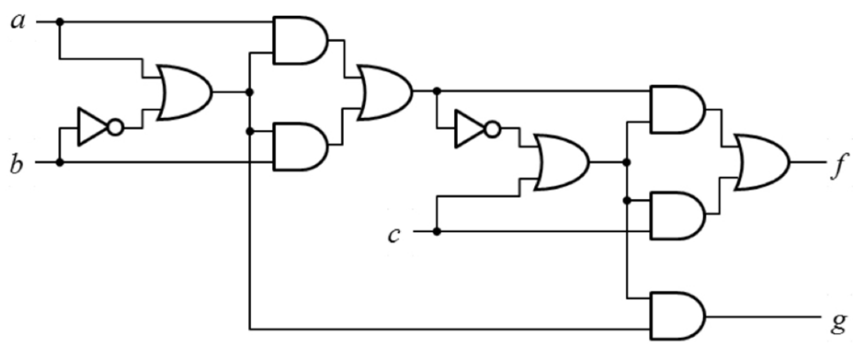
Start from output toward inputs converting gate by gate

❖ Example: Implement the given circuit using only **NAND** gates



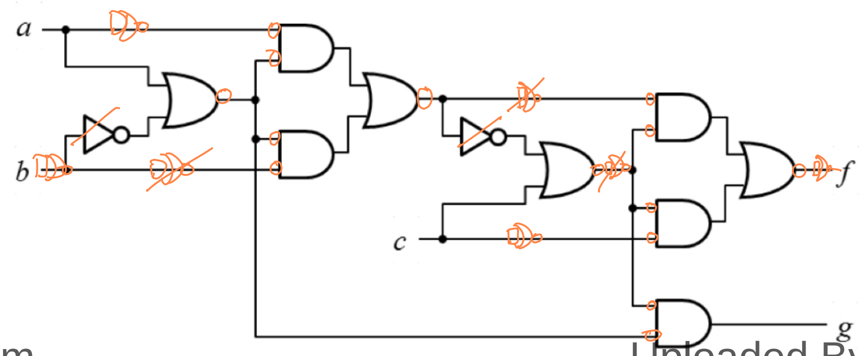


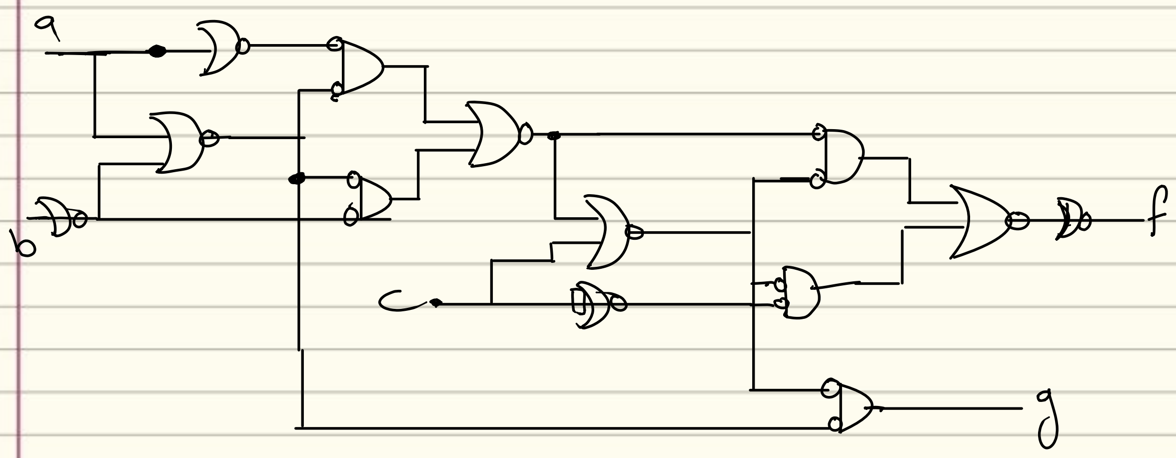
❖ Example: Implement the given circuit using only **NOR** gates



