ALGEBRA DE BOOLE

ALGEBRA DE BOOLE Y FUNCIONES LÓGICAS

Introducción

Álgebra de *Boole*

Análisis booleano de circuitos lógicos. Tablas de verdad

Algoritmos de simplificación de expresiones lógicas

Implementación de funciones lógicas mediante puertas lógicas

- La **Electrónica Digital** -+ la realización electrónica (implementación) de funciones booleanas
  - En esta lección vamos a plantear el formalismo asociado al **Álgebra de Boole**.

- PASOS:
  - Inicialmente estableceremos las bases matemáticas asociadas al **Álgebra de Boole**.
  - Después analizaremos la representación de las variables lógicas por magnitudes físicas, indicando los módulos mínimos para la síntesis de funciones.
  - Estudiaremos algún método para simplificar en alguna forma las funciones booleanas.
  - Por último, se realiza su implementación en circuitos.

- El **álgebra booleana** son reglas algebraicas, basadas en la teoría de conjuntos, para manejar ecuaciones de lógica matemática.
  - La lógica matemática trata con proposiciones, elementos de circuitos de dos estados, etc., asociados por medio de operadores como Y, O, NO, EXCEPTO, SI...
  - Permite cálculos y demostraciones como cualquier parte de las matemáticas.
  - Es llamada así en honor del matemático **George Boole**, que la introdujo en 1847.

- **DEFINICION de Algebra boole :**
- Se dice que un conjunto de elementos ***B***, en el que existen definidas dos operaciones binarias (que representaremos por + y por $\cdot$) tiene estructura de **Álgebra de Boole** si y solo si se cumplen los siguientes cuatro postulados:

  - Las operaciones + y $\cdot$ son conmutativas.

    Ejemplo: $a + b = b + a$ y $a \cdot b = b \cdot a$

  - Existen en ***B*** dos elementos neutros, que denotaremos por **0** y **1**, para las operaciones + y $\cdot$, respectivamente.

    $a + 0 = a$ y $a \cdot 1 = a$

  - Cada operación es distributiva con respecto a la otra (expresa el proceso de sacar factor común).

    Ejemplo (tres variables): $a(b+c) = ab + ac$

  - Para cada elemento a de ***B*** existe un $\bar{a}$ tal que: $a + \bar{a} = 1$ y $a \cdot \bar{a} = 0$

- En Electrónica Digital estamos interesados en el **Álgebra de Boole** establecida en el conjunto $B = \{0,1\}$ con las operaciones + y $\cdot$ definidas por:

+	**0**	**1**
0	0	1
1	1	1

$\cdot$	**0**	**1**
0	0	0
1	0	1

- Estas operaciones se cumplen los 4 postulados anteriores.Llamaremos:

AND (**Y lógico**) a la operación $\cdot$
OR (**O lógico**) a la operación **+**
NOT (**NO lógico**) a la operación de complementación.

NOTA; Por simplicidad, de ahora en adelante usaremos la representación xy en vez de la $x \cdot y$ para la composición de las variables x e y mediante la operación $\cdot$.

- *La Electrónica Digital:* ***estudio*** *y realización de circuitos que realicen las funciones AND, OR, NOT y sus combinaciones*.

- Existen una serie de **teoremas**, válidos en cualquier **álgebra de Boole**, que vamos a enunciar y que no demostraremos, los cuales nos serán de gran utilidad para la simplificación de funciones:

  - Teorema 1. **Principio de dualidad:** Cada proposición o identidad algebraica deducible de los postulados del Álgebra de Boole permanece válida si:
    - cambiamos entre si las operaciones + y ·,
    - y también cambiamos entre si los elementos neutros **0** y **1**.
  - Teorema2. $\boxed{x + x = x}$ $\boxed{xx = x}$
  - Teorema3. $\boxed{x + 1 = 1}$ $\boxed{x0 = 0}$
  - Teorema 4. **Ley de absorción** $\boxed{x + xy = x}$ $\boxed{x(x + y) = x}$
  - Teorema5. $\boxed{x + (y + z) = (x + y) + z}$ $\boxed{x(yz) = (xy)z}$

    Asociatividad de las operaciones + y ·

- Teorema 6. El elemento $\boxed{\overline{x}}$ asociado a $\boxed{x}$ es único
- Teorema7. $\boxed{\overline{(\overline{x})} = x}$
- Teorema8. **Teorema de De Morgan**: $\boxed{\overline{x\,y} = \overline{x} + \overline{y}}$ $\boxed{\overline{x + y} = \overline{x}\,\overline{y}}$
- Teorema9. $\boxed{xy + \overline{x}y = y}$ $\boxed{(x + y)(\overline{x} + y) = y}$
- Teorema 10.
  - Si $\boxed{x \subseteq y \text{ e } y \subseteq z \Rightarrow x \subseteq z}$
  - Si $\boxed{x \subseteq y \text{ e } x \subseteq z \Rightarrow x \subseteq yz}$
  - Si $\boxed{x \subseteq y \Rightarrow x \subseteq y + z}$ para cualquier z $\boxed{x \subseteq y}$ si y solo si $\boxed{\overline{y} \subseteq \overline{x}}$

- DEFINICION: Llamaremos **funciones lógicas** o **funciones booleanas** (*f*) a todo conjunto de variables relacionadas entre sí por una expresión que representa:
  - La combinación de un conjunto finito de símbolos, representando constantes o variables
  - Unidos por las operaciones *AND (producto lógico)* , *OR (suma lógica)* o *NOT (complementación)*.

