

Álgebra de Boole. Diseño Lógico

Fundamentos de Computadores
Escuela Politécnica Superior. UAM



Índice de la Unidad 1

U1. Álgebra de Boole y Diseño Lógico.

U1.1. Análogo vs Digital

U1.2. Sistema numérico binario. Conversión entre sistemas.

U1.3. Propiedades y teoremas básicos del álgebra booleana.

U1.3.1. Operaciones y expresiones booleanas.

U1.3.2. Leyes y reglas del álgebra de Boole. Leyes de De Morgan.

U1.4. Funciones lógicas.

U1.4.1. Expresiones booleanas y tabla de la verdad.

U1.4.2. Ampliación a varias entradas.

U1.4.3. Habilitación funcional.

U1.4.4. Implementaciones de puertas SOP y POS.

U1.5. Mapas de Karnaugh.

U1.5.1. Minimización de una suma de productos mediante el mapa K.

U1.5.2. Minimización de un producto de sumas mediante el mapa K.

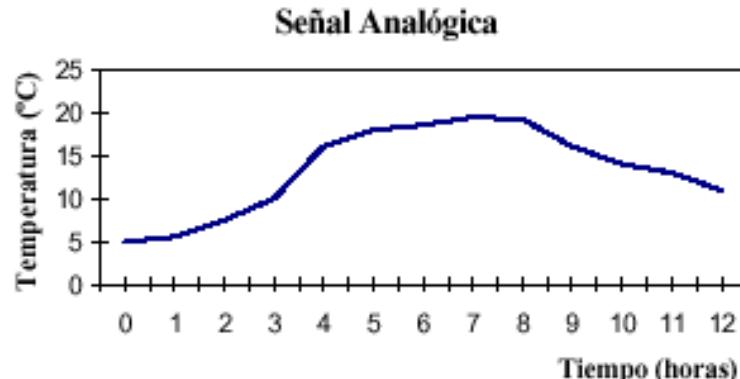
U1.5.3. Implementación de la lógica combinacional. Funciones lógicas.

Analógico vs Digital

Analógico vs Digital

Sistemas analógicos

- Trabajan con variables analógicas
- Señales físicas para representarlas: **Señales analógicas**
- Señal analógica: Puede tomar infinitos valores reales, puesto que varían de forma continua.
- Ejemplo: Termómetro de mercurio

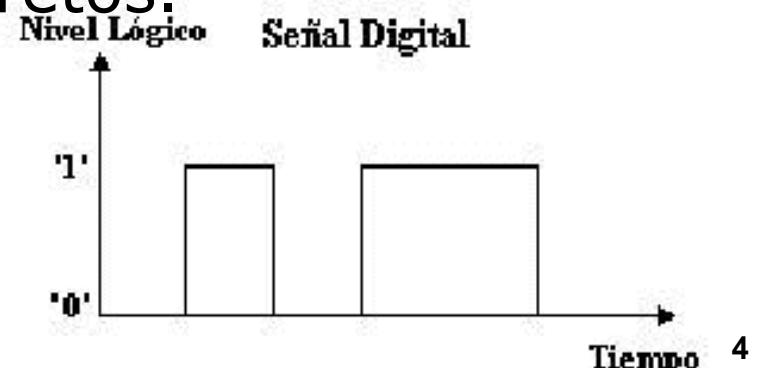


Analógico vs Digital

Analógico vs Digital

Sistemas digitales

- Trabajan con variables digitales.
 - Toma valores entre dos posibles
 - Los valores se expresan por sentencias declarativas
 - Los dos valores son excluyentes entre ellos
- Variables físicas para representarlas: **Señales digitales.**
- Señal digital: Toma valores discretos.
- Ejemplo: Interruptor de la luz



Sistema numérico binario. Conversión entre sistemas

Sistema numérico posicional

El valor del dígito depende de su posición en el número.

➤ Sistema Decimal:

Equivalente numérico en decimal					
Columna 1000's	Columna 100's	Columna 10's	Columna 1's	BASE SISTEMA	
5	3	7	4	10	

➤ Sistema binario:

Equivalente numérico en decimal					
Columna 8's	Columna 4's	Columna 2's	Columna 1's	BASE SISTEMA	
1	1	0	1	2	

Sistema numérico binario.

Conversión entre sistemas

Ejemplos de conversión entre sistemas

- Convertir de binario a decimal el número 10101_2

- Convertir de decimal a binario el número 47_{10}

Sistema numérico binario.

Conversión entre sistemas

Se recomienda el aprendizaje de las siguientes potencias:

$$\checkmark 2^0 = 1$$

$$\checkmark 2^5 = 32$$

$$\checkmark 2^{10} = 1024 = 1 \text{ k}$$

$$\checkmark 2^1 = 2$$

$$\checkmark 2^6 = 64$$

$$\checkmark 2^2 = 4$$

$$\checkmark 2^7 = 128$$

$$\checkmark 2^{20} = 1.048.576 = 1 \text{ M}$$

$$\checkmark 2^3 = 8$$

$$\checkmark 2^8 = 256$$

$$\checkmark 2^{30} = 1.073.741.824 = 1 \text{ G}$$

$$\checkmark 2^4 = 16$$

$$\checkmark 2^9 = 512$$

$$\checkmark 2^{32} = 2^2 * 2^{30} = 4 \text{ G}$$

Ejemplos:

$$\textcolor{red}{?} 2^{13} = \textcolor{red}{?}$$

$$\textcolor{red}{?} 2^{24} = \textcolor{red}{?}$$

$$\textcolor{red}{?} 2^{15} = \textcolor{red}{?}$$

Sistema numérico binario. Conversión entre sistemas

Rango de representación del sistema binario.

- Con un número de n dígitos decimales {0-9}, se representan 10^n números diferentes en el rango **[0, 10^n-1]**.
 - **Ejemplo:** con $n = 3$, $10^3 = 1000$ números diferentes. en el rango [0, 999]
- Con un número de n dígitos binarios {0, 1}, se representan 2^n números diferentes en el rango **[0, 2^n-1]**.
 - **Ejemplo:** con $n = 3$, $2^3 = 8$ números diferentes en el rango [0, 7]

Sistema numérico binario. Conversión entre sistemas

Sistema Hexadecimal

Dígito Hexadecimal	Decimal Equivalente	Binario Equivalente	Dígito Hexadecimal	Equivalente Decimal	Equivalente Binario
0	0	0000	8	8	1000
1	1	0001	9	9	1001
2	2	0010	A	10	1010
3	3	0011	B	11	1011
4	4	0100	C	12	1100
5	5	0101	D	13	1101
6	6	0110	E	14	1110
7	7	0111	F	15	1111

El hexadecimal es un sistema numérico posicional con base 16, que se utiliza para escribir de forma abreviada números en binario.

Sistema numérico binario.

Conversión entre sistemas

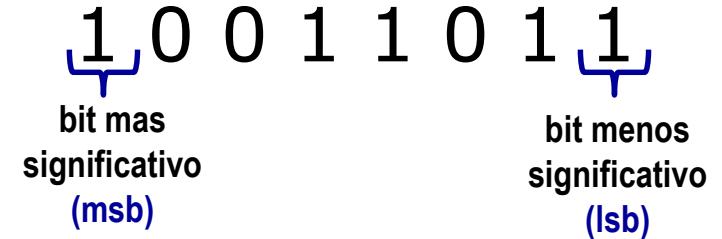
Ejemplos de conversión entre sistemas

- Convertir de hexadecimal a binario el número $4AF_{16}$; también (0x4AF)

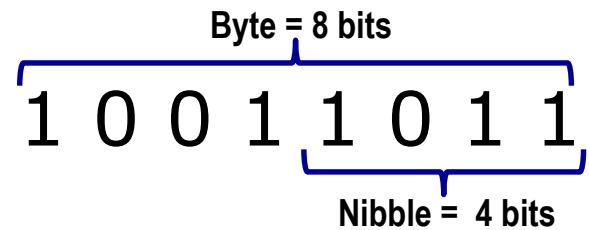
- Convertir de hexadecimal a decimal el número 0x4AF

Sistema numérico binario. Conversión entre sistemas

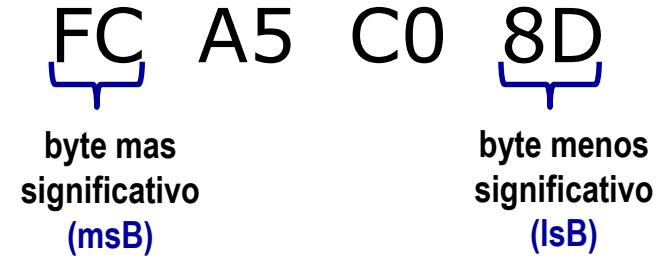
- Bits



- Bytes & Nibbles

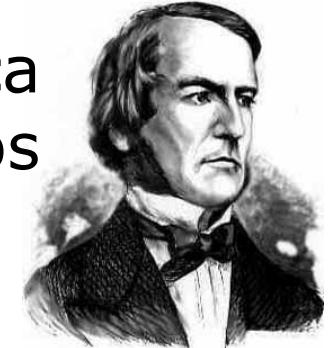


- Bytes



Operaciones y expresiones booleanas

Álgebra de Boole: es la herramienta matemática utilizada para el análisis y la síntesis de los sistemas digitales binarios.



