

Teorema de D'Morgan



No hay viento favorable para el que no sabe dónde va
Lucio Anneo Séneca

Calendario propuesto

	Lunes	Martes	Miércoles	Jueves	viernes
4	25 Teorema de D'Morgan Ocho formas estándar	26	27 Ocho Formas Estándar	28	29 Trabajo en el aula de la AF1
6	1 Ejercicios de la PF3	2	3 Minimización de funciones Booleanas	4	5 Minimización de funciones Booleanas
7	8 Ejercicios en clase	9	10 Mapas de Karnaugh	11	12 Mapas de Karnaugh
8	15 Fecha Límite entrevista AF1	16	17 Ejercicios de la PF4	18	19 Reglas para el examen de M. C. AF2
9	22	23	24	25 M1	26 M2
10	29 M3	30 M4	19	20	21

Este próximo viernes habrá sesión de trabajo, traer su material de la AF1

Augustus de Morgan ? -1871

- Augustus de Morgan nació en Madura, India de padres Ingleses
- Profesor de matemáticas en el Colegio Universitario de Londres entre 1828 y 1866
- Primer presidente de la Sociedad de Matemáticas de Londres.
- De Morgan se interesó especialmente por el álgebra.
- Fue tutor de **Ada Lovelace**. Quien fue la primera programadora, desde que escribió la manipulación de los símbolos, de acuerdo a las normas para una máquina de Charles Babbage que aún no había sido construida.



Augustus de Morgan ? -1871

Escribió varias obras de lógica en las que se encuentra la idea de aplicar en esta esfera los métodos matemáticos, así como los primeros resultados de tal aplicación.

En la moderna lógica matemática, llevan el nombre de De Morgan las siguientes leyes fundamentales del álgebra de la lógica:

“Si se tienen dos conjuntos A y B , el complemento de su unión es igual a la intersección de sus complementos y el complemento de su intersección es igual a la unión de sus complementos.”

Teorema de D'Morgan

Una adivinanza

Augustus de Morgan (?-1871) fue un matemático inglés nacido en la India.

Acostumbraba a recrearse en el planteamiento de adivinanzas y problemas ingeniosos.

Este personaje nacido en el siglo XIX, planteaba esta adivinanza sobre su edad:

"El año x^2 tenía x años. ¿En qué año nací?".

Augustus de
Morgan
?-1871



Teorema de D'Morgan

Este personaje nacido en el siglo XIX planteaba esta adivinanza sobre su edad:

"El año x^2 tenía x años. ¿En qué año nació?".

Ecuación que describe, en el año x^2 tenía x años.

?

$$x^2 - x = A$$

Teorema de D'Morgan

Augustus de Morgan (?-1871) fue un matemático inglés nacido en la India.

"El año x^2 tenía x años. ¿En qué año nací?".

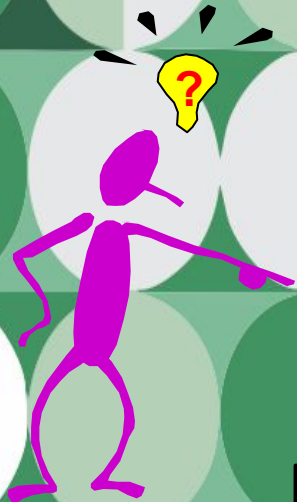
$$x^2 - x = A$$

$$43^2 = 1849$$

$$1849 - 43 = 1806$$

Augustus
de Morgan
? -1871

Teorema de D'Morgan

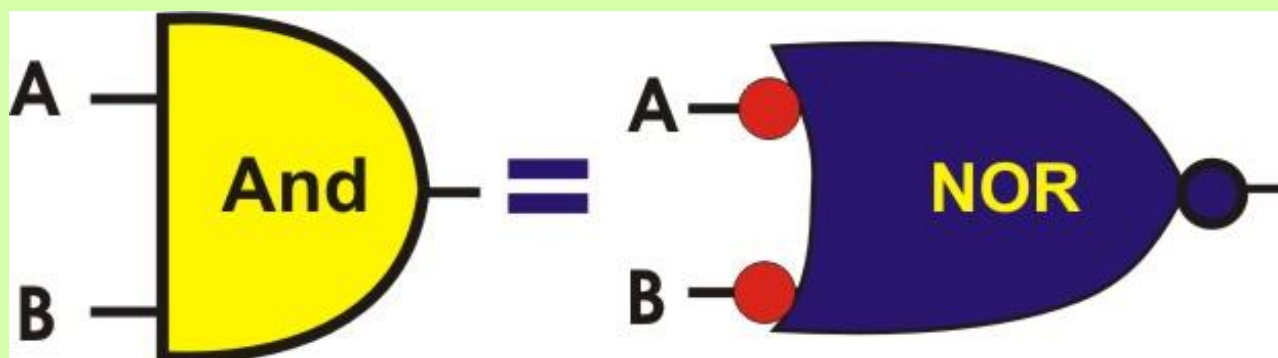


Por medio de este teorema se obtiene el
Equivalente
De una función Booleana

El teorema de De Morgan permite
simplificar expresiones lógicas y diseñar
circuitos digitales equivalentes usando
solo compuertas NAND o NOR

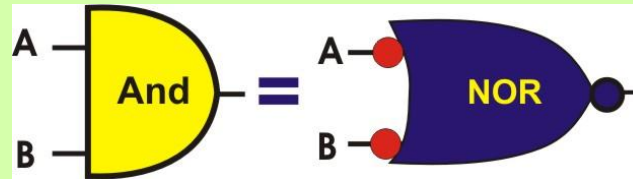
Teorema de D'Morgan

AND  **NOR**
Negando las entradas



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

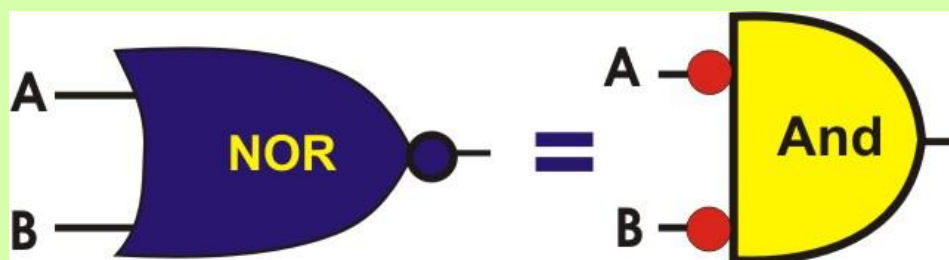
Teorema de D'Morgan



m	A	B	$A \cdot B$	$\overline{\overline{A} + \overline{B}}$
0	0	0	0	0
1	0	1	0	0
2	1	0	0	0
3	1	1	1	1

Teorema de D'Morgan

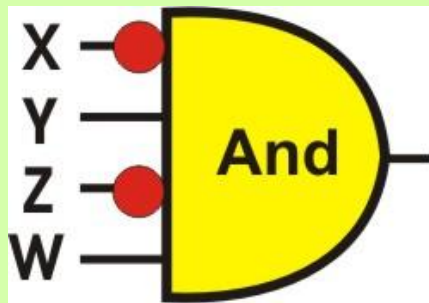
NOR  **AND**
Negando las entradas



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

Teorema de D'Morgan

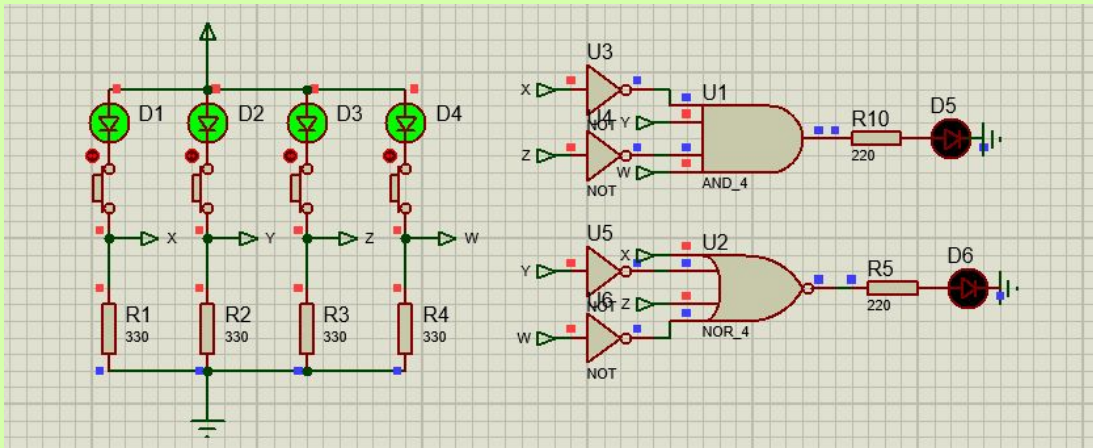
AND $\longleftrightarrow$ **NOR**
Negando las entradas