- **Término producto:** es una expresión lógica que consiste en un conjunto de variables (o sus complementadas) unidas por la operación AND.

$$f(x, y, z) = \overline{x}\, y$$

- **Término suma:** es una expresión lógica que consiste en un conjunto de variables (o sus complementadas) unidas por la operación OR.

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

- **Un término producto standard o MINTERM:** expresión lógica que consiste en un conjunto de TODAS las variables (o sus complementadas) unidas por la operación AND.

$$f(x, y, z) = \overline{x}\, y\, z$$

- **Un término suma standard o MAXTERM:** expresión lógica que consiste en un conjunto de TODAS las variables (o sus complementadas) unidas por la operación OR.

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

- **Con n variables** se pueden formar **2^n MINTERMS y 2^n MAXTERMS**

- **Formas canónicas de una función**
  - Toda función booleana puede expresarse de dos formas canónicas diferentes:
    - ponerse en forma **de suma de Minterms** → ***forma canónica disyuntiva.***
    - O ponerse en forma de **producto de Maxterms** → ***forma canónica conjuntiva.***

  - Esa función algebraica se podrá simplificar aplicando directamente las leyes del álgebra de Boole, o bien, sistemáticamente, a través de métodos de reducción que veremos más adelante.

  - EJEMPLO Nº 1: Consideramos la función $\boxed{f(x,y,z) = x(y+z)}$

    - Tenemos una función de 3 variables
    - Vamos a mostrar los 2^3=8 Minterms y los 2^3=8 Maxterms que podemos formar.

    - A los Minterms los vamos a llamar : $\boldsymbol{m_i}$ (i=0 hasta 7), en total 8
    - A los Maxterms los vamos a llamar : $\boldsymbol{M_i}$ (i=0 hasta 7), en total 8

    - Cada Mi cumple -+ $\boxed{M_i = \overline{m_i}}$

- EJEMPLO Nº 1: Consideramos la función $f(x, y, z) = x(y + z)$
  - Para los Minterms: Los subíndices i, en decimal, indican en binario la complementación o no complementación de la correspondiente variable:
    - Asignando un “1” para la variable sin complementar
    - Asignando un “0” para la variable complementada.
  - Para los MAXTERMS van al contrario

Minterms	$e_1(x)$	$e_2(y)$	$e_3(z)$	$f(e_1,e_2,e_3)$	Maxterms
$m_0 = \bar{x}\,\bar{y}\,\bar{z}$	0	0	0	0	$M_0 = x + y + z$
$m_1 = \bar{x}\,\bar{y}\,z$	0	0	1	0	$M_1 = x + y + \bar{z}$
$m_2 = \bar{x}\,y\,\bar{z}$	0	1	0	0	$M_2 = x + \bar{y} + z$
$m_3 = \bar{x}\,y\,z$	0	1	1	0	$M_3 = x + \bar{y} + \bar{z}$
$m_4 = x\,\bar{y}\,\bar{z}$	1	0	0	0	$M_4 = \bar{x} + y + z$
$m_5 = x\,\bar{y}\,z$	1	0	1	1	$M_5 = \bar{x} + y + \bar{z}$
$m_6 = x\,y\,\bar{z}$	1	1	0	1	$M_6 = \bar{x} + \bar{y} + z$
$m_7 = x\,y\,z$	1	1	1	1	$M_7 = \bar{x} + \bar{y} + \bar{z}$

D. Pardo, et al. 1999

- Ejemplos:
  - el **minterm** *m6= (f=1)* →corresponde al 110 de la tabla de verdad (asignando y a las variables sin complementar y complementadas, respectivamente).
  - Ejemplo: el **maxterm** *M2= (f=0)* → se asigna al contrario

- Si disponemos de la **tabla de verdad de una función**, es inmediato obtener la **expresión de la función**, ya que
  - Si deseamos obtener la función como **suma de minterms**, podemos obtenerla escribiendo la suma de los minterms asociados a aquellas combinaciones de valores de las variables para las cuales la función vale " **1** ".
    - Esta suma vale uno solamente para los conjuntos de valores de las variables que hacen uno la función.
  - Si deseamos obtenerla como **producto de maxterms**, debemos escribir el producto de los maxterms asociados a aquellas combinaciones de valores de las variables para las cuales la función vale " **0** ".
    - De este producto tenemos la certeza que es cero únicamente para los conjuntos de valores de las variables que hacen cero la función.

$$f(x,y,z) = x(y+z)$$

- *Teniendo en cuenta la función:*
  - *f=1 si se cumple que x=1 y además (y=1 o z=1)*
  - *Podemos expresar la función como:*
    - ***La suma de los MINTERMS para los que la función toma valor "1"***
    - ***Producto de MAXTERMS para los que la función toma valor "0".***