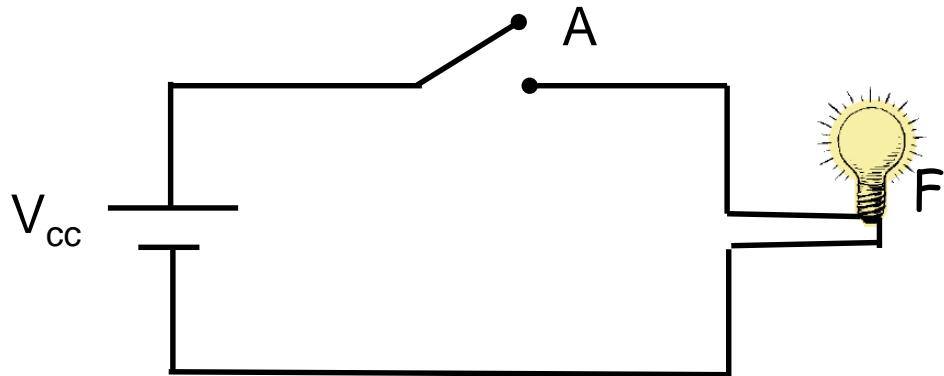
George Boole
Matemático inglés
(1815-1864)

Variable booleana: es una señal digital que en un instante determinado sólo puede tomar uno de dos valores. Los valores a tomar son mutuamente excluyentes.

- Se representan como: **0** y **1**; **OFF** y **ON**; **etc...**

Operaciones y expresiones booleanas

- Variables lógicas y circuitos eléctricos:



- Estado del interruptor **A**:
 - Abierto (0)
 - Cerrado (1)
- Estado de la bombilla **F**:
 - Apagada (0)
 - Encendida (1)

El estado de la variable lógica bombilla es función del estado de la variable lógica interruptor

FUNCIÓN: "La bombilla está encendida si el interruptor está cerrado"

Operaciones y expresiones booleanas

- **Función lógica:** Circuito que acepta valores lógicos a la entrada y produce un valor lógico a la salida
- **Tabla de verdad:** describe el funcionamiento de las funciones lógicas.
 - Especifica la salida de la puerta o función lógica para todas las posibles combinaciones de entradas
 - Son representaciones gráficas de todos los casos que se pueden dar en una relación algebraica y de sus respectivos resultados
- **Puertas lógicas:** Implementan a las funciones lógicas más elementales.

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

Operaciones y expresiones booleanas

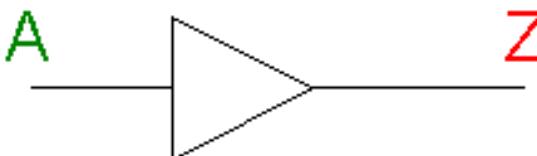
- EL AMPLIFICADOR (BUFFER)

- Puerta lógica más sencilla
- Una entrada (A) y una salida (Z)
- Tabla de verdad:

A	Z
1	1
0	0

- Ecuación lógica: **$Z = A$**

- Representación gráfica:



Operaciones y expresiones booleanas

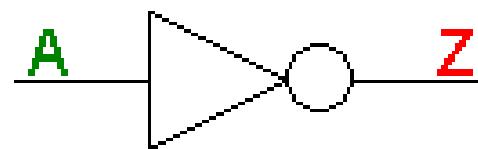
- LA PUERTA NOT O INVERSOR

- Una entrada (A) y una salida (Z)
- Tabla de verdad:

A	Z
1	0
0	1

- Ecuación lógica: $Z = \bar{A}$

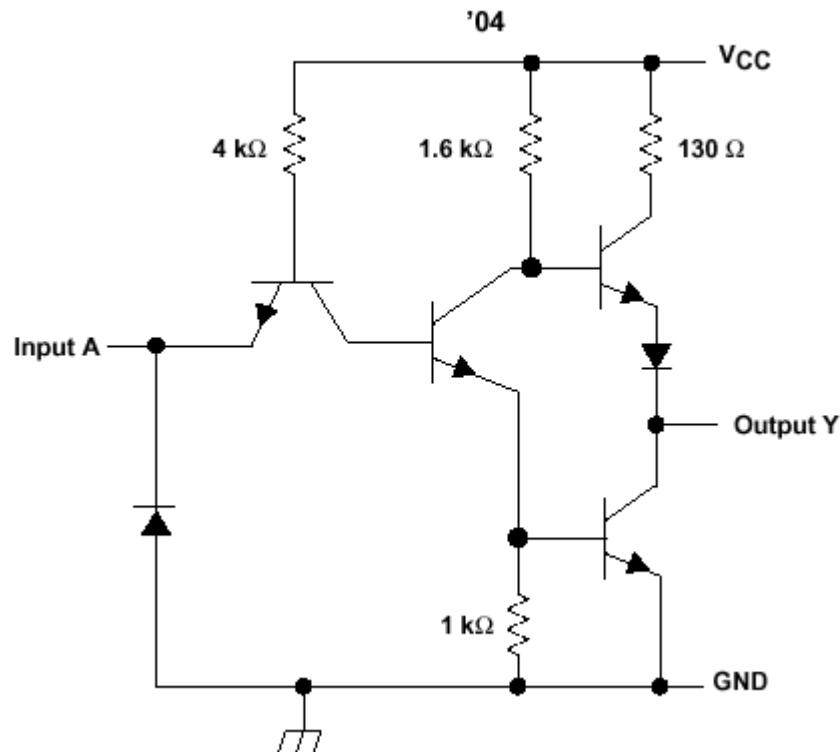
- Representación gráfica:



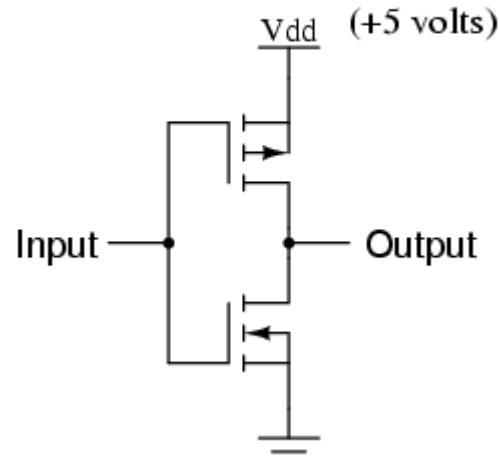
Operaciones y expresiones booleanas

- LA PUERTA NOT O INVERSOR

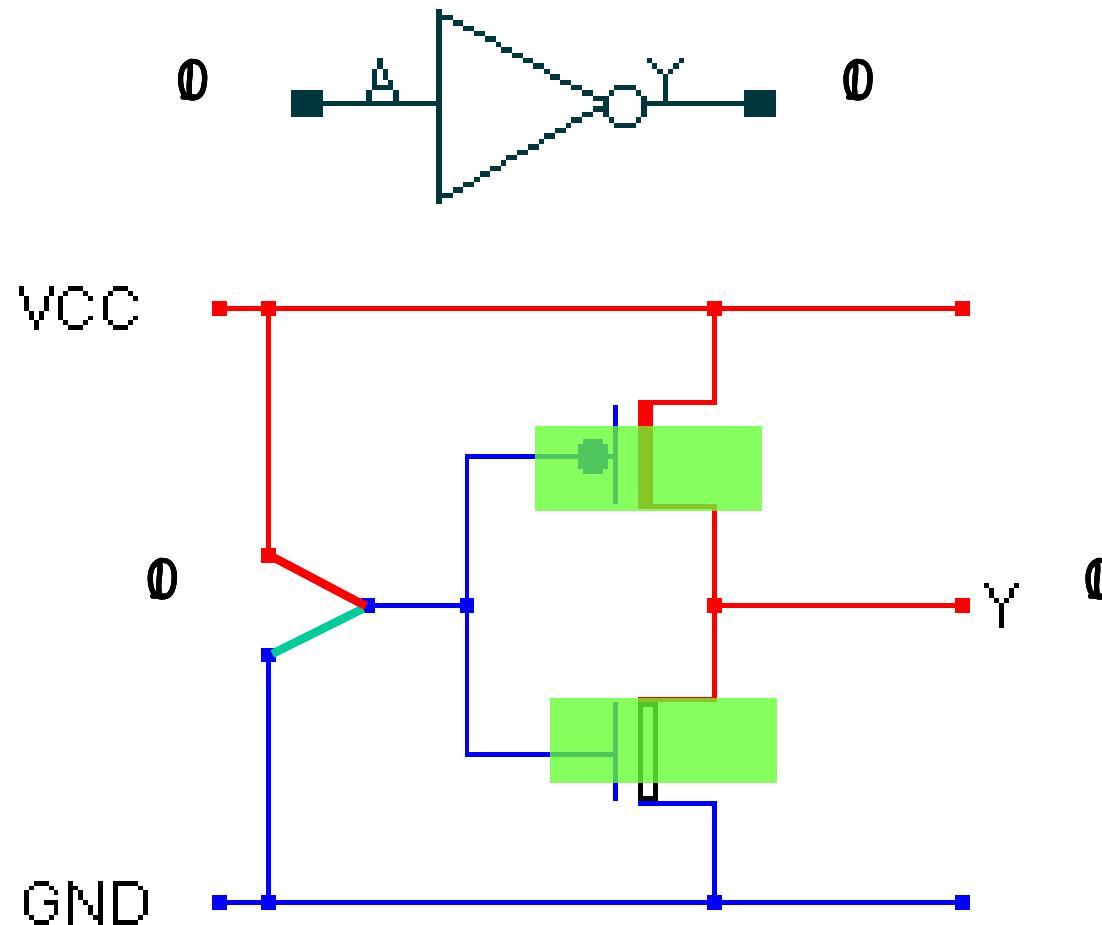
- Lógica interna de la puerta NOT:



Inverter circuit using IGFETs



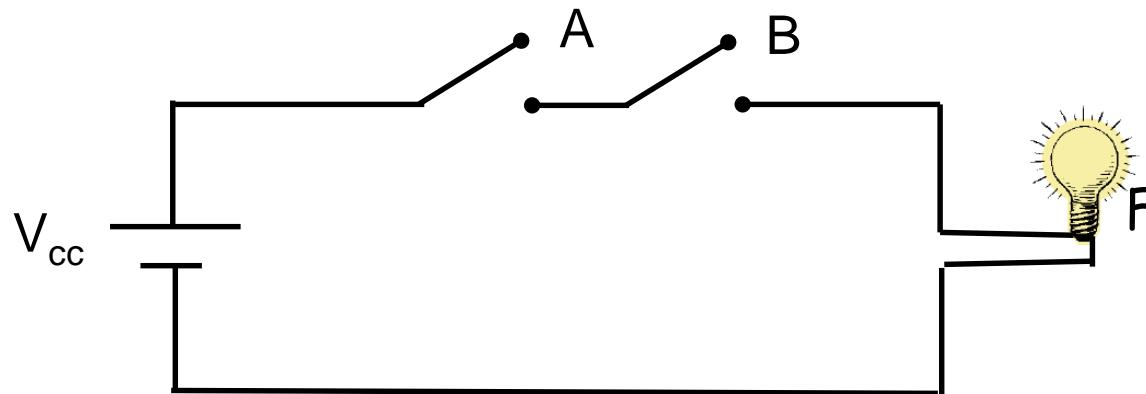
Operaciones y expresiones booleanas



Operaciones y expresiones booleanas

- LA PUERTA AND

- La puerta AND vista como interruptores:



FUNCIÓN: La bombilla está encendida si:

“el interruptor A **Y** el interruptor B están cerrados”

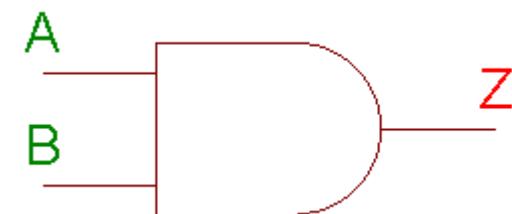
Operaciones y expresiones booleanas

- LA PUERTA AND (2 entradas {A,B} y 1 salida {Z})
 - $Z=1$ sólo si las dos entradas están simultáneamente a 1
 - Tabla de verdad:

A	B	Z
0	0	0
0	1	0
1	0	0
1	1	1

- Ecuación lógica: $Z = A \cdot B$

- Representación gráfica:

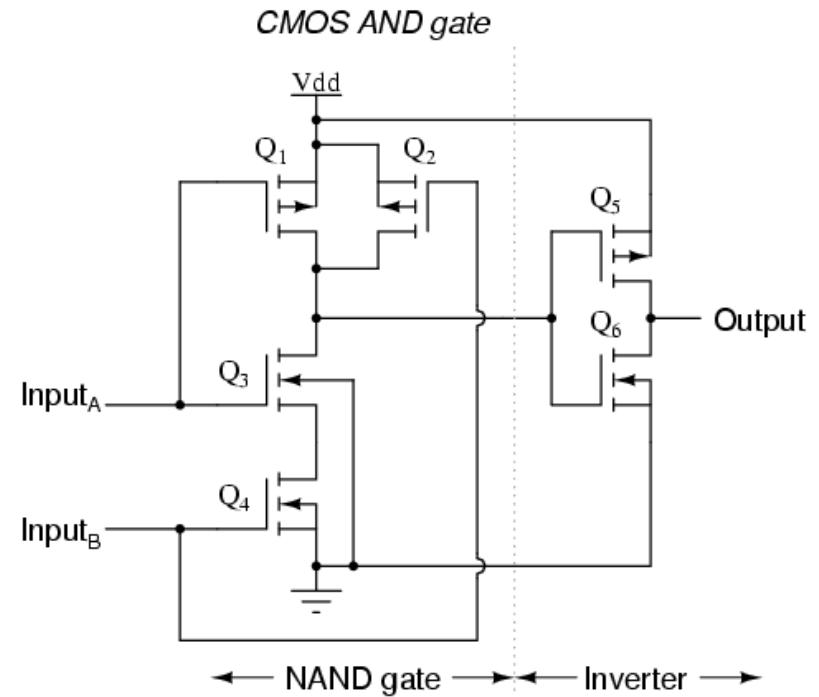
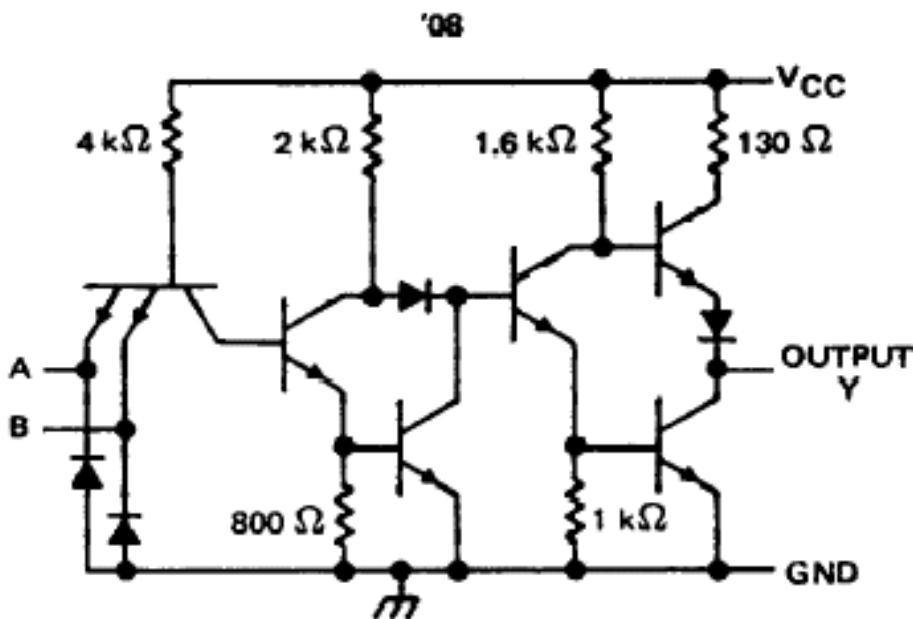


¿Puerta AND de varias entradas?

Operaciones y expresiones booleanas

- LA PUERTA AND

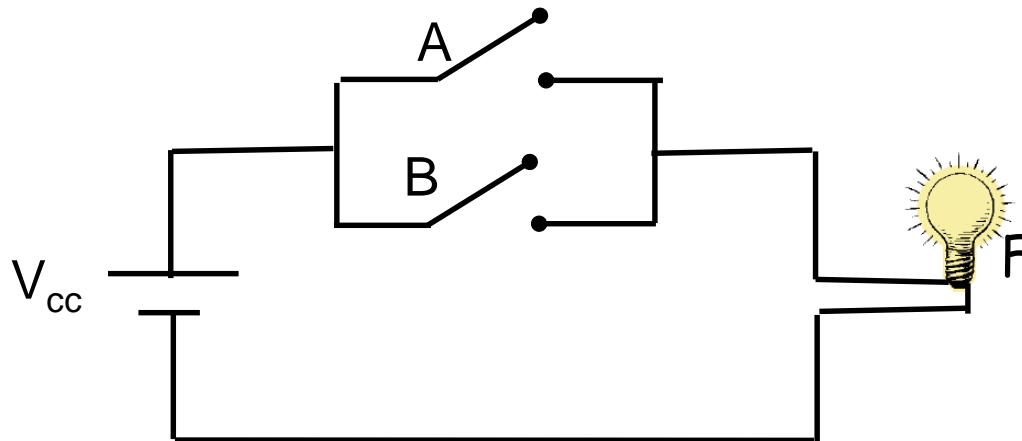
- Lógica interna de la puerta AND:



Operaciones y expresiones booleanas

- LA PUERTA OR

- La puerta OR vista como interruptores:



FUNCIÓN: La bombilla está encendida si:

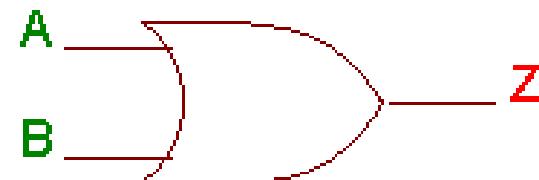
“el interruptor A, **O** el interruptor B **O** ambos están cerrados”

Operaciones y expresiones booleanas

- LA PUERTA OR: (2 entradas {A,B} y 1 salida {Z})
 - $Z = 1$ cuando alguna de las dos entradas vale 1
 - Tabla de verdad:

A	B	Z
0	0	0
0	1	1
1	0	1
1	1	1

- Ecuación lógica: $Z = A + B$



- Representación gráfica:

¿Puerta OR de varias entradas?