$X' Y Z' W$

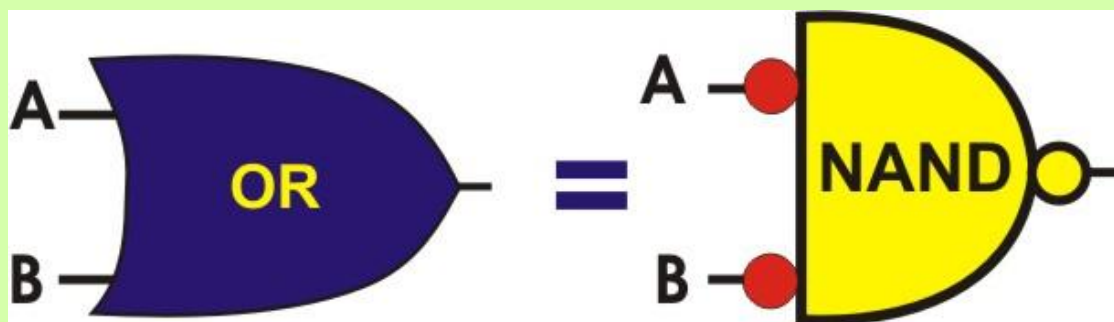


$X + Y' + Z + W'$



Teorema de D'Morgan

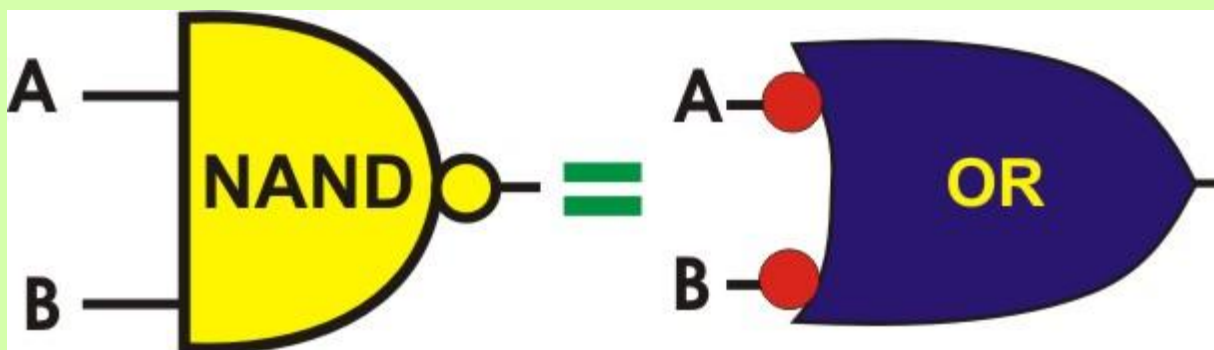
OR $\rightarrow$ NAND
Negando las entradas



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

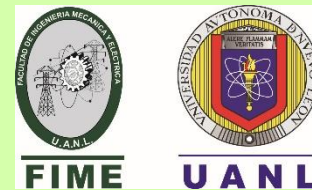
Teorema de D'Morgan

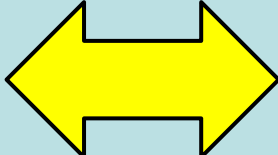
NAND → **OR**
Negando las entradas



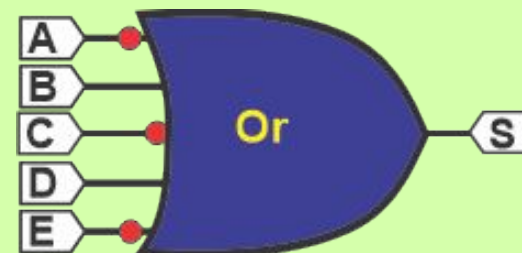
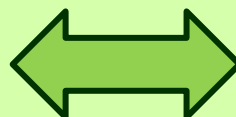
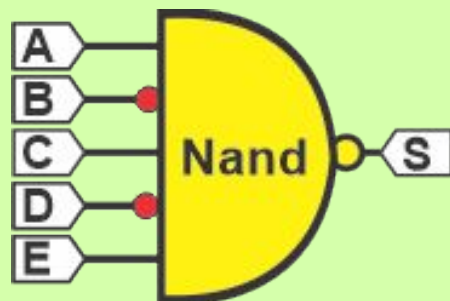
$$\overline{A \cdot B} = \overline{A} + \overline{B}$$

Teorema de D'Morgan



OR  **NAND**

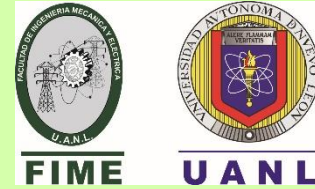
Negando las entradas

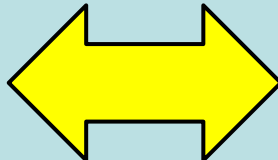


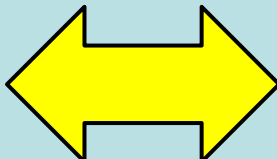
$$(A B' C D' E)' =$$

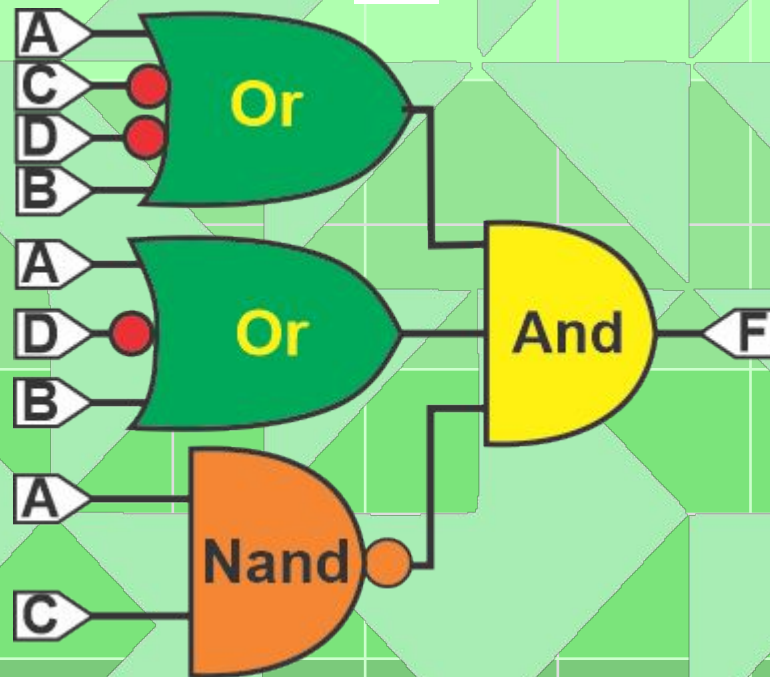
$$A' + B + C' + D + E'$$

Teorema de D'Morgan



AND  **NOR**
D Negando las entradas

OR  **NAND**
Negando las entradas



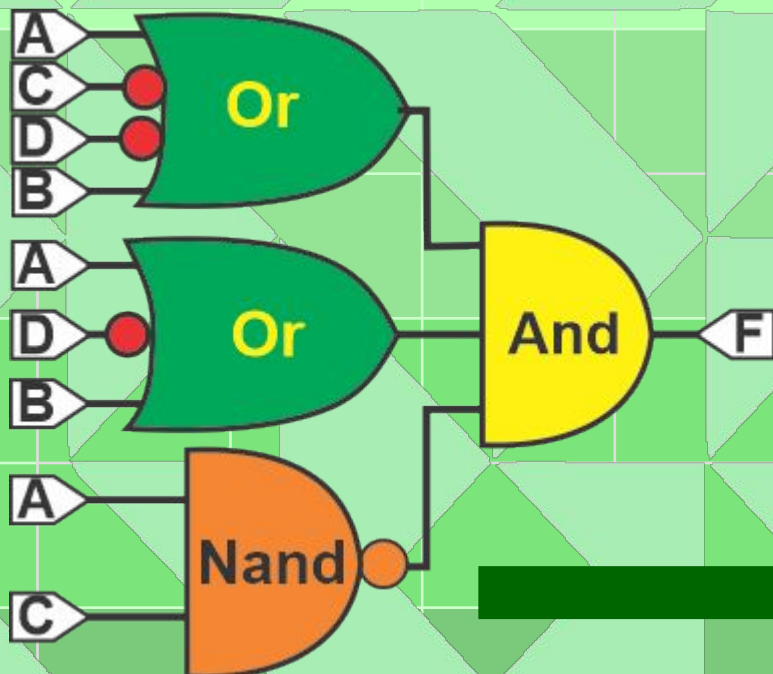
$$F_{(A,B,C,D)} = (A + \textcolor{red}{C}' + \textcolor{red}{D}' + B)(A + \textcolor{red}{D}' + B)(\textcolor{red}{A} \textcolor{red}{C})'$$

NAND ↔ OR

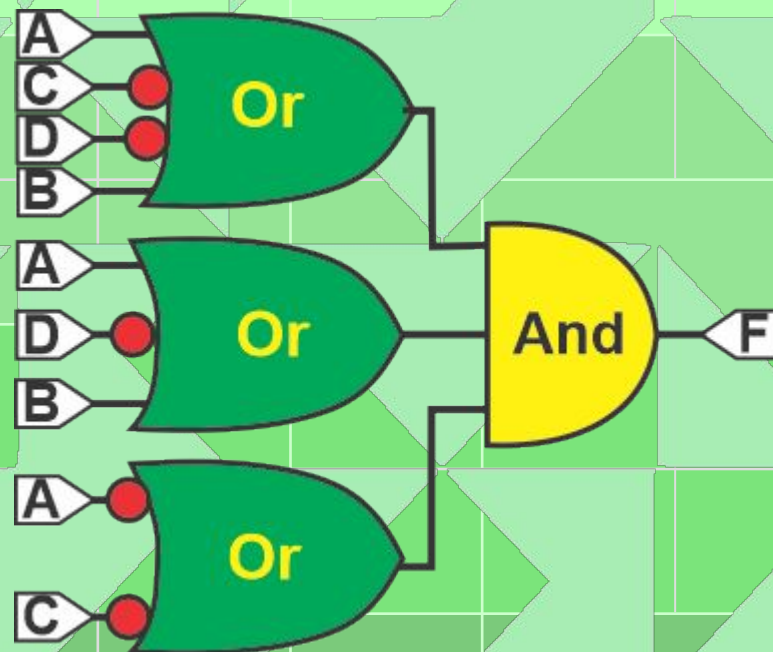
Negando las entradas

?