Minterms	$e_1(x)$	$e_2(y)$	$e_3(z)$	$f(e_1,e_2,e_3)$	Maxterms
$m_0 = \bar{x}\ \bar{y}\ \bar{z}$	0	0	0	0	$M_0 = x + y + z$
$m_1 = \bar{x}\ \bar{y}\ z$	0	0	1	0	$M_1 = x + y + \bar{z}$
$m_2 = \bar{x}\ y\ \bar{z}$	0	1	0	0	$M_2 = x + \bar{y} + z$
$m_3 = \bar{x}\ y\ z$	0	1	1	0	$M_3 = x + \bar{y} + \bar{z}$
$m_4 = x\ \bar{y}\ \bar{z}$	1	0	0	0	$M_4 = \bar{x} + y + z$
$m_5 = x\ \bar{y}\ z$	1	0	1	1	$M_5 = \bar{x} + y + \bar{z}$
$m_6 = x\ y\ \bar{z}$	1	1	0	1	$M_6 = \bar{x} + \bar{y} + z$
$m_7 = x\ y\ z$	1	1	1	1	$M_7 = \bar{x} + \bar{y} + \bar{z}$

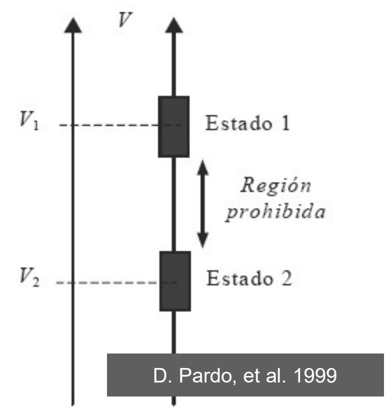
D. Pardo, et al. 1999

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

$$f(x,y,z) = x(y+z) = \sum(5,\ 6,\ 7) = \prod(0,\ 1,\ 2,\ 3,\ 4)$$

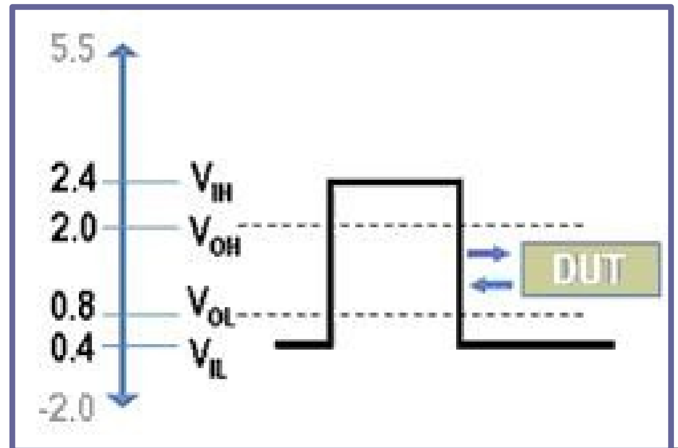
- La electrónica digital utiliza sistemas y circuitos en los que sólo existen dos estados posibles.
  - Estos estados se representan mediante dos niveles de tensión discretos y diferentes:
    - ALTO (H: high)
    - BAJO (L: low).
  - Estos dos estados pueden representarse también mediante niveles de corriente, interruptores abiertos o cerrados, o lámparas encendidas o apagadas.
  - En los sistemas digitales, las combinaciones de estos dos estados se utilizan para representar números, símbolos, caracteres alfabéticos y cualquier otro tipo de información.

- **Los dos Digitos** del sistema binario, 1 y 0, se denominan bits (contracción de BInary digiT).
  - Un 1 se representa mediante un nivel de tensión más elevado, que se denomina nivel ALTO (HIGH)
  - Un 0 se representa mediante un nivel más bajo de tensión, que se denomina nivel BAJO (LOW)
- Este convenio se denomina: Lógica positiva y es la que vamos a emplear a lo largo del curso.



Asignación lógica

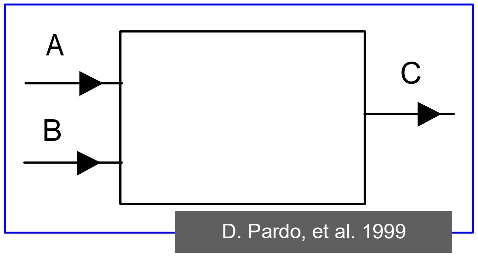
- **Rango de niveles lógicos de tensión para un circuito digital**
  - Las tensiones que se utilizan para representar los unos y ceros reciben el nombre de niveles lógicos.
  - Lo "ideal" sería que un nivel de tensión representara el nivel ALTO y otro nivel de tensión representar el nivel BAJO.
    - Sin embargo, en la práctica, un nivel ALTO o HIGH puede ser cualquier tensión entre un máximo (VHmáx) y un mínimo (VHmin) especificados.
    - De igual manera un nivel BAJO o LOW puede ser cualquier tensión comprendida entre un máximo (VLmáx) y un mínimo (VLmin) especificados.
  - Los valores de tensión comprendidas entre VLmáx y VHmin no son aceptables para un funcionamiento correcto → Una tensión dentro de este rango podría interpretarse tanto como nivel ALTO como BAJO en un circuito.



http://zone.ni.com/cms/images/devzone/tut/voltagelevel.JPG

  - Ejemplo: En una lógica digital TTL:
    - Los valores del nivel ALTO pueden variar desde 2 V a 5 V
    - Los valores del nivel bajo lo pueden hacer entre 0 V y 0.8 V.
    - Si se aplica una tensión de 3.5 V, el circuito lo interpretará como un BAJO (LOW) o 0 binario. Para este tipo de circuito las tensiones comprendidas entre 0.8 V y 2 V no son aceptables y nunca deben ser utilizadas.