Operaciones y expresiones booleanas

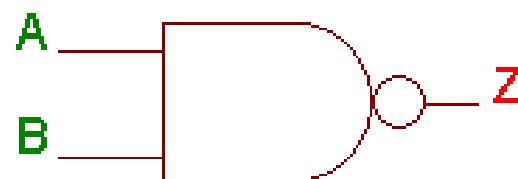
- LA PUERTA NAND: (2 entradas {A,B} y 1 salida {Z})

- $Z=1$ si al menos una de las dos entradas vale 0
 - Tabla de verdad:

A	B	Z
0	0	1
0	1	1
1	0	1
1	1	0

- Ecuación lógica: $Z = \overline{A \cdot B}$

- Representación gráfica:



¿Puerta NAND de varias entradas?

Operaciones y expresiones booleanas

- LA PUERTA NOR: (2 entradas {A,B} y 1 salida {Z})

- Z=0 si al menos una de las dos entradas vale 1
 - Tabla de verdad:

A	B	Z
0	0	1
0	1	0
1	0	0
1	1	0

- Ecuación lógica: $Z = \overline{A + B}$

- Representación gráfica:



¿Puerta NOR de varias entradas?

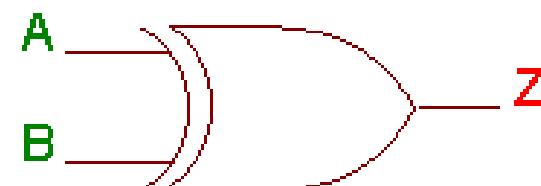
Operaciones y expresiones booleanas

- LA PUERTA XOR (OR-Exclusiva): (2 entradas {A,B} y 1 salida {Z})
 - $Z=1$ si y sólo si una de las entradas está a 1
 - Tabla de verdad:

A	B	Z
0	0	0
0	1	1
1	0	1
1	1	0

- Ecuación lógica: $Z = A \oplus B$

- Representación gráfica:



¿Puerta XOR de varias entradas? -> Número impar de 1s

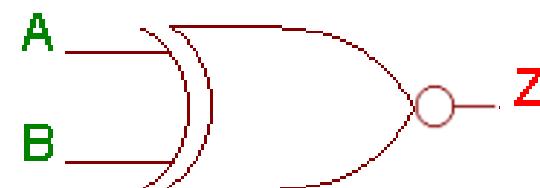
Operaciones y expresiones booleanas

- LA PUERTA XNOR (NOR-Exclusiva): (2 entradas {A,B} y 1 salida {Z})
 - Z=0 si sólo una de las entradas está a 0
 - Tabla de verdad:

A	B	Z
0	0	1
0	1	0
1	0	0
1	1	1

- Ecuación lógica: $Z = \overline{(A \oplus B)}$

- Representación gráfica:



¿Puerta XNOR de varias entradas? -> Número par de 1s

Leyes y reglas del álgebra de Boole

Nombre		Teorema		Dual
Identidad	T1	$B \bullet 1 = B$	T1'	$B + 0 = B$
Elemento nulo	T2	$B \bullet 0 = 0$	T2'	$B + 1 = 1$
Idempotencia	T3	$B \bullet B = B$	T3'	$B + B = B$
Involución	T4		//B	= B
Complemento	T5	$B \bullet /B = 0$	T5'	$B + /B = 1$
Prop. Comutativa	T6	$B \bullet C = C \bullet B$	T6'	$B + C = C + B$
Prop. Asociativa	T7	$(B \bullet C) \bullet D = B \bullet (C \bullet D)$	T7'	$(B + C) + D = B + (C + D)$
Prop. Distributiva	T8	$(B \bullet C) + (B \bullet D) = B \bullet (C + D)$	T8'	$(B + C) \bullet (B + D) = B + (C \bullet D)$
Ley de De Morgan	T12	$/(B_0 \bullet B_1 \bullet \dots \bullet B_{n-2} \bullet B_{n-1}) =$ $= (/B_0 + /B_1 + \dots + /B_{n-2} + /B_{n-1})$	T12'	$/(B_0 + B_1 + \dots + B_{n-2} + B_{n-1}) =$ $= (/B_0 \bullet /B_1 \bullet \dots \bullet /B_{n-2} \bullet /B_{n-1})$

Ecuaciones duales en el Álgebra de Boole

Circuitos lógicos

Las combinaciones de diferentes valores lógicos a la entrada hacen que aparezcan distintos valores lógicos a la salida => **CIRCUITO LÓGICO**

Un circuito lógico se compone de:

- Entradas
- Salidas
- Especificación funcional
- Especificación temporal

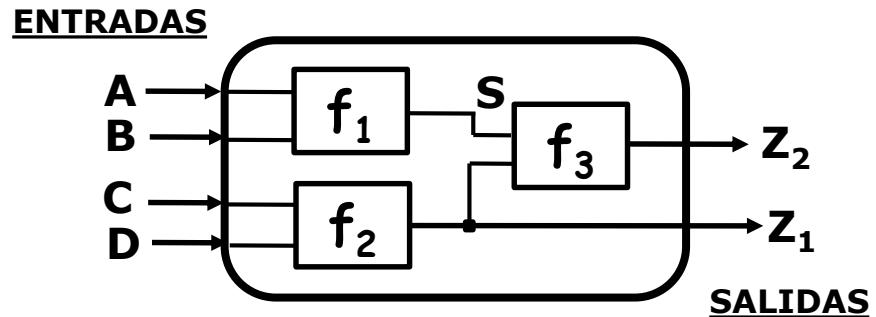


Cualquier función lógica puede expresarse como función de las puertas AND, OR y NOT

Circuitos lógicos

Ejemplo de un circuito lógico:

- Entradas: A, B, C y D
- Salidas: Z_1 y Z_2
- Especificación funcional
 - $S = f_1(A, B)$
 - $Z_1 = f_2(C, D)$
 - $Z_2 = f_3(S, Z_1) = (A, B, C, D)$
- Especificación temporal: $(\Delta t_{Z_2} = \text{MAX}\{\Delta t_{f_1}, \Delta t_{f_2}\} + \Delta t_{f_3})$



Lógica combinacional: si el estado de las salidas depende sólo del estado de las entradas. Sistema sin memoria.

Lógica secuencial: si el estado de la salida también depende del estado anterior del sistema. El circuito tiene memoria.

Funciones lógicas

- **Función lógica:** Expresión Booleana que relaciona variables lógicas directas o complementadas por medio de operaciones AND y OR
- Las funciones lógicas se expresan con circuitos lógicos de dos niveles que en su forma canónica pueden ser:
 - Suma de productos de todas las variables o sus conjugadas:
Suma de **Minterms** // Circuitos **SOP** (*Sum Of Products*)
 - Producto de sumas de todas las variables o sus conjugadas:
Producto de **Maxterms** // Circuitos **POS** (*Product Of Sums*)

Funciones lógicas

- Todas las ecuaciones Booleanas pueden ser descritas como suma de minterms (SOP)
- Cada fila en una tabla de verdad es un minterm
- Un minterm es un producto (AND) de las variables y sus complementos
- Cada minterm es VERDADERO ('1') para esa fila (y sólo para esa fila)
- La función se construye por la suma (OR) de los minterms para los cuales la salida es VERDADERA
- Se trata por tanto de una suma (OR) de productos (AND)

A	B	Y	minterm
0	0	0	$\bar{A} \bar{B}$
0	1	1	$\bar{A} B$
1	0	0	$A \bar{B}$
1	1	1	$A B$

$$Y = F(A, B) = \overline{A}B + A\overline{B}$$

Funciones lógicas

- Todas las ecuaciones Booleanas pueden ser descritas como producto de maxterms (POS)
- Cada fila en una tabla de verdad es un maxterm
- Un maxterm es una suma (OR) de las variables y sus complementos
- Cada maxterm es FALSO ('0') para esa fila (y sólo para esa fila)
- La función se construye por el producto (AND) de los maxterms para los cuales la salida es FALSA
- Se trata por tanto de un producto (AND) de sumas (OR)

A	B	Y	maxterm
0	0	0	$A + B$
0	1	1	$A + \bar{B}$
1	0	0	$\bar{A} + B$
1	1	1	$\bar{A} + \bar{B}$

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

Funciones lógicas

- Ejemplo: Desarrollo canónico de una función a partir de su tabla de verdad.

Nº	A	B	C	$F(A,B,C)$	Minterms	Maxterms
0	0	0	0	1	$(\bar{A} \cdot \bar{B} \cdot \bar{C})$	
1	0	0	1	0		$(A + B + \bar{C})$
2	0	1	0	1	$(\bar{A} \cdot B \cdot \bar{C})$	
3	0	1	1	1	$(\bar{A} \cdot B \cdot C)$	
4	1	0	0	0		$(\bar{A} + B + C)$
5	1	0	1	0		$(\bar{A} + B + \bar{C})$
6	1	1	0	1	$(A \cdot B \cdot \bar{C})$	
7	1	1	1	1	$(A \cdot B \cdot C)$	

Funciones lógicas

- Utilizando Minterms:

$$\begin{aligned} F &= (\bar{A} \cdot \bar{B} \cdot \bar{C}) + (\bar{A} \cdot B \cdot \bar{C}) + (\bar{A} \cdot B \cdot C) + (A \cdot B \cdot \bar{C}) + (A \cdot B \cdot C) = \\ &= m_0 + m_2 + m_3 + m_6 + m_7 = \sum m(0,2,3,6,7) \end{aligned}$$

- Utilizando Maxterms:

$$F = (A + B + \bar{C}) \cdot (\bar{A} + B + C) \cdot (\bar{A} + B + \bar{C}) = M_1 \cdot M_4 \cdot M_5 = \prod M(1,4,5)$$

- La implementación de una función lógica por medio de Minterms o de Maxterms requiere la mayor cantidad de recursos. Por tanto es conveniente, si es posible, obtener expresiones más simplificadas.