Start from output toward inputs converting gate by gate

❖ Example: Implement the given circuit using only **NOR** gates





NAND → NOR

NOR → NAND

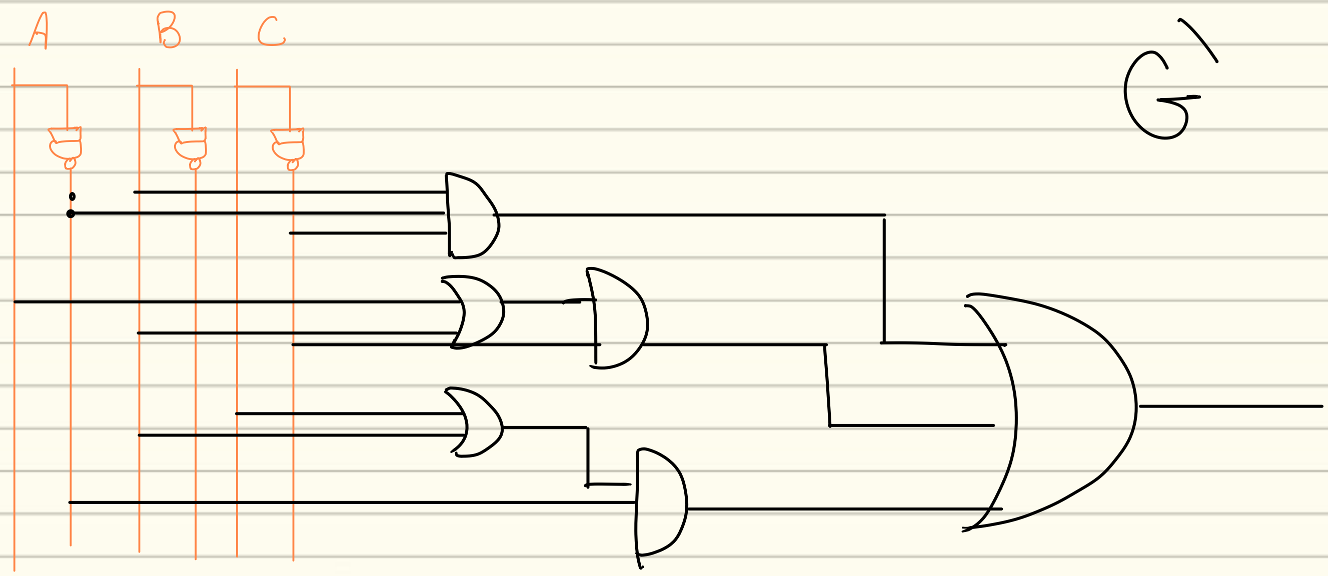
بنشتغل على
input:

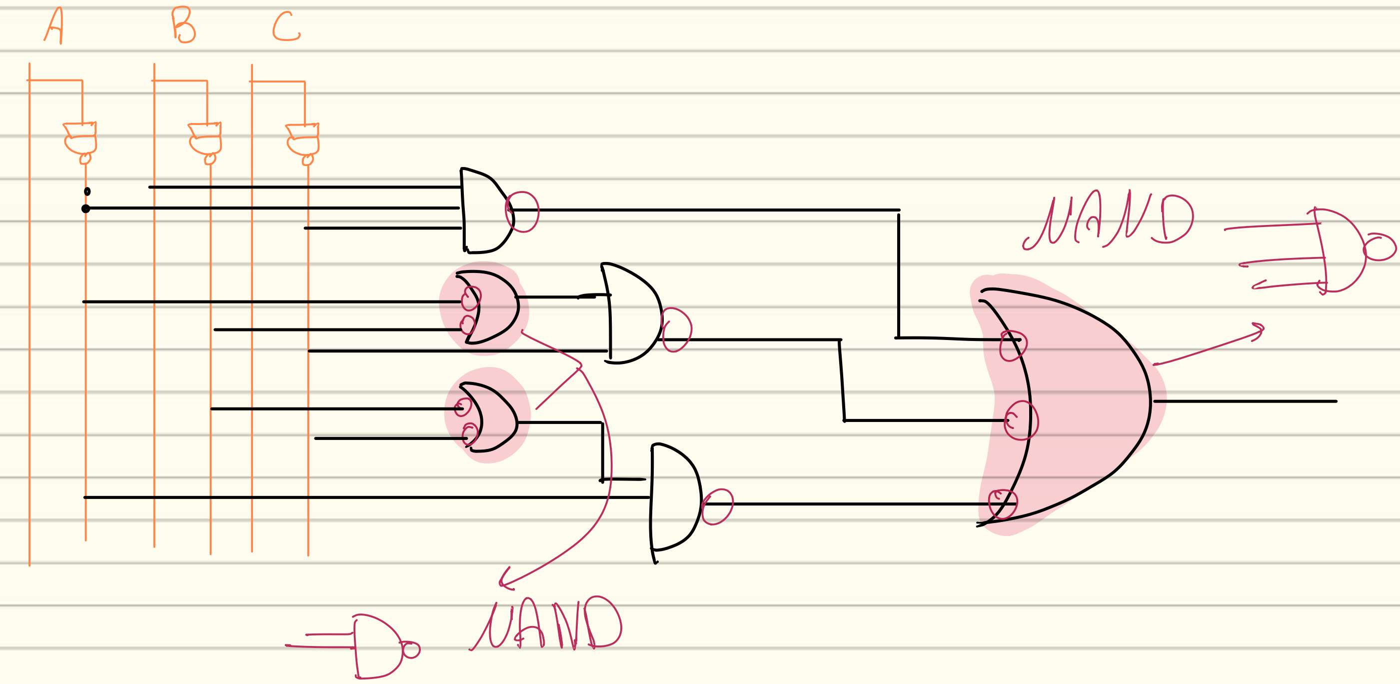
❖ Example: Find the complement of the following expression and implement it using (1) NAND gates, and (2) NOR gates:

G(A, B, C) = (A + B' + C)(A'B' + C)(A + B'C')

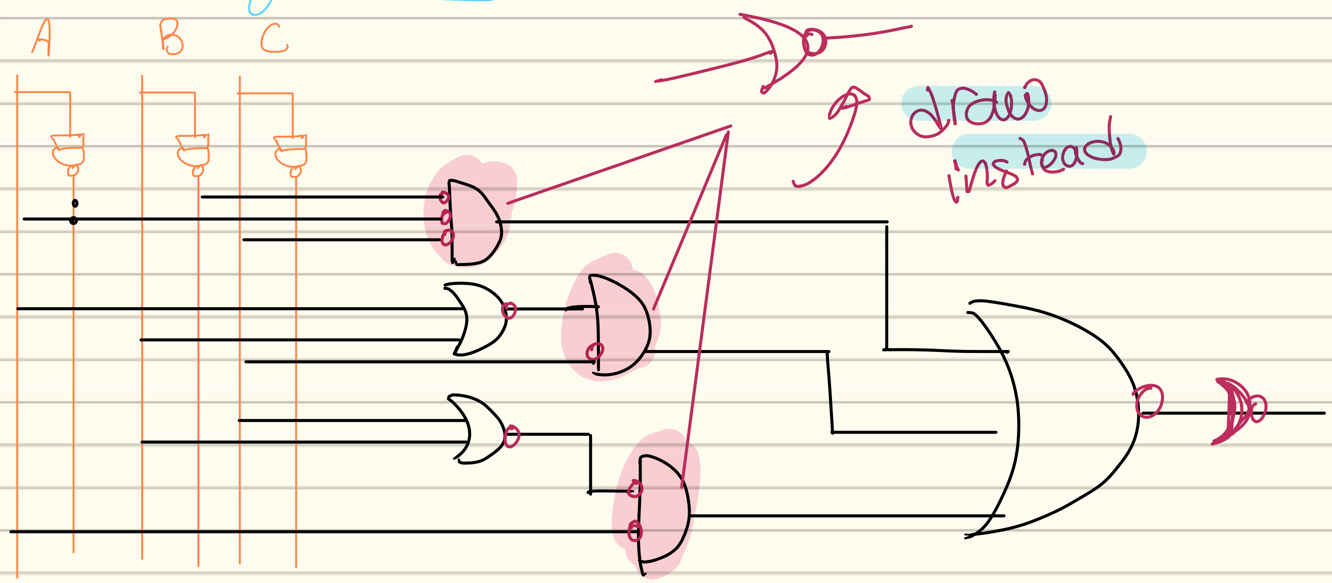
$$G'(A,B,C) = (A+B'+C)' + (A'B'+C)' + (A+B'C')'$$