- **Ejemplo:** Sin tener en cuenta su constitución interna, consideremos un circuito de dos entradas y una salida, cuya representación y salida en función de las entradas como el que se muestra en la Figura.



V_A	V_B	V_C
0V	0V	0V
5V	0V	0V
0V	5V	0V
5V	5V	5V

  - Para una lógica definida positiva, este circuito realiza la operación *AND*
  - Sin embargo, con una lógica definida negativa el circuito realiza la operación *OR*.

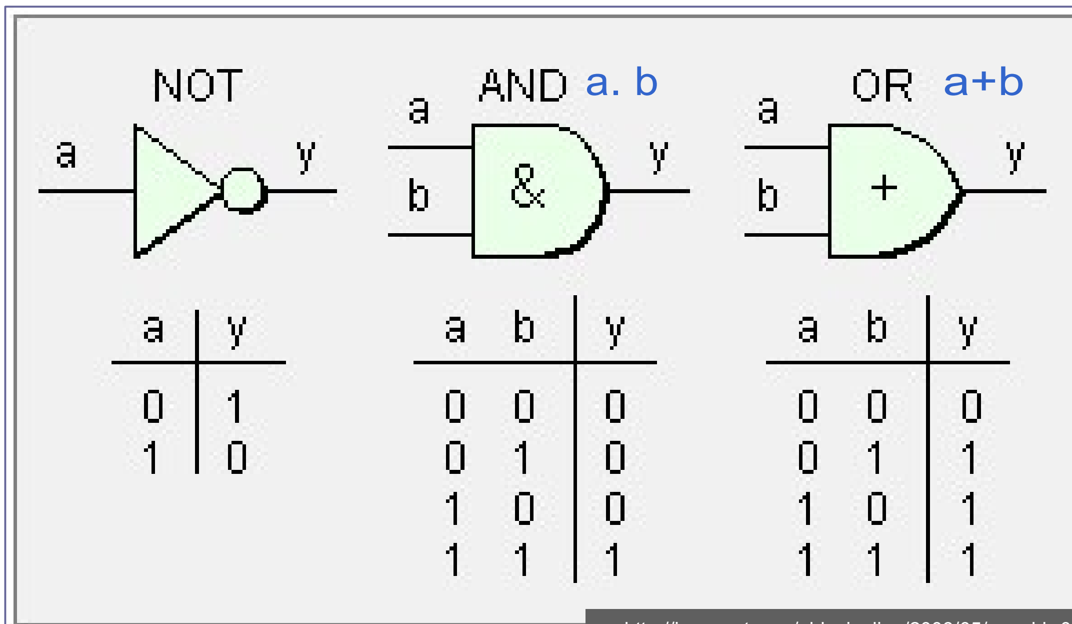
- Por tanto, al especificar el tipo de operación que realiza un circuito, debe indicarse también para qué tipo de lógica la realiza. En el caso de no especificarse, debe entenderse que es para lógica definida positiva

- Es conveniente representar los circuitos que realizan las funciones lógicas por **ciertos símbolos** que a la hora de trabajar con ellos simplifiquen su manejo.

- En lógica definida positiva, y dos entradas → Las funciones lógicas elementales, también llamadas puertas lógicas, son básicamente tres:

Los circuitos que realiza la operación:

NOT **AND** **OR** **se representan por el símbolo:**



a	y
0	1
1	0

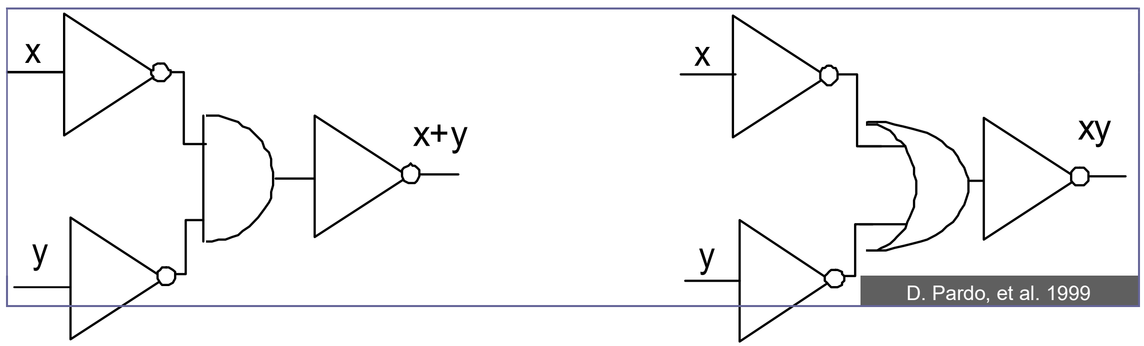
a	b	y
0	0	0
0	1	0
1	0	0
1	1	1

a	b	y
0	0	0
0	1	1
1	0	1
1	1	1

http://i.cmpnet.com/pldesignline/2006/05/max-bb-02.gif

- Debemos hacer notar que no son necesarios los tres tipos de circuitos lógicos (puertas) para realizar todas las operaciones en un Álgebra de Boole → Con dos de ellas puede realizarse la tercera.