Or/And



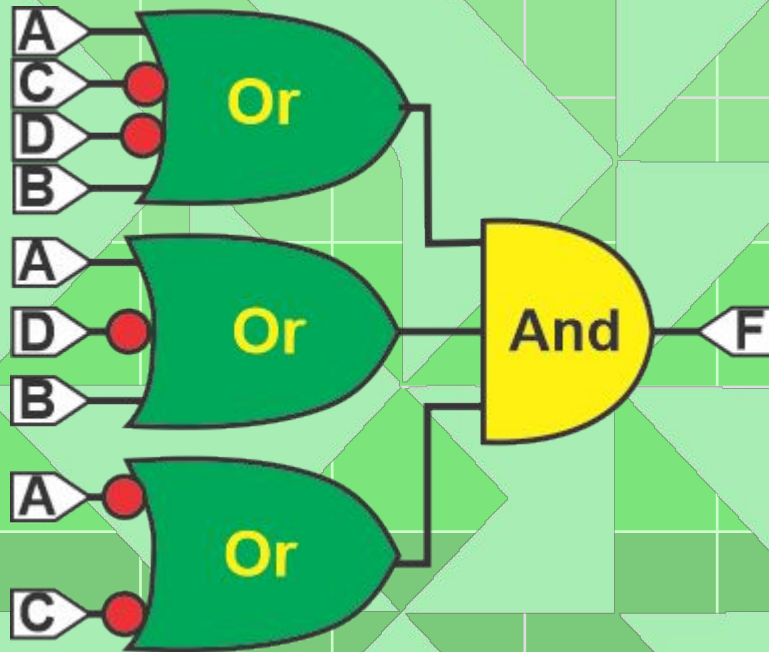
$$F_{(A,B,C,D)} = (A + \mathbf{C'} + \mathbf{D'} + B)(A + \mathbf{D'} + B)(\mathbf{A C'})'$$



$$F_{(A,B,C,D)} = (A + \mathbf{C'} + \mathbf{D'} + B)(A + \mathbf{D'} + B)(\mathbf{A'} + \mathbf{C'})$$

Forma POS

Or/And



$$F_{(A,B,C,D)} = (A + C' + D' + B)(A + D' + B)(A' + C')$$

Forma POS

m	A	B	C	D	F
0	0	0	0	0	1
1	0	0	0	1	0
2	0	0	1	0	1
3	0	0	1	1	0
4	0	1	0	0	1
5	0	1	0	1	1
6	0	1	1	0	1
7	0	1	1	1	1
8	1	0	0	0	1
9	1	0	0	1	1
10	1	0	1	0	0
11	1	0	1	1	0
12	1	1	0	0	1
13	1	1	0	1	1
14	1	1	1	0	0
15	1	1	1	1	0

m	A	B	C	D	F
0	0	0	0	0	1
1	0	0	0	1	0
2	0	0	1	0	1
3	0	0	1	1	0
4	0	1	0	0	1
5	0	1	0	1	1
6	0	1	1	0	1
7	0	1	1	1	1
8	1	0	0	0	1
9	1	0	0	1	1
10	1	0	1	0	0
11	1	0	1	1	0
12	1	1	0	0	1
13	1	1	0	1	1
14	1	1	1	0	0
15	1	1	1	1	0

Formas Canónicas

$$F_{(A,B,C,D)} = \sum m(0, 2, 4, 5, 6, 7, 8, 9, 12, 14)$$

$$F_{(A,B,C,D)} = \prod m(1, 3, 10, 11, 14, 15)$$

$$F_{(A,B,C,D)} = \prod M(1, 3, 10, 11, 14, 15)$$

INPUT TERMS FORMAT

Number of Variables: OK

Number of Functions: Cancel

Variable and Function Names:

☐ Use Default Names (Do Not Enter) ☐ Use Current Names

☒ Enter Names

Input Format:

☒ Dec ☐ Hexadecimal

☐ Binary (0,1,-1) ☐ Octal

INPUT VARIABLE NAMES:

OUTPUT FUNCTION NAMES:

Select Range: ☒ 1-16 ☐ 17-32

OK Cancel

Terms2

Input Variable Names: A B C D

Output Function Names: F

,1 3 10 11 14 15.

SIMPLIFICATION OPTIONS:

Output Format:

☒ Alphanumeric Form ☐ Binary Form ☐ PLA Table

☒ Sum-of-Products Form ☐ Product-of-Sums Form

Simplification Routine

- ☒ Pi Chart -- One Irredundant Solution
☐ Petrick -- All Minimal Solutions
☐ Petrick -- Maximum of 5 Minimal Solutions
☐ Espresso II -- multiple-output minimization
☐ Espresso II -- single-output minimization

Options

- ☐ Complemented Functions
☐ Display All Prime Implicants
☐ Identify Essential Prime Implicants
☒ Display Input and Gate Costs
☐ Overwrite the Output Window

OK

Cancel

Simplification Routine: PI Chart

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

Input Cost = 9

Gate Cost = 4

Input Cost=9, Gate Cost=4

SIMPLIFICATION OPTIONS:

Output Format:

☒ Alphanumeric Form ☐ Binary Form ☐ PLA Table

☐ Sum-of-Products Form ☒ Product-of-Sums Form

Simplification Routine

- ☒ Pi Chart -- One Irredundant Solution
☐ Petrick -- All Minimal Solutions
☐ Petrick -- Maximum of 5 Minimal Solutions
☐ Espresso II -- multiple-output minimization
☐ Espresso II -- single-output minimization

Options

- ☐ Complemented Functions
☐ Display All Prime Implicants
☐ Identify Essential Prime Implicants
☐ Display Input and Gate Costs
☐ Overwrite the Output Window

OK

Cancel

Simplification Routine: PI Chart

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

Input Cost = 7

Gate Cost = 3

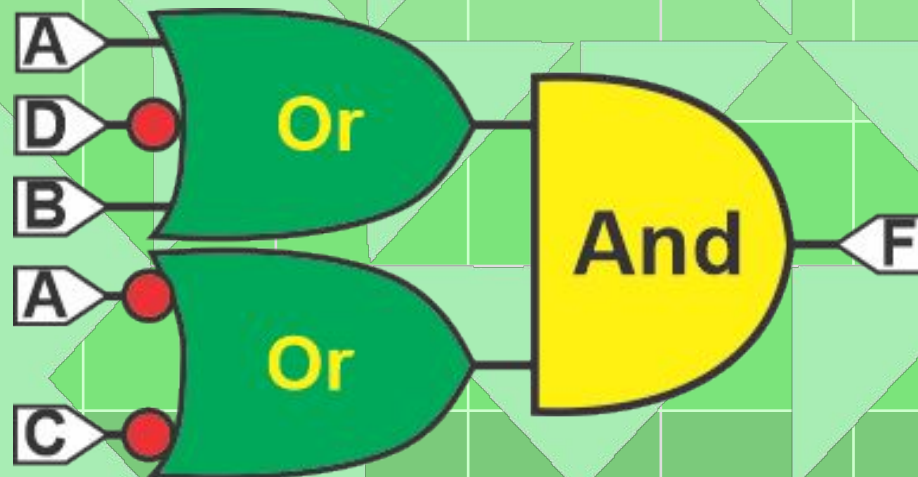
Input Cost=7, Gate Cost=3

Simplification Routine: PI Chart

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

Input Cost = 7

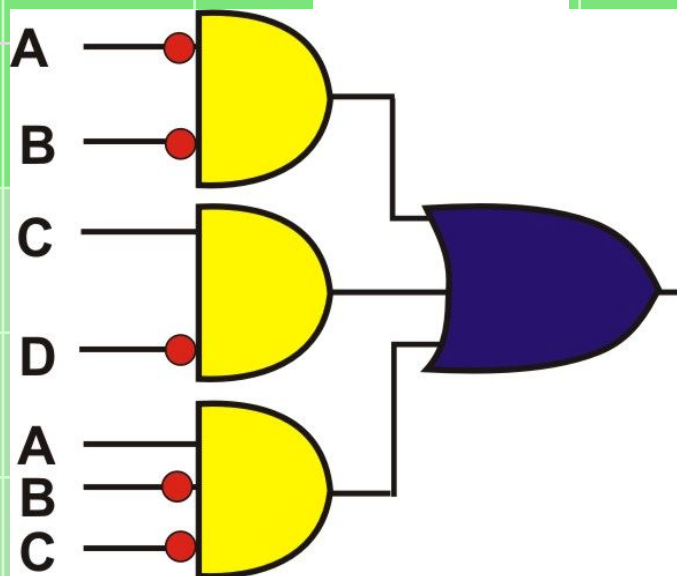
Gate Cost = 3



Obtenga el circuito equivalente aplicando el teorema de D'Morgan y compare el resultado usando la tabla de verdad