Funciones lógicas

- Diagramas de Karnaugh (mapas k)
 - Facilitan la realización de diseños lógicos con la estructura de puertas más sencilla  mayor economía de diseño.
 - Secuencia de **celdas** en la que cada celda representa un valor binario de las variables de entrada. Cada celda contiene el valor correspondiente de la función para dicha combinación.
 - Las celdas se disponen de manera que la simplificación de una determinada expresión consiste en agrupar adecuadamente las celdas.
 - Pueden utilizarse para expresiones de 2, 3, 4, 5 ó 6 variables.
 - Para n variables hacen falta 2^n celdas.
 - Para un número mayor de variables se utilizan otros métodos (Quine-McClusky) o métodos CAD.

Funciones lógicas

- Diagramas de Karnaugh de tres variables

- A, B, C: Variables. Conjunto de 8 celdas
- Los valores binarios de A y B se encuentran en la parte izquierda y los de C en la parte superior (puede hacerse al revés)

	AB	C	
	0	1	
00	1	0	
01	0	0	
11	0	0	
10	0	1	

	AB	C	
	0	1	
00	($\bar{A} \cdot \bar{B} \cdot \bar{C}$)	($\bar{A} \cdot \bar{B} \cdot C$)	
01	($\bar{A} \cdot B \cdot \bar{C}$)	($\bar{A} \cdot B \cdot C$)	
11	($A \cdot B \cdot \bar{C}$)	($A \cdot B \cdot C$)	
10	($A \cdot \bar{B} \cdot \bar{C}$)	($A \cdot \bar{B} \cdot C$)	

Ejemplo: La función vale '1' en la celda de la esquina superior izquierda corresponde a un valor de las variables A, B y C de 000 (/A./B./C) y en la celda de la esquina inferior derecha corresponde a un valor de las variables de 101 (A./B.C)

Funciones lógicas

- Diagramas de Karnaugh de cuatro variables
 - A, B, C, D: Variables. Matriz de 16 celdas
 - Los valores binarios de A y B se muestran en la parte izquierda de la tabla y los de C y D en la parte superior

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

		CD	
		00	01
AB	00	$(\bar{A} \cdot \bar{B} \cdot \bar{C} \cdot \bar{D})$	$(\bar{A} \cdot \bar{B} \cdot \bar{C} \cdot D)$
	01	$(\bar{A} \cdot B \cdot \bar{C} \cdot \bar{D})$	$(\bar{A} \cdot B \cdot \bar{C} \cdot D)$
AB	11	$(A \cdot B \cdot \bar{C} \cdot \bar{D})$	$(A \cdot B \cdot \bar{C} \cdot D)$
	10	$(A \cdot \bar{B} \cdot \bar{C} \cdot \bar{D})$	$(A \cdot \bar{B} \cdot \bar{C} \cdot D)$

Ejemplo: La función vale '1' en la celda de la esquina superior derecha corresponde a un valor de las variables A, B, C y D de 0010 (/A./B.C./D), en la celda de la segunda fila izquierda que corresponde a un valor de 0100 (/A.B./C./D) y en la última fila que corresponde a un valor de 1001 (/A./B./C.D).

Funciones lógicas

- Diagramas de Karnaugh de cinco variables

- A, B, C, D, E: Variables. Matriz de 32 celdas
- Se consideran dos tablas de cuatro variables (B, C, D y E), una correspondiendo al valor de la variable A=0 y otra a A=1
- Los valores binarios de B y C se muestran en la parte izquierda de la tabla y los de D y E en la parte superior

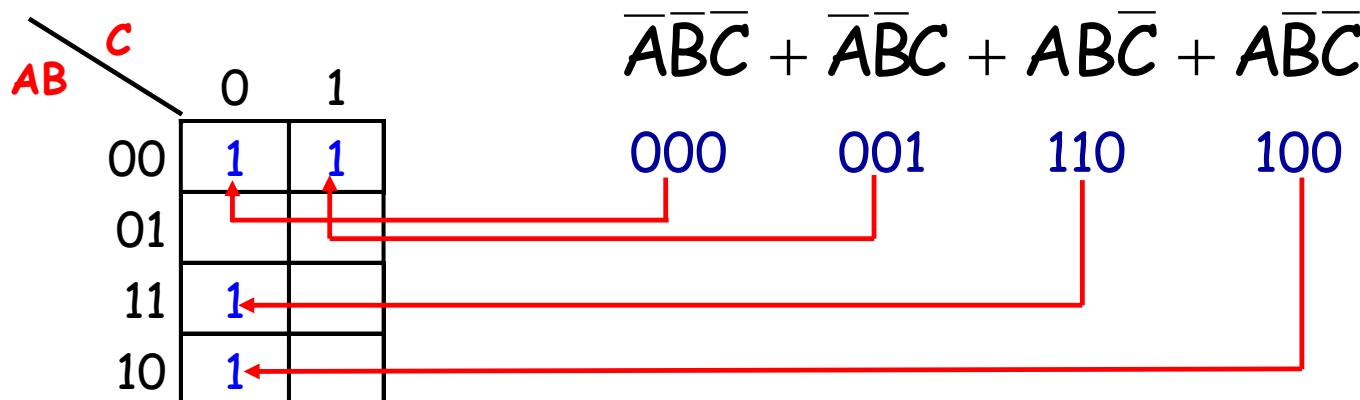
		DE							
		00	01	11	01	00	01	11	10
BC		00				00			
	00								
	01								
	11								
	10								
A = 0					A = 1				

Funciones lógicas

- Diagramas de Karnaugh. Adyacencia de celdas
 - Las celdas de un diagrama de Karnaugh se disponen de manera que sólo cambia una única variable entre celdas adyacentes
 - Celdas que difieren en una única variable: Adyacentes
 - Celdas cuyo valor difiere en más de una variable: No adyacentes
 - Físicamente, *cada celda es adyacente a las celdas que están situadas inmediatas a ella por cualquiera de sus cuatro lados*
 - Las celdas adyacentes tienen una distancia de Hamming de 1.
 - Una celda no es adyacente a aquellas que tocan diagonalmente alguna de sus esquinas
 - Las celdas de la fila superior son adyacentes a las de la fila inferior y las celdas de la columna izquierda son adyacentes a la de la comuna derecha ➤ **Adyacencia cíclica**

Funciones lógicas

- Diagrama de Karnaugh de una suma de productos estándar
 - Por cada término de la expresión suma de productos, se coloca un 1 en el diagrama de Karnaugh en la celda correspondiente al valor del producto



Funciones lógicas

- Diagrama de Karnaugh de una suma de productos no estándar
 - A un término en forma no estándar le faltan una o más variables en su expresión, que habrá que completar

		0	1
		00	1
		01	1
		11	1
		10	1

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

000	100	110
001	101	
010		
011		

Funciones lógicas

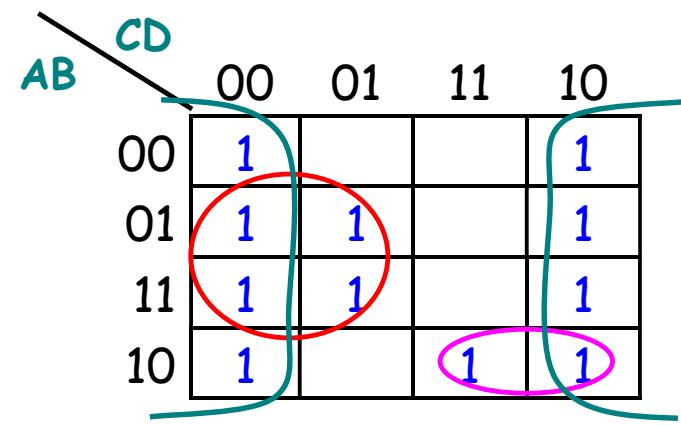
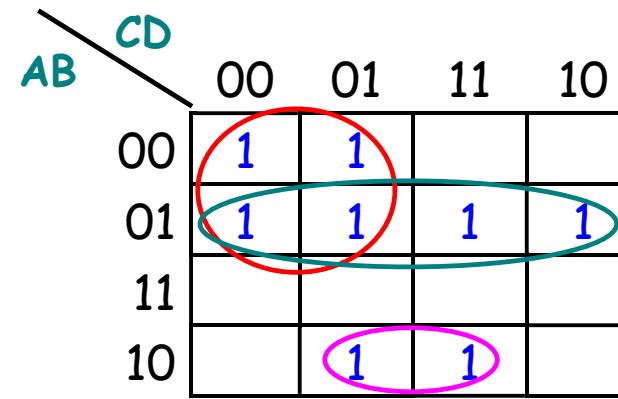
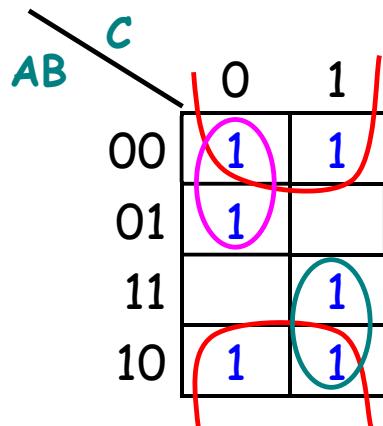
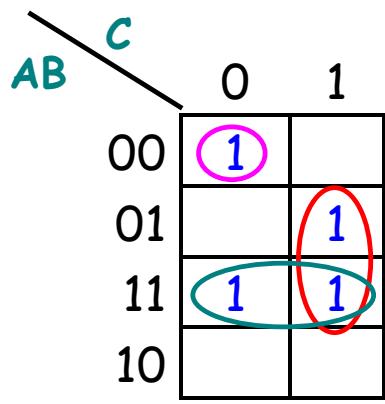
- Diagramas de Karnaugh. Simplificación de una suma de productos
 - **Minimización:** Proceso que genera una expresión que contiene el menor número posible de términos con el mínimo número de variables posibles
 - Tras obtener el diagrama de Karnaugh, hay que seguir tres pasos para obtener la expresión de productos mínima:

Funciones lógicas

- Diagramas de Karnaugh. Simplificación de una suma de productos
 - Agrupación de 1's
 - Un grupo tiene que contener 1, 2, 4, 8 ó 16 celdas (potencias de 2).
Diagrama de 3 variables: Grupo máximo de 8 celdas
 - Cada celda de un grupo tiene que ser adyacente a una o más celdas del mismo grupo, pero no todas las celdas del grupo tienen que ser adyacentes entre sí
 - Incluir siempre en cada grupo el mayor número posible de 1's
 - Cada 1 del diagrama tiene que estar incluido en al menos un grupo. Los 1's que ya pertenezcan a un grupo pueden estar incluidos en otro siempre que los grupos que solapen contengan 1's no comunes

Funciones lógicas

- Diagramas de Karnaugh. Simplificación de una suma de productos
 - Agrupación de 1's

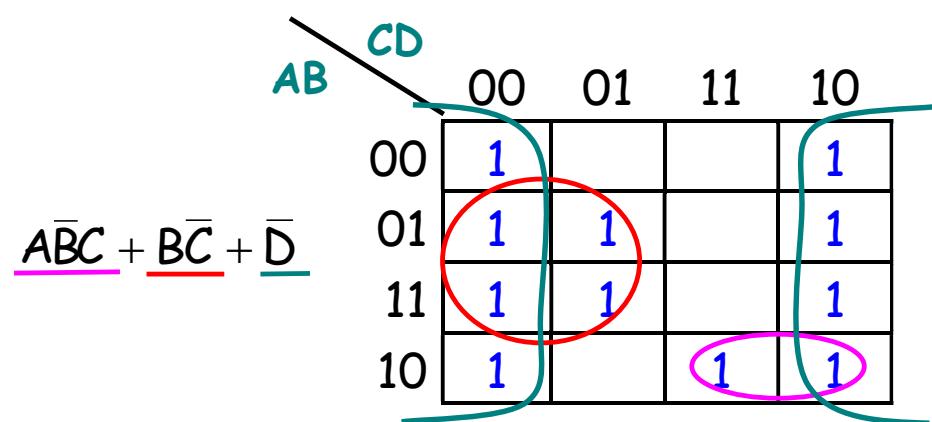
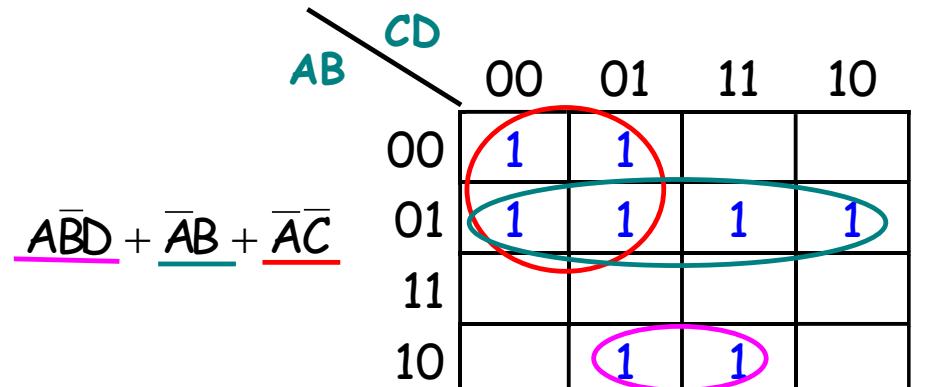
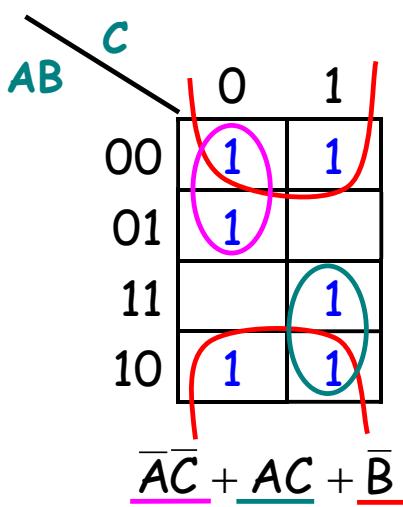
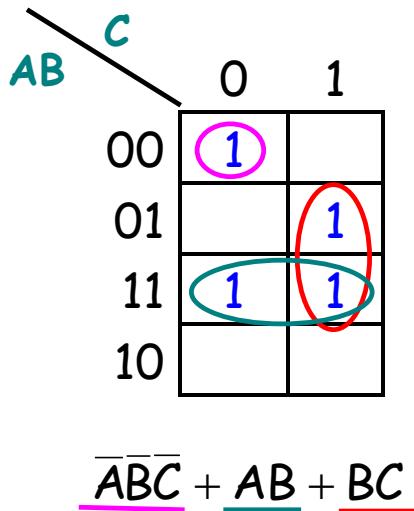


Funciones lógicas

- Diagramas de Karnaugh. Simplificación de una suma de productos
 - Determinación de la expresión suma de productos mínima a partir del diagrama. Suma de los términos producto obtenidos
 - Cada grupo de celdas que contiene 1's da lugar a un término producto compuesto por todas las variables que aparecen en el grupo en sólo una forma (no complementada o complementada)
 - Las variables que aparecen complementadas y sin complementar dentro del mismo grupo se eliminan
 - Una vez obtenidos todos los términos mínimos a partir del diagrama de Karnaugh, se suman para obtener la expresión de productos mínima

Funciones lógicas

- Diagramas de Karnaugh. Simplificación de una suma de productos
 - Determinación de la expresión suma de productos mínima a partir del diagrama. Suma de los términos producto obtenidos



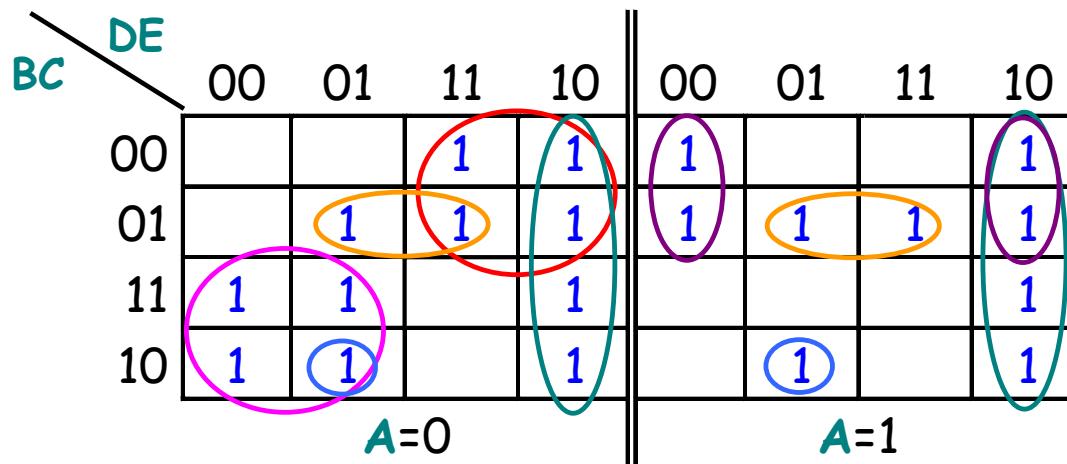
Funciones lógicas

- Diagramas de Karnaugh de cinco variables.