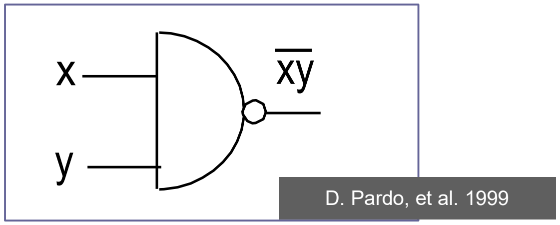
- Así tenemos, como se muestra en la Figura, que tres inversores y una puerta AND (OR) actúan como una puerta OR (AND).



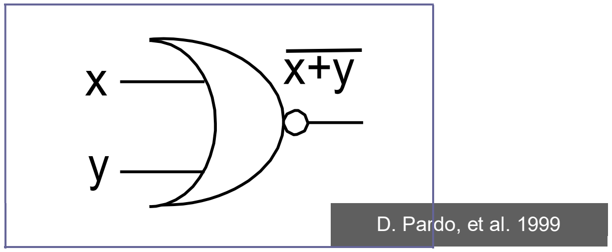
Operaciones OR y AND

- Existen operaciones compuestas de estas, como son:

  - La ***NAND*** (operación *AND* seguida de la *NOT*). Su símbolo es:



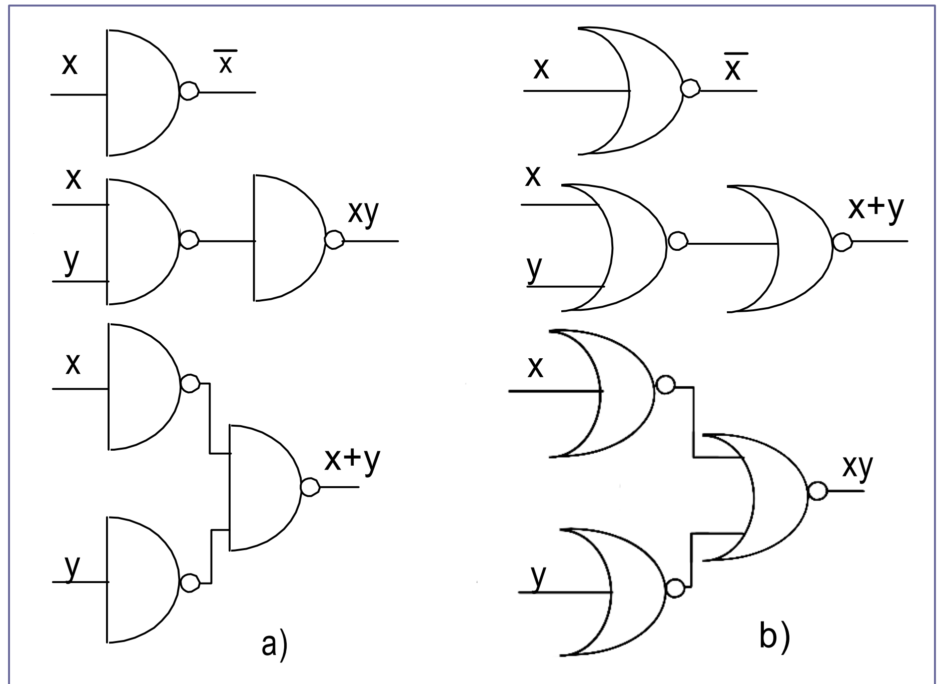
  - La ***NOR*** (operación *OR* seguida de la *NOT*), cuyo símbolo es:



- La importancia de estas puertas radica en la simplificación de ciertas funciones.

- De forma análoga a lo visto anteriormente para las puertas AND, OR y NOT.
  - Podemos comprobar que con un solo tipo de puerta (NAND o NOR) puede realizarse cualquier operación básica en un Álgebra de Boole.

- En general, puede decirse que:
  - Como NOT pueden actuar:
    - Una puerta NAND con una sola entrada (o con todas menos una puestas a 1).
    - Una puerta NOR con una sola entrada (o con todas menos una puestas a 0).
  - Una asociación en serie de un **número par** de puertas
    - NAND tienen la apariencia de una puerta AND.
    - NOR tienen la apariencia de una puerta OR
  - Una asociación en serie de un **número impar de puertas:**
    - NAND tienen la apariencia de una puerta OR.
    - NOR tienen la apariencia de una puerta AND.