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

And/Or



A $\leftrightarrow$ NO

N $\leftrightarrow$ Negando las entradas

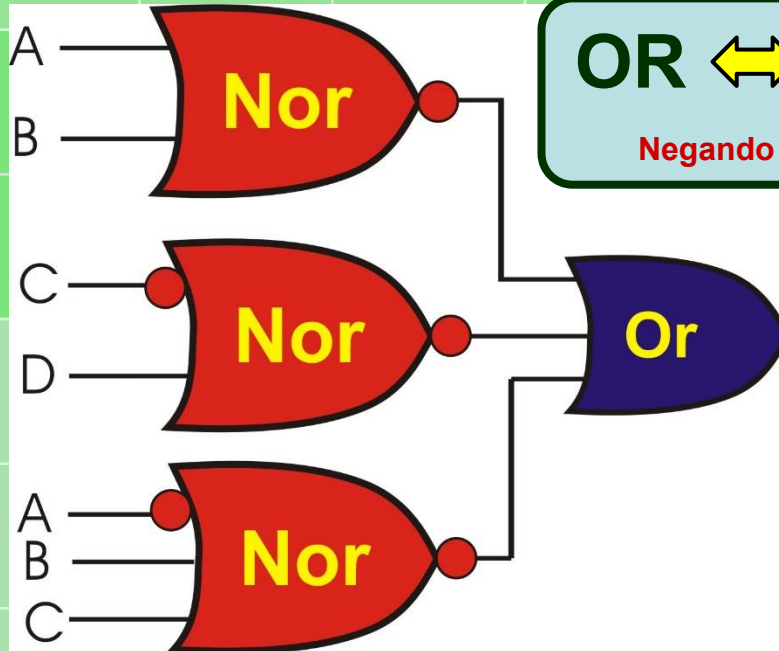
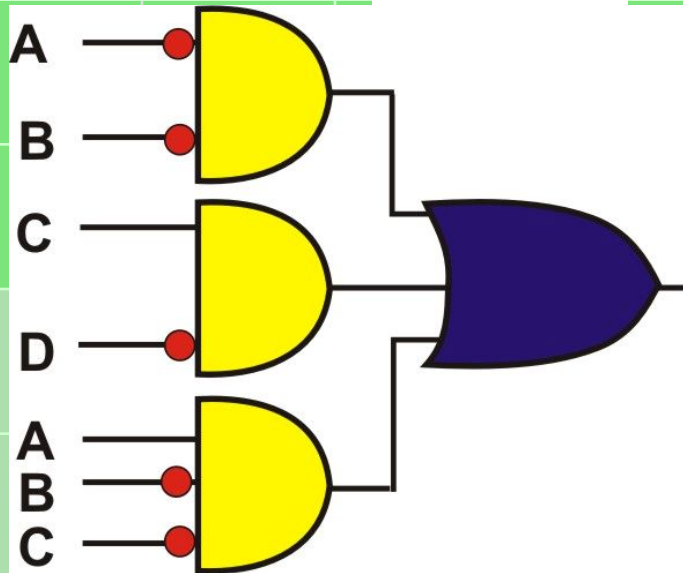
OR $\leftrightarrow$ NAND

Negando las entradas

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

$$F_{(A,B,C,D)} = (A + B)' + (C' + D)' + (A' + B + C)'$$

And/Or



Nor/Or

A $\leftrightarrow$ NO
N $\leftrightarrow$ Negando las entradas

OR $\leftrightarrow$ NAND
Negando las entradas

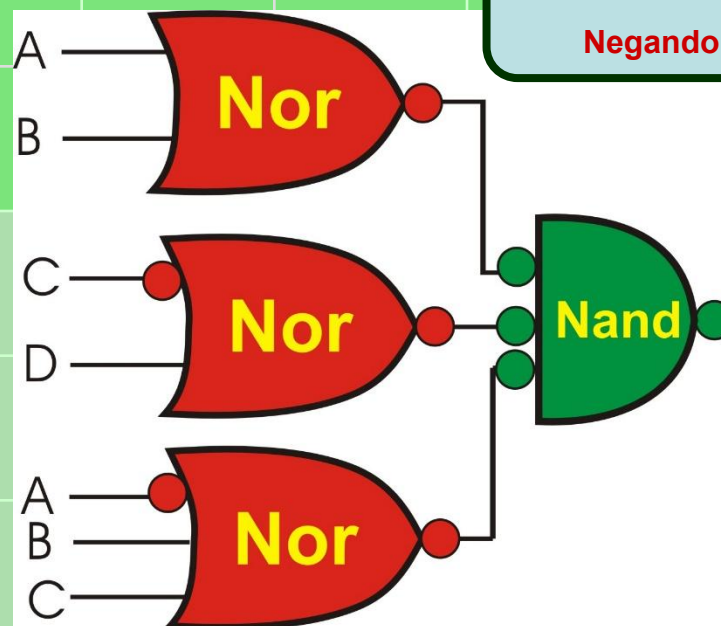
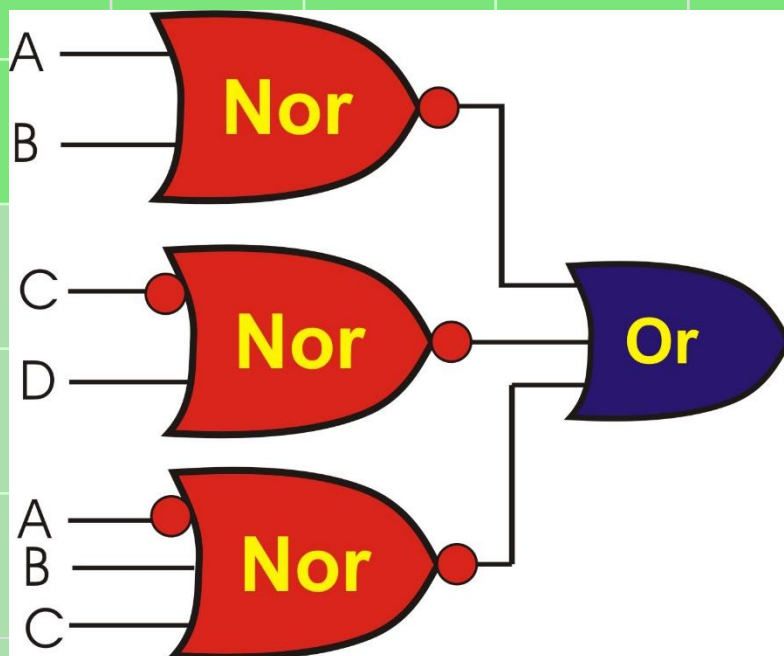
$$F_{(A,B,C,D)} = \bar{A}\bar{B} + C\bar{D} + A\bar{B}\bar{C}$$