$$= A'BC' + ((A+B).C') + (A'.(B+C))$$





Same thing to NOR



XOR / XNOR

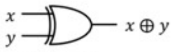
complement for each other.

❖ Exclusive NOR (XNOR) is the complement of XOR

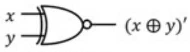
x y	XOR
0 0	0
0 1	1
1 0	1
1 1	0

x y	XNOR
0 0	1
0 1	0
1 0	0
1 1	1

XNOR is also known as **equivalence**



XOR gate



XNOR gate

XOR : Odd function
↳ equals 1, when has odd # of ones

XNOR : equals 0, when has odd # of zeros.

* they don't exist for more than two inputs.

* XOR function :- $x \oplus y = xy' + x'y$
* XNOR function :- $(x \oplus y)' = xy + x'y'$

❖ $x \oplus 0 = x$

❖ $x \oplus 1 = x'$

❖ $x \oplus x = 0$

❖ $x \oplus x' = 1$

❖ $x \oplus y = y \oplus x$

❖ $x' \oplus y' = x \oplus y$

❖ $(x \oplus y)' = x' \oplus y = x \oplus y'$

XOR and XNOR are **associative** operations

❖ $(x \oplus y) \oplus z = x \oplus (y \oplus z) = x \oplus y \oplus z$

❖ $((x \oplus y)' \oplus z)' = (x \oplus (y \oplus z)')' = x \oplus y \oplus z$

Positive and negative logic

* choosing (1) to High level → positive logic system.
* choosing (1) to low level → negatie logic system.

❖ Choosing the high-level **H** to represent logic **1** defines a positive logic system

(a) Positive logic

❖ Choosing the low-level **L** to represent logic **1** defines a negative logic system

(b) Negative logic

❖ It is up to the user to decide on a positive or negative logic polarity

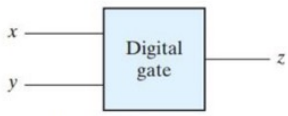
* the conversion from positive to negatie logic it as using dual.
* changing 0's to 1's and vice versa.
* changing (+) to (.) and vice versa.

❖ The *conversion* from positive logic to negative logic and vice versa is essentially an operation that changes 1's to 0's and 0's to 1's in both the inputs and the output of a gate

❖ Since this operation produces the *dual* of a function, the change of all terminals from one polarity to the other results in taking the dual of the function

x	y	z
L	L	L
L	H	L
H	L	L
H	H	H

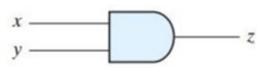
(a) Truth table with H and L



(b) Gate block diagram

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

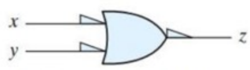
(c) Truth table for positive logic



(d) Positive logic AND gate

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

(e) Truth table for negative logic



(f) Negative logic OR gate

ربنا تقبل منا إنك أنت السميع العليم

روان فارس

من القدس الا أنت