D. Pardo, et al. 1999

- Por último mencionaremos que existen como bloques básicos dos puertas denominadas OR-EXCLUSIVO y NOR-EXCLUSIVO, cuya función y símbolo son las indicadas:

Diagrama lógico y símbolo 0R "exclusiva"

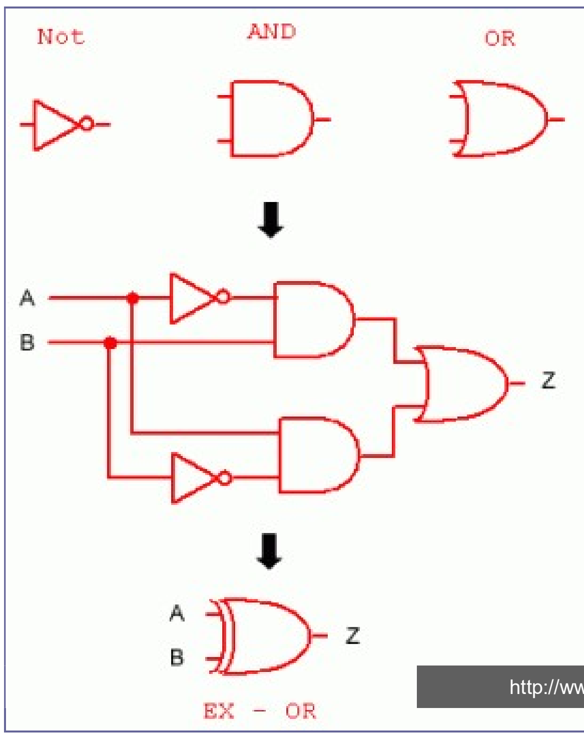
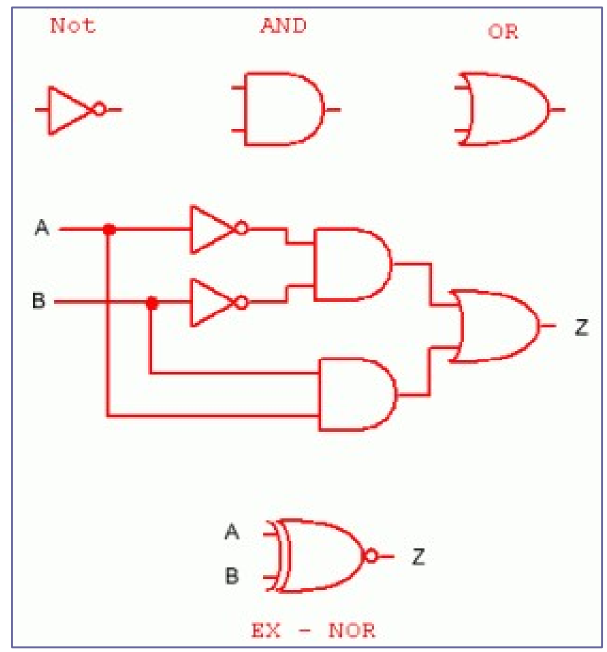


Diagrama lógico y símbolo N0R "exclusiva"



http://www.forosdeelectronica.com/tutoriales/compuertas-digitales.htm

- Una función de Boole puede representarse, por una expresión algebraica. Pero esta representación no es única.

- **Representación → Tablas de verdad**
  - Son unas representaciones gráficas de todos los casos que se pueden dar en una relación algebraica y de sus respectivos resultados.

X	Y	Z
0	0	0
0	1	1
1	0	1
1	1	0

$Z = f\,(X, Y)$

  - Una vez que se ha determinado la expresión booleana de un circuito dado, puede desarrollarse una tabla de verdad que representa la salida (función) del circuito lógico para todos los posibles valores de las entradas
  - Normalmente, la tabla de verdad suele ser el dato inicial a la hora del diseño de circuitos digitales.
  - Cuanto mayor sea el número de variables de que consta la función, más tediosa será la representación.

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

Tabla de verdad de $f(x,y,z)=xy+z$

- **Representación -+ Mapas de Karnaugh**
  - Son similares a una tabla de verdad ya que muestra todos los posibles valores de las variables de entrada y la salida resultante para cada valor.
  - Características de los Mapas de Karnaugh:
    - Esta organizado en una cuadrícula en forma de encasillado cuyo número de casillas depende del número de variables que tenga la función a simplificar.
    - Para una función de n variables, el mapa consta de 2^n cuadros, cada uno asociado a uno de los 2^n minterms diferentes que son suficientes para generar la función.
    - Las celdas se disponen de manera que la cada una se diferencia de la contigua (tanto en vertical como en horizontal) justamente en el estado de complementación de una variable.

x \ y	0	1
0	0	0
1	1	1

Mapa de Karnaugh de 2 variables : matriz de 2^2 celdas.

x y \ z	00	01	11	10
0	0	0	1	0
1	1	1	1	1

Mapa de Karnaugh de 3 variables: matriz de 2^3 celdas.

AB \ CD	00	01	11	10
00	0	4	12	8
01	1	5	13	9
11	3	7	15	11
10	2	6	14	10

Mapa de Karnaugh de 4 variables.:matriz de 2^4 celdas. Asignación de los minterms

http://www.ee.surrey.ac.uk/Projects/Labview/minimisation/graphics/nak.gif

- **Representación** -+ **Mapas de Karnaugh**
  - En el caso de tener una función booleana de 5 variables, se construye mediante 2 mapas de 4 variables (con 16 celdas cada uno, uno de ellos correspondiente al valor 0 de la quinta variable y el segundo correspondiente al valor 1.