$$F_{(A,B,C,D)} = [(A + B)'' + (C' + D)'' + (A' + B + C)'']'$$

$$F_{(A,B,C,D)} = [(A + B) (C' + D) (A' + B + C)]'$$

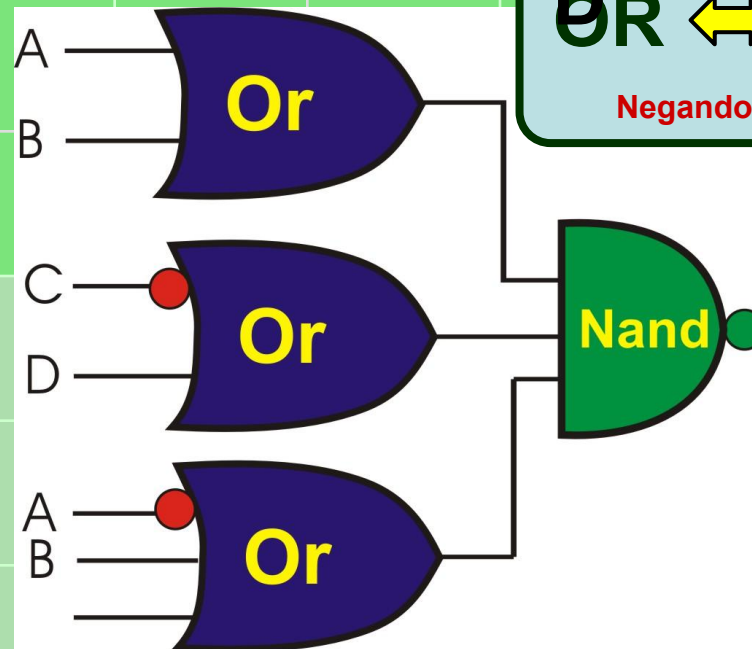
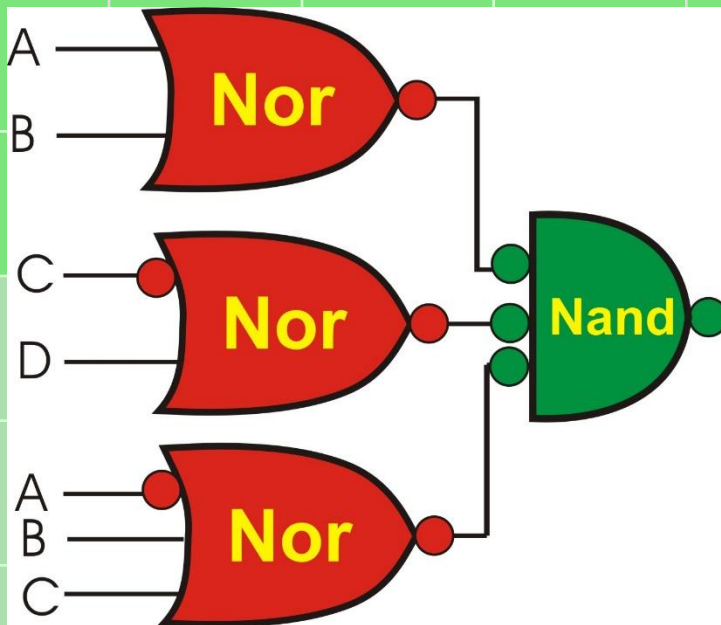
A $\leftrightarrow$ NO
N Negando las entradas R

OR $\leftrightarrow$ NAND
Negando las entradas



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

$$F_{(A,B,C,D)} = [(A + B) (C' + D) (A' + B + C)]'$$



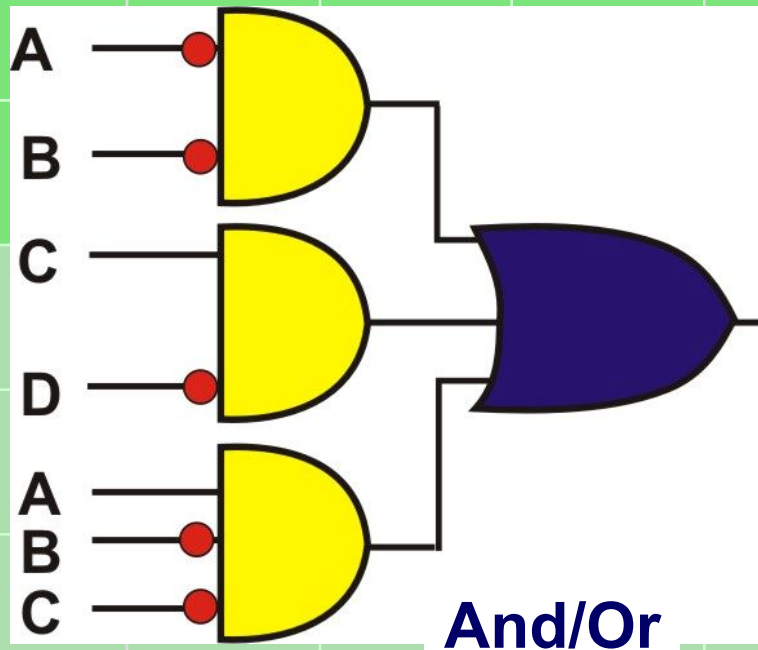
Or/Nand

A $\leftrightarrow$ NO
N $\leftrightarrow$ Negando las entradas

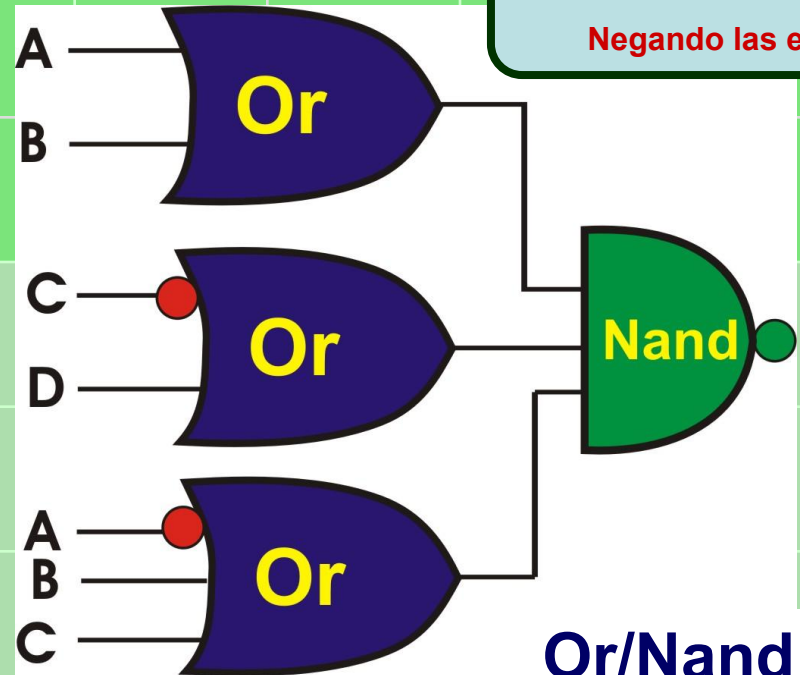
B $\leftrightarrow$ NAND
R $\leftrightarrow$ Negando las entradas

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

$$F_{(A,B,C,D)} = [(A + B) (C' + D) (A' + B + C)]'$$



And/Or



Or/Nand

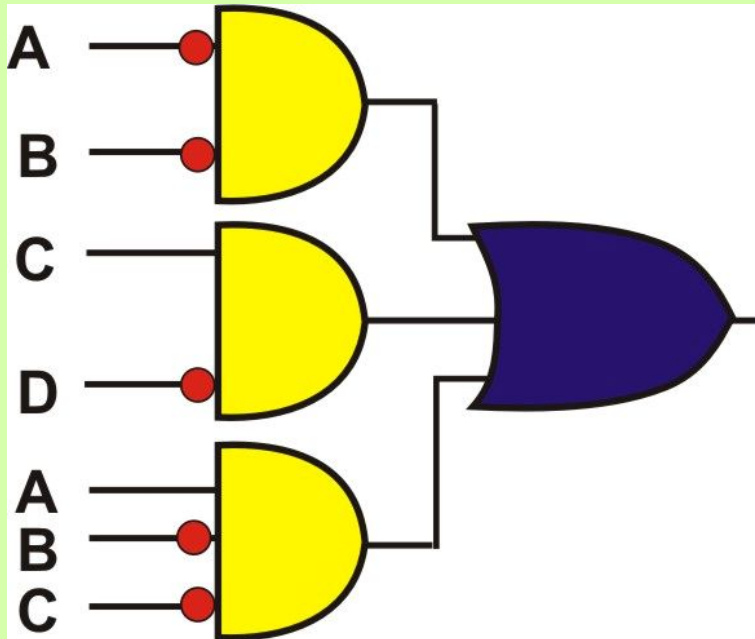
A $\leftrightarrow$ NO

N $\leftrightarrow$ Negando las entradas

OR $\leftrightarrow$ NAND

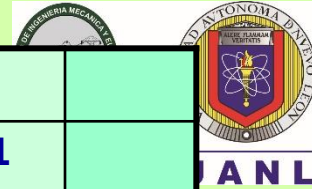
Negando las entradas

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



And/Or

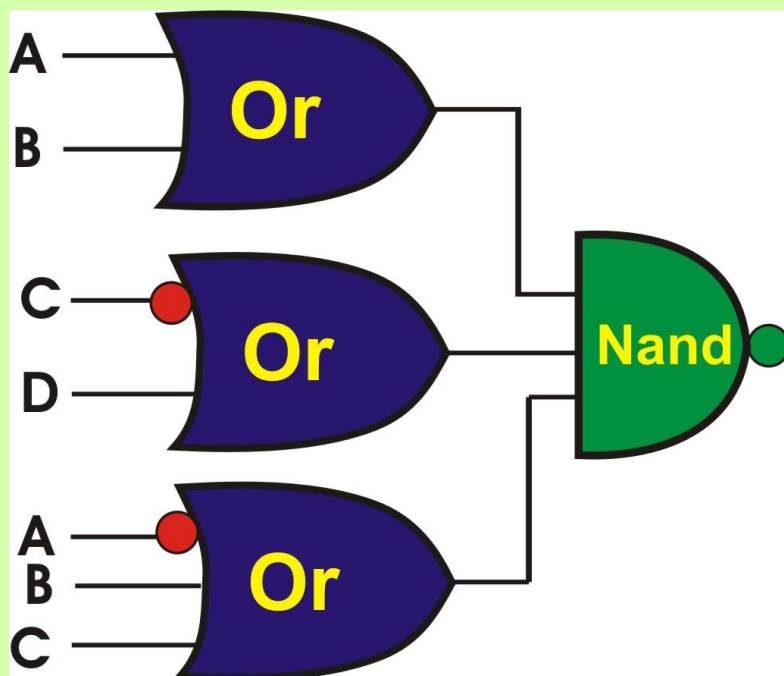
m	A	B	C	D		
0	0	0	0	0	1	
1	0	0	0	1	1	
2	0	0	1	0	1	
3	0	0	1	1	1	
4	0	1	0	0	0	
5	0	1	0	1	0	
6	0	1	1	0	1	
7	0	1	1	1	0	
8	1	0	0	0	1	
9	1	0	0	1	1	
10	1	0	1	0	1	
11	1	0	1	1	0	
12	1	1	0	0	0	
13	1	1	0	1	0	
14	1	1	1	0	1	
15	1	1	1	1	0	