- **Representación** -+ **Tablas de verdad y Mapas de Karnaugh**
  - Cada una de las casillas que forman el mapa puede representar términos tanto minterms como maxterms.
    - Si representamos una función en forma de **MINTERMs**, pondremos un "1" en la casilla correspondiente a cada término de la tabla de verdad y del mapa de Karnaugh.
    - Si la representamos en forma de **MAXTERMS**, pondremos un "0" en la casilla correspondiente a cada término.
  - Si disponemos del mapa de Karnaugh de una función es inmediato obtener la expresión de la función, de igual modo que desde una tabla de verdad.

- **Ejemplo -+ Tablas de verdad y Mapas de Karnaugh**

  - Consideremos la función definida por la tabla de verdad. Podrá ponerse como

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

  - Sin embargo, puede comprobarse que también las expresiones

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

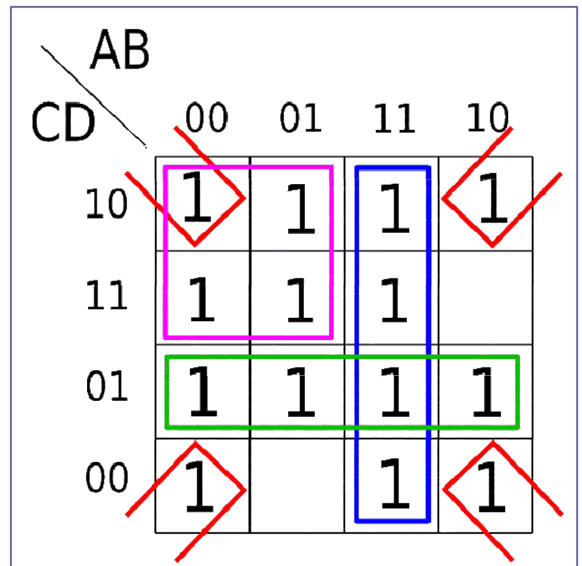
corresponden a la tabla de verdad.

x	y	z	f	Minterms	Maxterms
0	0	0	1	$\bar{x}\,\bar{y}\,\bar{z}$	
0	0	1	0		$x+y+\bar{z}$
0	1	0	0		$x+\bar{y}+z$
0	1	1	1	$\bar{x}\,y\,z$	
1	0	0	1	$x\,\bar{y}\,\bar{z}$	
1	0	1	0		$\bar{x}+y+\bar{z}$
1	1	0	1	$x\,y\,\bar{z}$	
1	1	1	0		$\bar{x}+\bar{y}+\bar{z}$

Ejemplo de tabla de verdad

  - Debemos por tanto decir que la forma anteriormente propuesta, aunque correcta, no es la más simple, salvo en casos excepcionales.
  - Antes de proceder a implementar una función es conveniente, en general, intentar obtener su forma más sencilla, **mediante su simplificación.**

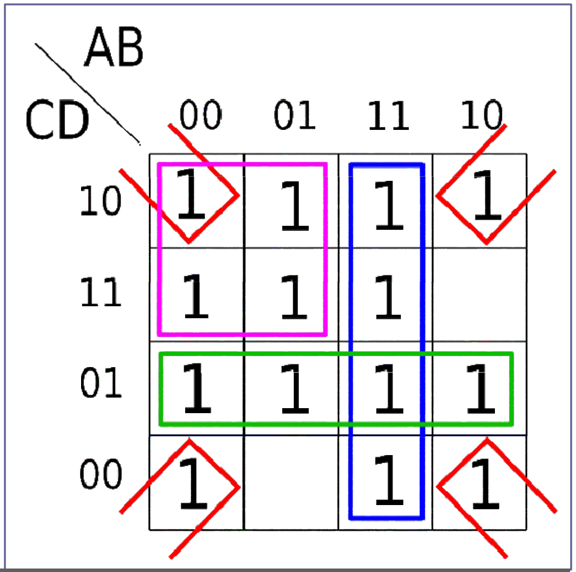
- Antes de proceder a implementar cualquier función con puertas lógicas es conveniente **obtenerla de forma más sencilla mediante simplificación**.
  - La simplificación de funciones lógicas puede obtenerse mediante la aplicación de los teoremas que hemos estudiado del Algebra de BOOLE.
  - Sin embargo, existen diferentes técnicas de simplificación de funciones lógicas basadas en agrupar MINTERMS que se diferencian en el estado de una variable
    - En un MINTERM figura complementada y en otro no
  - Las técnicas mas usuales de simplificación de modo gráfico son:
    - La utilización de los mapas de Karnaugh
    - La técnica de Quine-McKluskey
  - Vamos a estudiar la de **MAPAS DE KARNAUGH por ser la mas sencilla.**



http://upload.wikimedia.org/wikipedia/commons/thumb/d/d6/Karnaugh_map_KV_4mal4_21.svg/600px-Karnaugh_map_KV_4mal4_21.svg.png

- **Simplificación de funciones lógicas mediante mapas de Karnaugh.** Consideraciones:
  - Se pueden hacer grupos de 2, 4, 8, 16, 32 o 64 …(**2^n**) casillas adyacentes
  - **Adyacencia de celdas o minterms** (como se muestra en la figura):
    - Según los ejes coordenados (horizontal y vertical), pero nunca según ejes diagonales.
    - Los grupos de casillas de los bordes de los mapas opuestos entre sí.
    - El grupo de casillas constituido por las cuatro esquinas del mapa.
  - Un minterm puede tomarse varias veces si conviene en orden a la simplificación de la función.
  - En un mapa de una función de ***n* variables**:
    - La adyacencia de dos 1 -+ simplificación de una variable: esos dos minterms son representados por un sólo término de n-1 variables.
    - Cuatro 1 adyacentes podrán ser representados por un término producto que simplifica dos variables: quedaría de n-2 , etc..
- *NOTA: Cuando tratemos de agrupar casillas para simplificar, deberemos procurar conseguir grupos del máximo número de casillas, pero respetando las normas anteriores.*

Adyacencia de celdas del Mapa de Karnaugh 4 variables

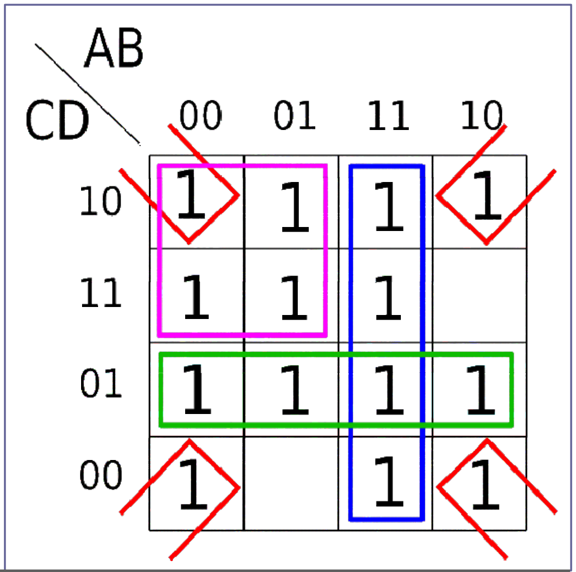


http://upload.wikimedia.org/wikipedia/commons/thumb/d/d6/Karnaugh_map_KV_4mal4_21.svg/600px-Karnaugh_map_KV_4mal4_21.svg.png

- **Simplificación de funciones lógicas mediante mapas de Karnaugh.** Consideraciones:

  - **TRUCO:** Señalemos que en algunos casos pueden introducirse MINTERMs inexistentes en la función a fin de simplificarla, si se dan una de las dos condiciones siguientes
    - NO OCURRE: es aquel minterm (combinación de las variables) que no tiene posibilidad física de aparecer.
    - NO IMPORTA: es aquel minterm que de estar presente en la entrada, no se toma en consideración el valor de la función.
  - En cualquiera de estos dos casos el minterm se tomará contribuyendo a la función como más nos interese para su máxima simplificación.

Adyacencia de celdas del Mapa de Karnaugh 4 variables



http://upload.wikimedia.org/wikipedia/commons/thumb/d/d6/Karnaugh_map_KV_4mal4_21.svg/600px-Karnaugh_map_KV_4mal4_21.svg.png

- **Ejemplo:** Simplificar, mediante el mapa de Karnaugh la siguiente función:

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

- donde el conjunto de valores:
  - * *el "x=1, y=1, z=1, v=1"* no ocurre
  - * el "x=1, y=0, z=1, v=0" no importa.

- El mapa de Karnaugh (denotando con * los minterms no ocurre y no importa):

  - La agrupación I puede representarse por: $\boxed{y\, \bar{z}\, \bar{v}}$
  - La agrupación II puede representarse por: $\boxed{z\, v}$
  - La agrupación III puede representarse por: $\boxed{\bar{y}\, z}$

zv \ xy	00	01	11	10
1	-	1	1	-
2	-	-	-	-
11	1	1	*	1
10	1	-	-	*

I

II

III

- La función esta representada por la suma de tres términos producto:

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

- *NOTA: los minterms no ocurre y no importa se han tomado como unos para mayor simplificación.*

- Agradecimientos: María Jesús Martín Martínez de la Universidad de Salamanca

- Referencias
  - Pardo Collantes, Daniel; Bailón Vega, Luís A., “Elementos de Electrónica”.Universidad de Valladolid. Secretariado de Publicaciones e Intercambio Editorial.1999.
  - http://www.forosdeelectronica.com/tutoriales/compuertas-digitales.htm
  - http://i.cmpnet.com/pldesignline/2006/05/max-bb-02.gif
  - http://zone.ni.com/cms/images/devzone/tut/voltagelevel.JPG
  - http://www.ee.surrey.ac.uk/Projects/Labview/minimisation/graphics/nak.gif
  - http://upload.wikimedia.org/wikipedia/commons/thumb/d/d6/Karnaugh_map_KV_4mal4_21.svg/600px-Karnaugh_map_KV_4mal4_21.svg.png