ANL

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



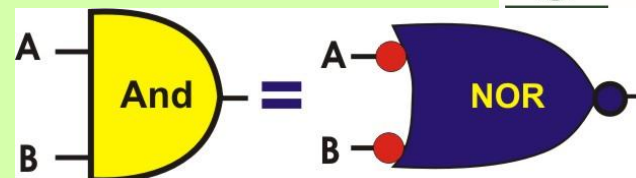
Or/Nand

m	A	B	C	D		
0	0	0	0	0	1	1
1	0	0	0	1	1	1
2	0	0	1	0	1	1
3	0	0	1	1	1	1
4	0	1	0	0	0	0
5	0	1	0	1	0	0
6	0	1	1	0	1	1
7	0	1	1	1	0	0
8	1	0	0	0	1	1
9	1	0	0	1	1	1
10	1	0	1	0	1	1
11	1	0	1	1	0	0
12	1	1	0	0	0	0
13	1	1	0	1	0	0
14	1	1	1	0	1	1
15	1	1	1	1	0	0

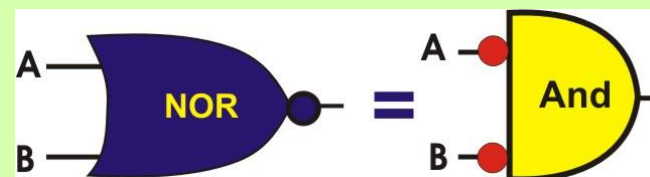


Teorema de D'Morgan

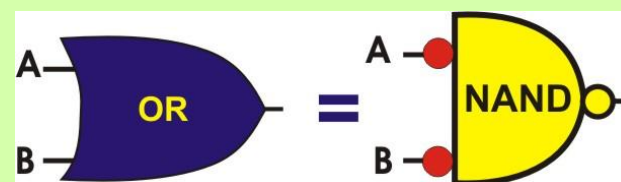
AND $\rightarrow$ NOR
Negando las entradas



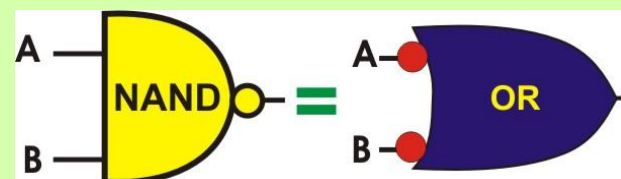
NOR $\rightarrow$ AND
Negando las entradas

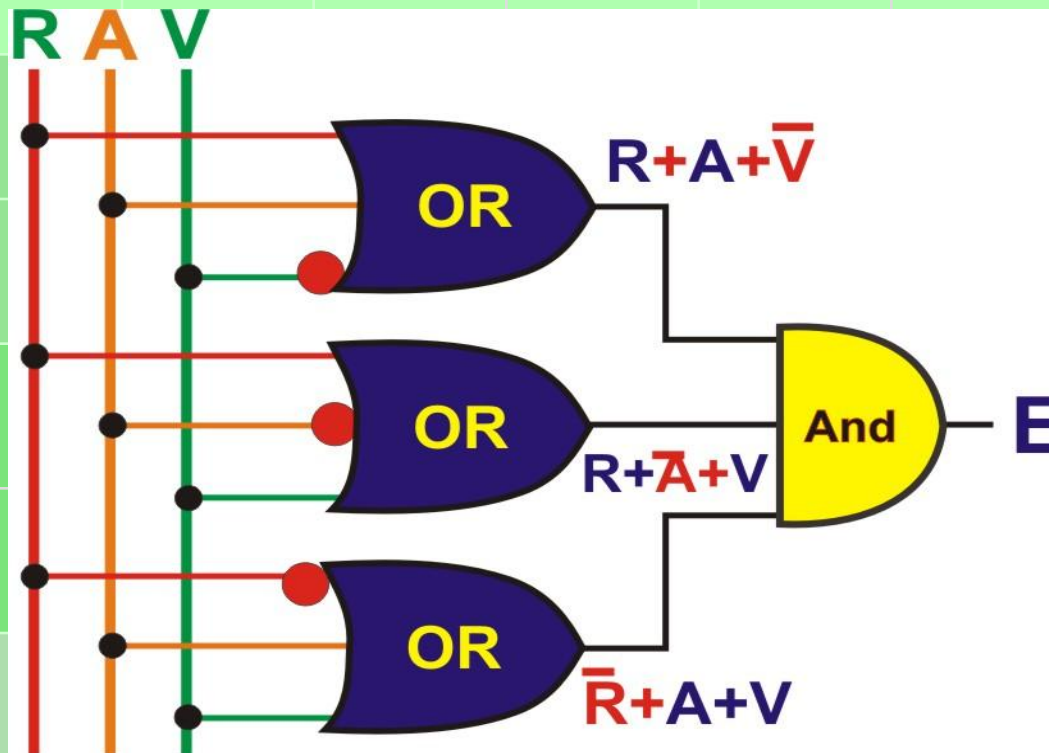


OR $\rightarrow$ NAND
Negando las entradas

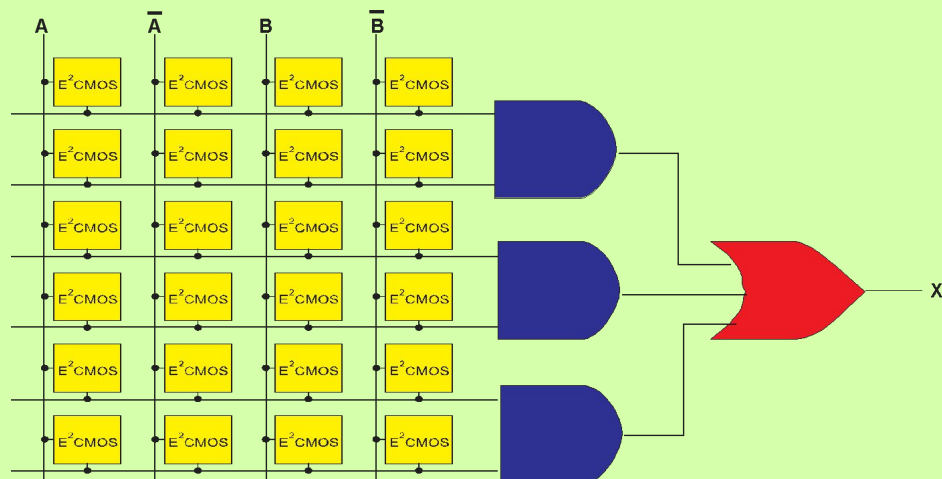
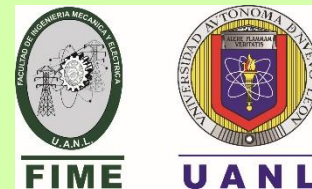


NAND $\rightarrow$ OR
Negando las entradas

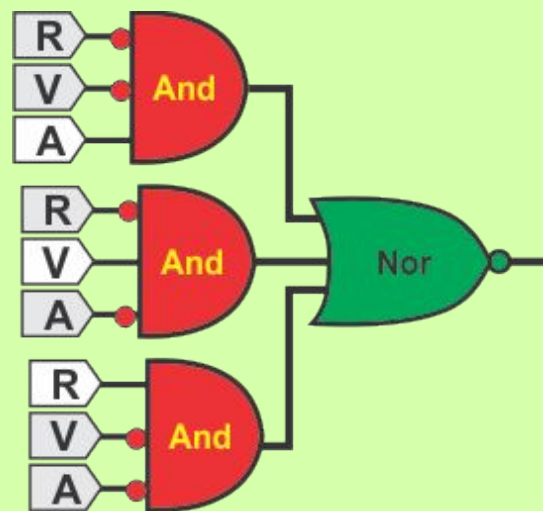
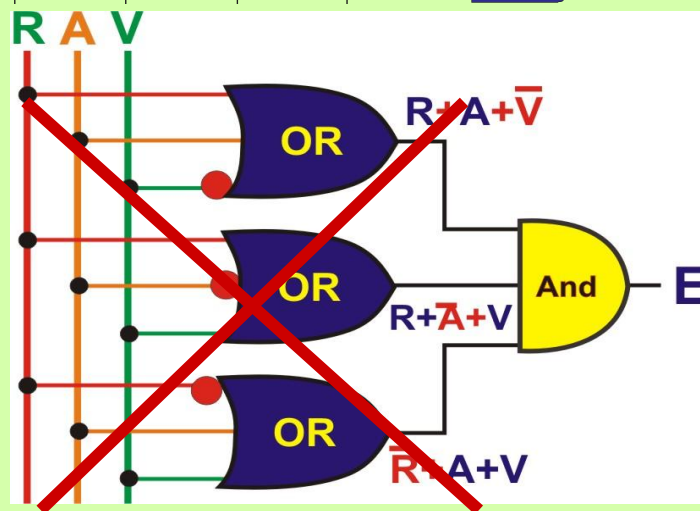




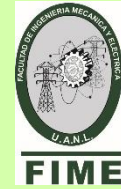
$$FE_{(R,A,V)} = (R+A+\overline{V}) (R+\overline{A}+V) (\overline{R}+A+V)$$



Estructura interna de un GAL
And/Or And/Nor

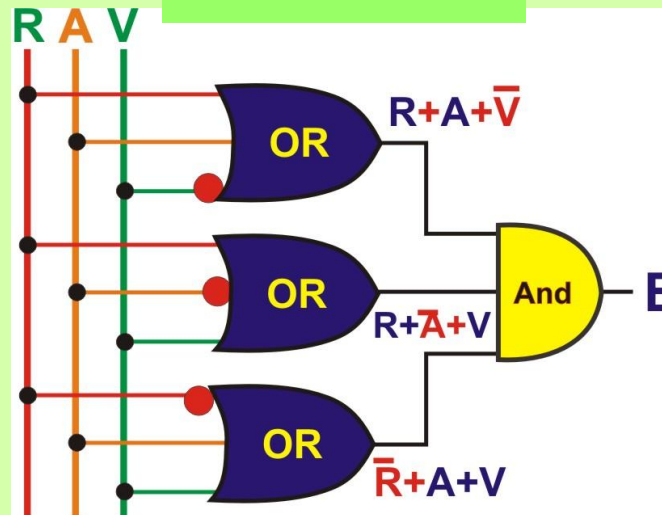


$$FE_{(R,A,V)} = (R+A+V') (R+A'+V) (R'+A+V)$$



$$F_{E_{(R,A,V)}} = (R + A + \bar{V}) (R + \bar{A} + V) (\bar{R} + A + V)$$

Or/And



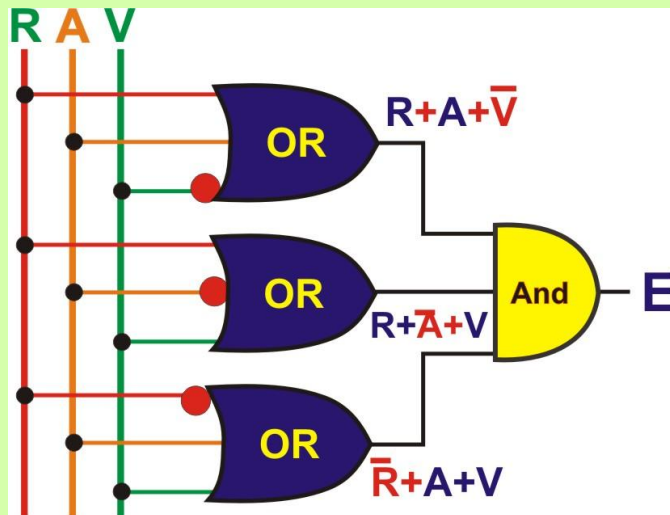
$$\text{ERROR} = \neg(\neg R \ \& \ \neg V \ \& \ A \ \# \neg R \ \& \ V \ \& \ \neg A \ \# R \ \& \ \neg V \ \& \ \neg A);$$

$$\text{ERROR} = (R' \ V' \ A + R' \ V \ A' + R \ V' \ A')'$$

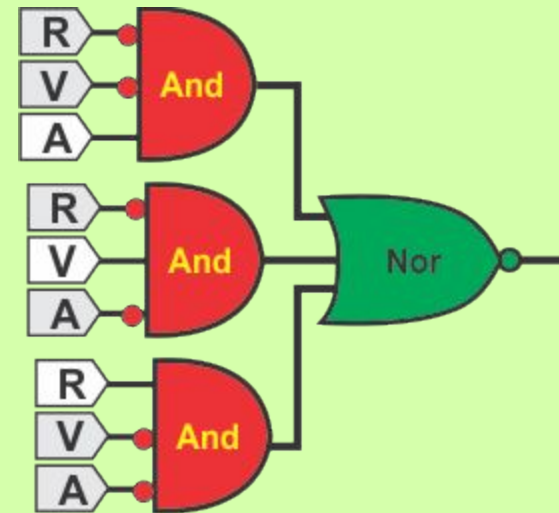
Or
& AND
! Not

$$FE_{(R,A,V)} = (R+A+\bar{V}) (R+\bar{A}+V) (\bar{R}+A+V)$$

Or/And



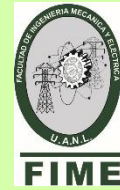
And/Nor



$$\text{ERROR} = (R' V' A + R' V A' + R V' A')'$$

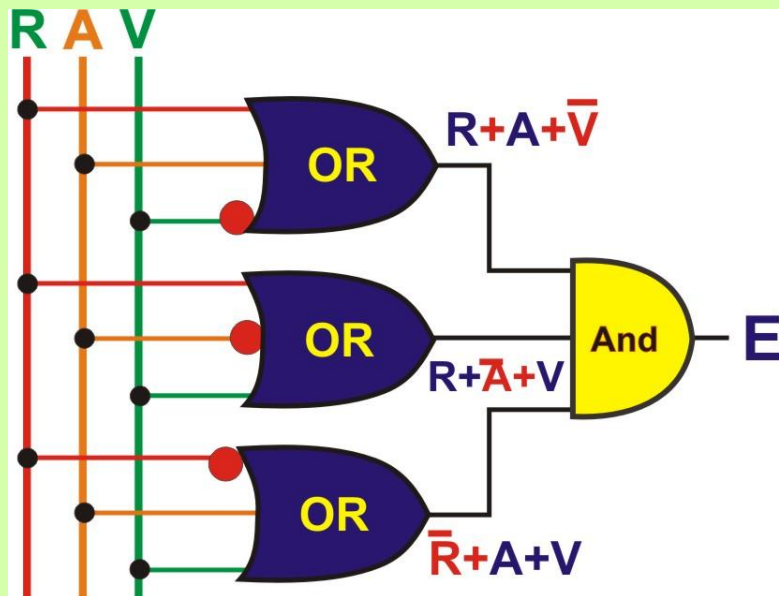
$$\text{ERROR} = !(R \& V \& A \# R \& V \& A \# R \& V \& A);$$

Or
& And
! Not

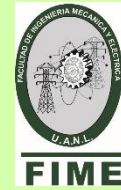
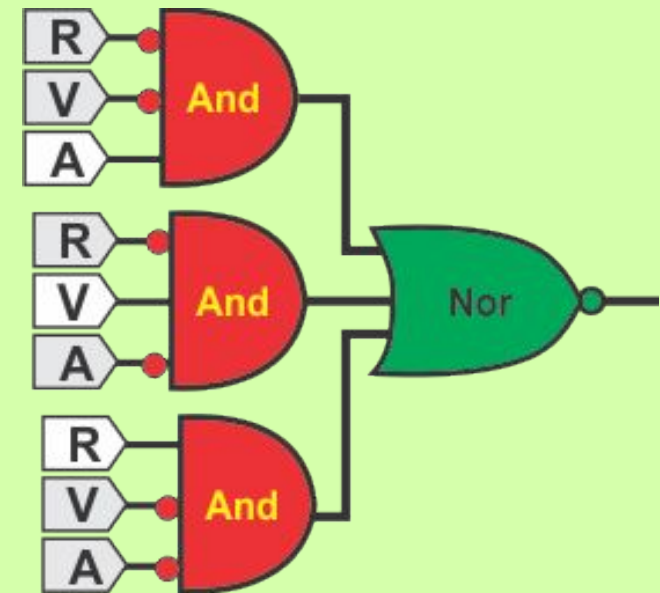


$$F_{E(R,A,V)} = (R+A+\bar{V}) (R+\bar{A}+V) (\bar{R}+A+V)$$

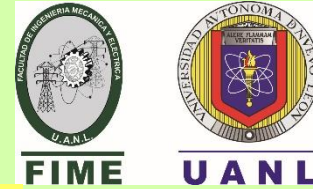
Or/And



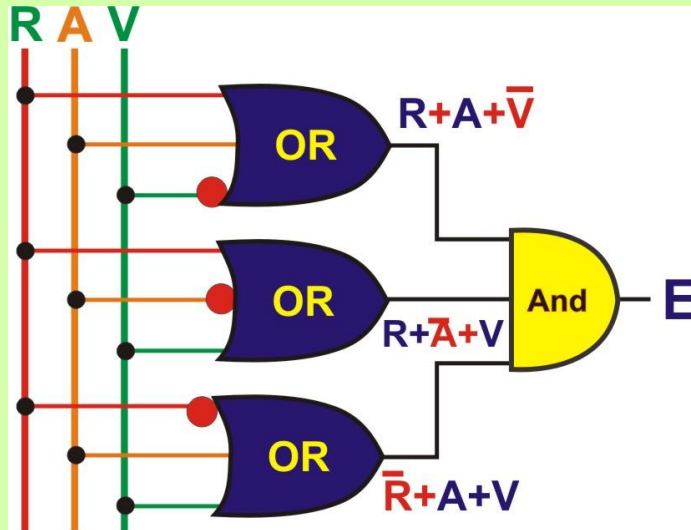
And/Nor



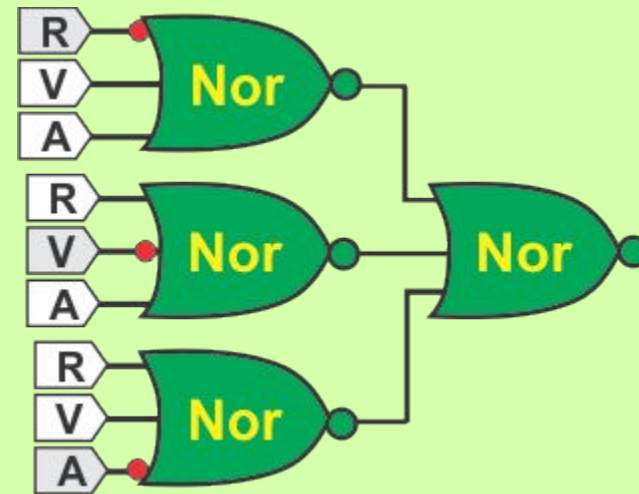
$$F_{E(R,A,V)} = (R+A+\bar{V}) (R+\bar{A}+V) (\bar{R}+A+V)$$



Or/And

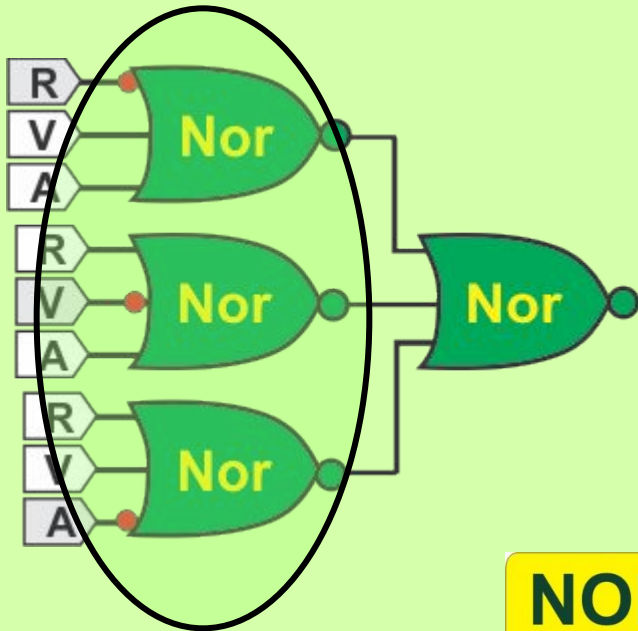


Nor/Nor



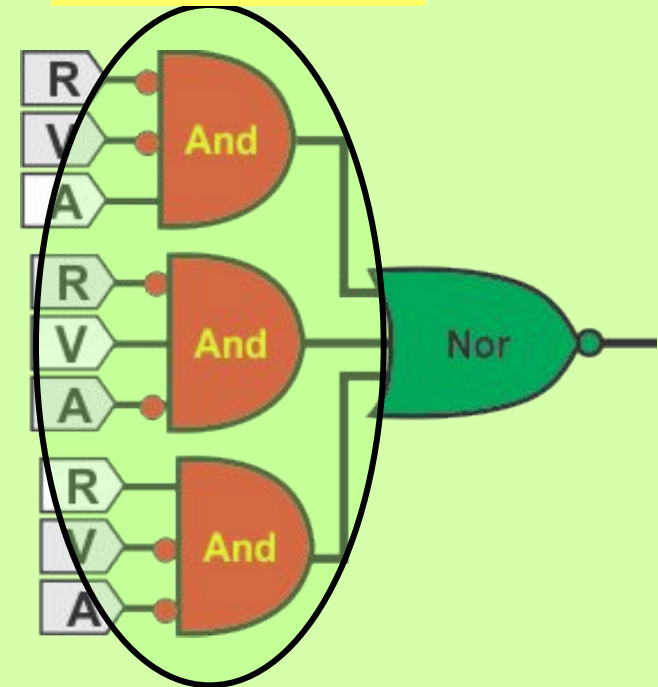
AND → NOR
Negando las entradas

Nor/Nor

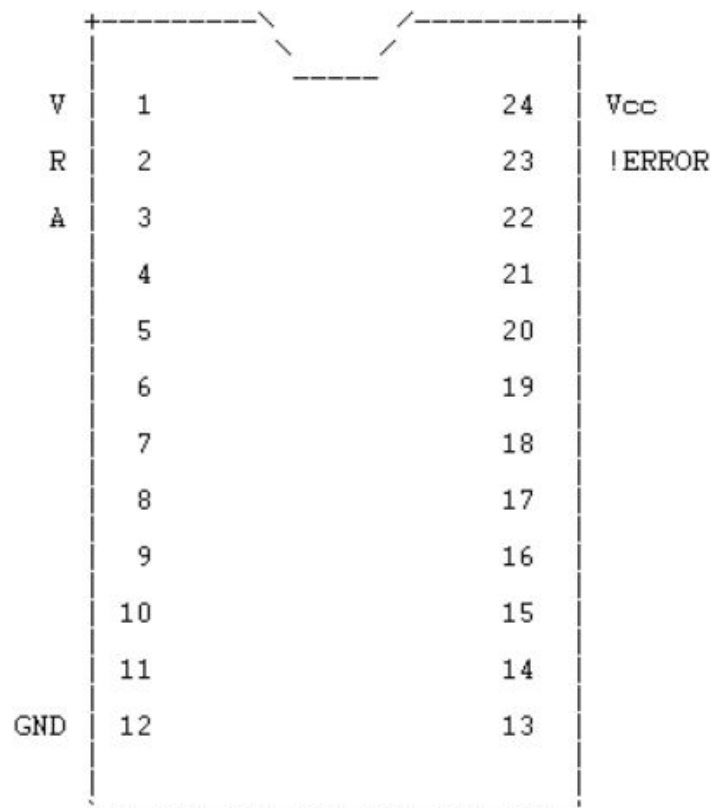
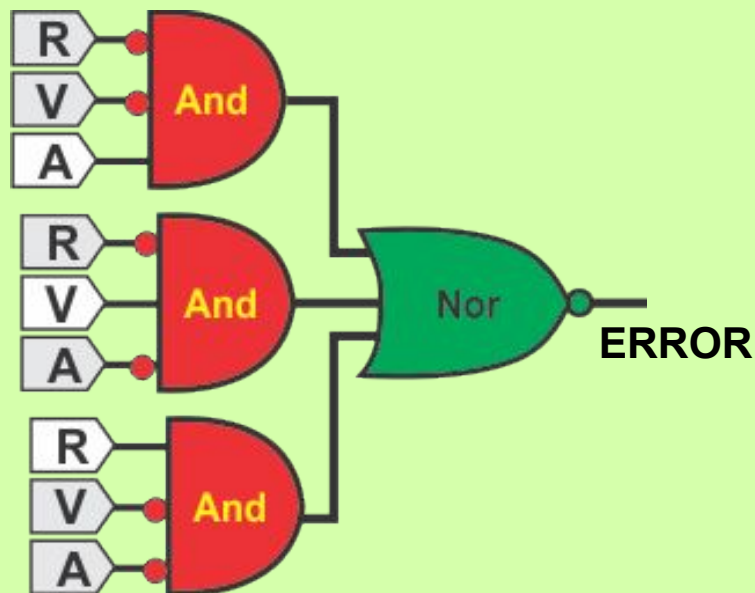
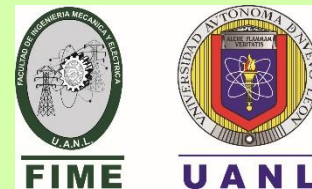


NOR → AND
Negando las entradas

And/Nor



And/Nor



ERROR = $\neg(\neg V \ \& \ \neg R \ \& \ A \ \# \ \neg V \ \& \ R \ \& \ \neg A \ \# \ V \ \& \ \neg R \ \& \ \neg A)$;

Las Ocho Formas estándar

Diferentes formas para expresar algebraicamente una función booleana Partiendo de:

a).- Suma de Productos SOP (And/OR) partiendo de los Minitérminos

b).-Productos de Sumas POS (Or/And) partiendo de los Maxitérminos

Para obtener las ocho formas estándar es necesario aplicar el Teorema de D' Morgan

	SOP			POS
1	And/Or		5	Or/And
2	Nor/Or		6	Nand/And
3	Or/Nand		7	And/Nor
4	Nand/Nand		8	Nor/Nor