Simplificar $F(A,B,C,D,E) =$

$\Sigma(2,3,5,6,7,8,9,10,12,13,14,16,18,20,21,22,23,25,26,30)$

- Dos mapas de 4 variables, uno para la quinta variable a 0 y otro para la quinta variable a 1
- **Adyacencia:** Imaginar que el mapa $A=0$ está colocado encima del mapa $A=1$. Cada celda del mapa $A=0$ es adyacente con la celda que está justo debajo en el mapa $A=1$



$$\underline{\overline{ABD}} + \underline{\overline{ABD}} + \underline{\overline{ABE}} + \underline{DE} + \underline{\overline{BCE}} + \underline{\overline{BCDE}}$$

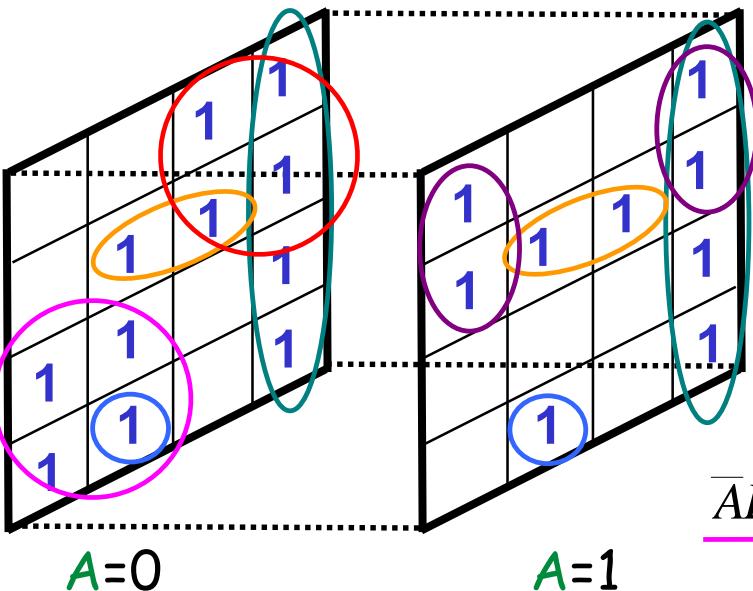
Funciones lógicas

- Diagramas de Karnaugh de cinco variables.

Simplificar

$$F(A,B,C,D,E) = \Sigma(2,3,5,6,7,8,9,10,12,13,14,16,18,20,21,22,23,25,26,30)$$

- Dos mapas de 4 variables, uno para la quinta variable a 0 y otro para la quinta variable a 1
- **Adyacencia:** Imaginar que el mapa $A=0$ está colocado encima del mapa $A=1$. Cada celda del mapa $A=0$ es adyacente con la celda que está justo debajo en el mapa $A=1$



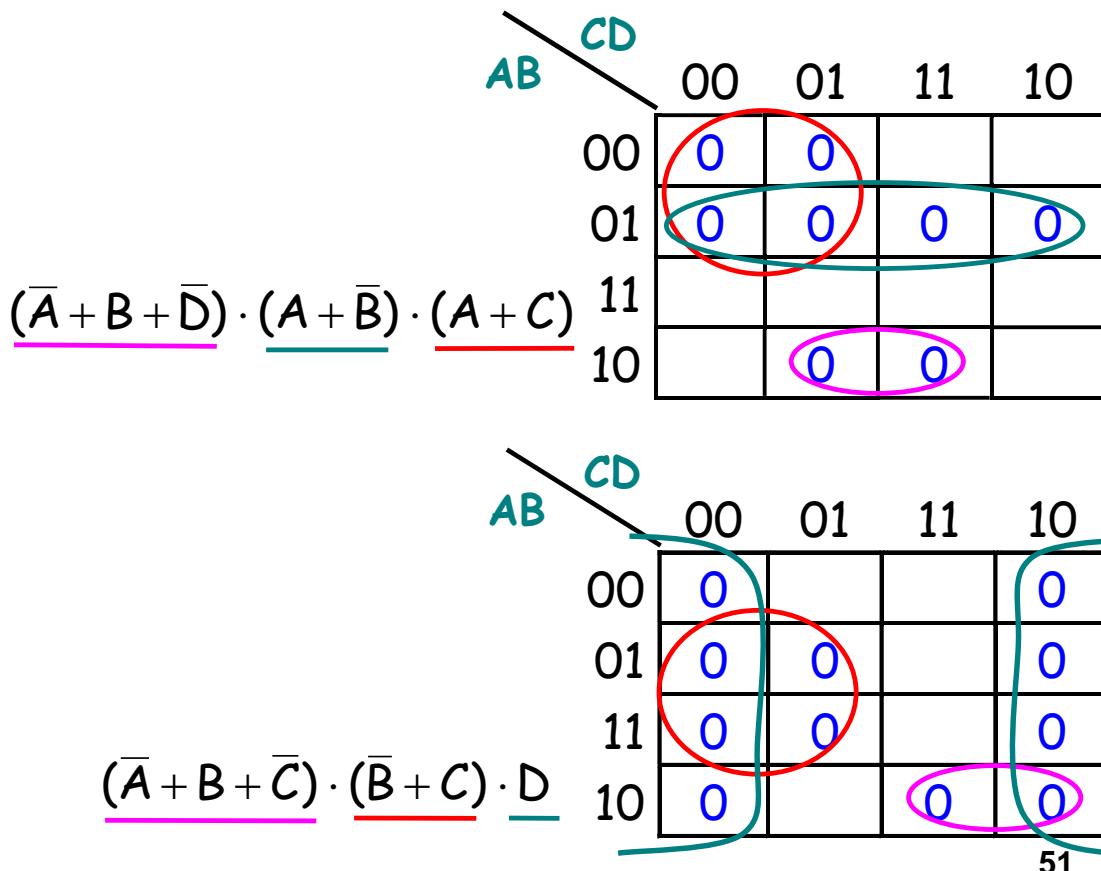
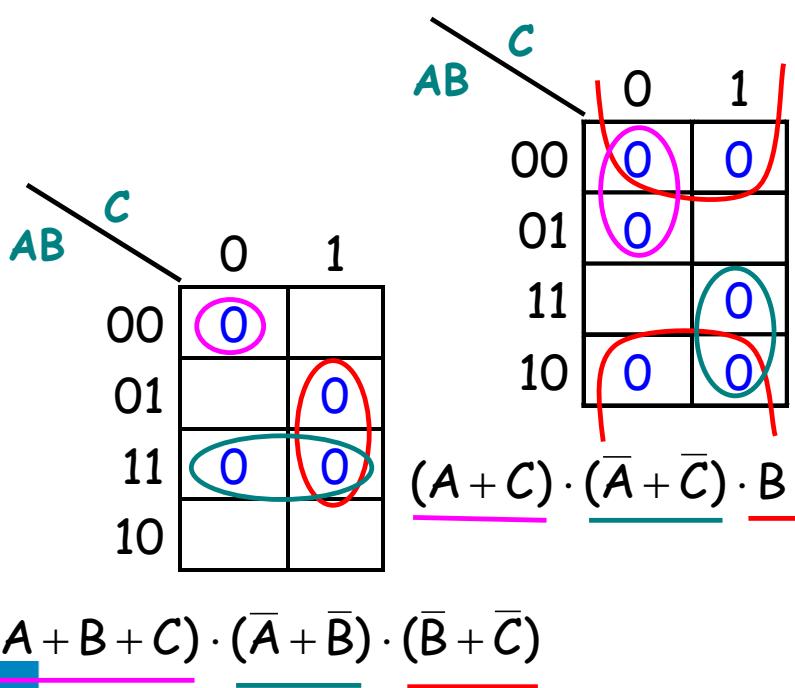
$$\underline{\overline{ABD}} + \underline{\overline{ABD}} + \underline{\overline{ABE}} + \underline{DE} + \underline{\overline{BCE}} + \underline{\overline{BCDE}}$$

Funciones lógicas

- Diagramas de Karnaugh. Simplificación de un producto de sumas
 - El proceso de minimización de un producto de sumas es básicamente el mismo que para una expresión suma de productos, salvo que ahora hay que agrupar los 0's para generar el mínimo número de términos suma
 - Las reglas para agrupar los 0's son las mismas que para agrupar los 1's

Funciones lógicas

- Diagramas de Karnaugh. Simplificación de un producto de sumas
 - Determinación de la expresión producto de sumas mínima a partir del diagrama. Producto de los términos suma obtenidos



Funciones lógicas

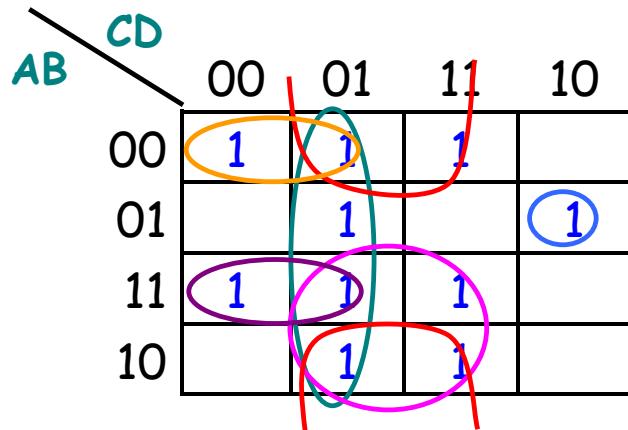
- Obtención del diagrama de Karnaugh a partir de la tabla de verdad

- Ejemplo:

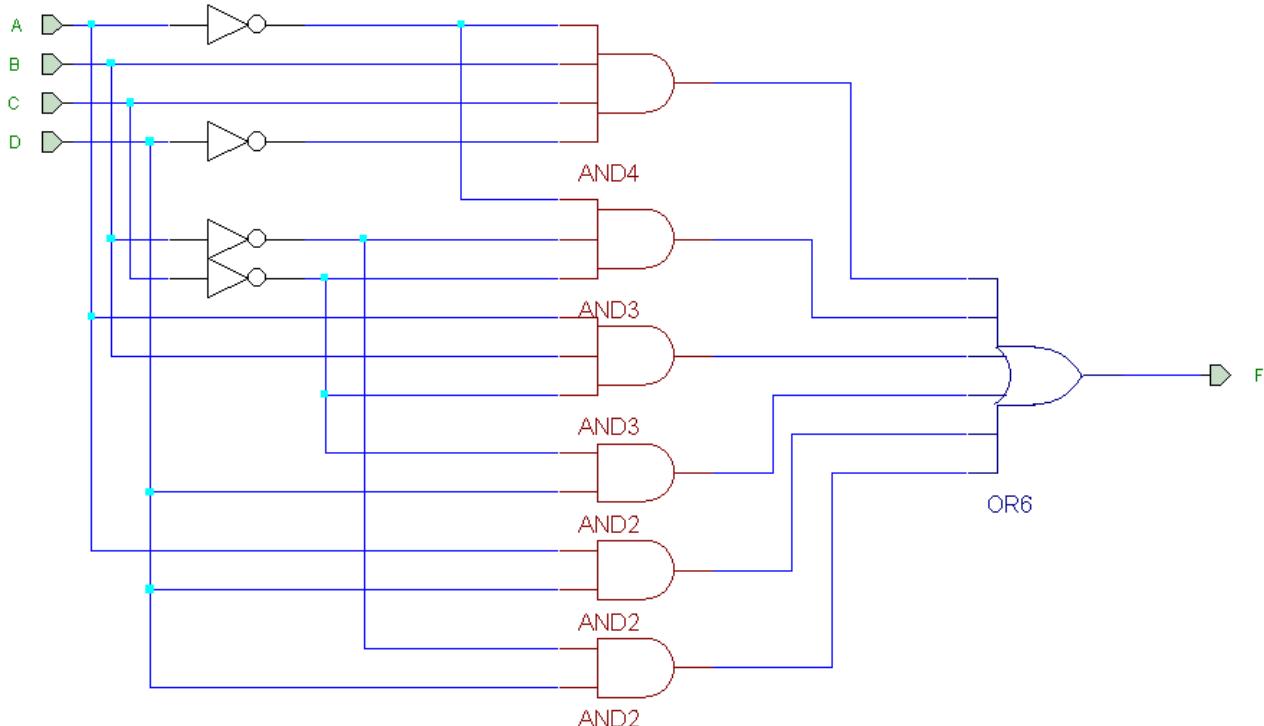
Nº	A	B	C	D	F	Minterms	Maxterms
0	0	0	0	0	1	$\rightarrow (\bar{A} \cdot \bar{B} \cdot \bar{C} \cdot \bar{D})$	
1	0	0	0	1	1	$\rightarrow (\bar{A} \cdot \bar{B} \cdot \bar{C} \cdot D)$	
2	0	0	1	0	0		$\rightarrow (A + B + \bar{C} + D)$
3	0	0	1	1	1	$\rightarrow (\bar{A} \cdot \bar{B} \cdot C \cdot D)$	
4	0	1	0	0	0		$\rightarrow (A + \bar{B} + C + D)$
5	0	1	0	1	1	$\rightarrow (\bar{A} \cdot B \cdot \bar{C} \cdot D)$	
6	0	1	1	0	1	$\rightarrow (\bar{A} \cdot B \cdot C \cdot \bar{D})$	
7	0	1	1	1	0		$\rightarrow (A + \bar{B} + \bar{C} + \bar{D})$
8	1	0	0	0	0		$\rightarrow (\bar{A} + B + C + D)$
9	1	0	0	1	1	$\rightarrow (A \cdot \bar{B} \cdot \bar{C} \cdot D)$	
10	1	0	1	0	0		$\rightarrow (\bar{A} + B + \bar{C} + D)$
11	1	0	1	1	1	$\rightarrow (A \cdot \bar{B} \cdot C \cdot D)$	
12	1	1	0	0	1	$\rightarrow (A \cdot B \cdot \bar{C} \cdot \bar{D})$	
13	1	1	0	1	1	$\rightarrow (A \cdot B \cdot \bar{C} \cdot D)$	
14	1	1	1	0	0		$\rightarrow (\bar{A} + \bar{B} + \bar{C} + D)$
15	1	1	1	1	1	$\rightarrow (A \cdot B \cdot C \cdot D)$	

Funciones lógicas

- Obtención del diagrama de Karnaugh a partir de la tabla de verdad
 - Ejemplo: Desarrollo por minterms



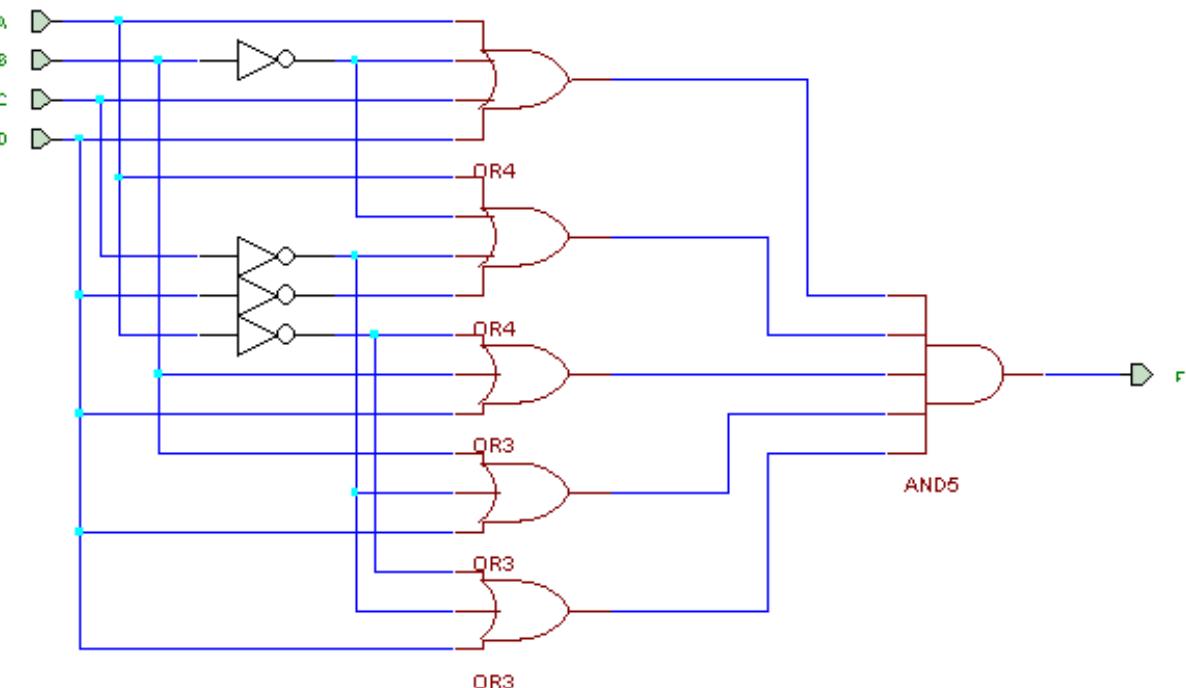
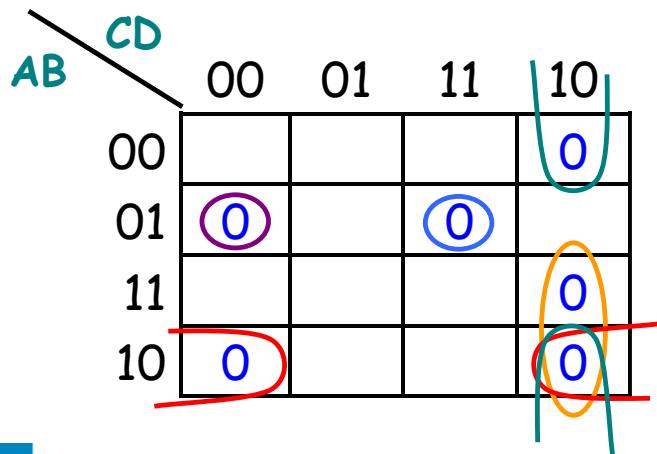
$$F = \overline{ABC}\overline{D} + \overline{ABC}\overline{C} + ABC\overline{C} + \overline{CD} + AD + \overline{BD}$$



Funciones lógicas

- Obtención del diagrama de Karnaugh a partir de la tabla de verdad
 - Ejemplo: Desarrollo por maxterms

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



Pasos para resolver cualquier función lógica

Funciones completamente especificadas

- Representar la tabla de verdad de la función
- Simplificar la función a su expresión más sencilla (Diagrama de Karnaugh).
- Construir el circuito combinacional con el número menor de puertas.

Ejemplo 1: Sea un circuito que tiene 4 entradas que representan un número binario (ABCD). El circuito dispone de una salida Z_1 que debe activarse ($Z_1=1$) cuando el número binario sea múltiplo de 3, y una salida Z_2 que debe activarse cuando el número binario sea múltiplo de 2.

Pasos para resolver cualquier función lógica

- Ejemplo 1:

Nº	A	B	C	D	Z ₁	Z ₂
0	0	0	0	0	1	1
1	0	0	0	1	0	0
2	0	0	1	0	0	1
3	0	0	1	1	1	0
4	0	1	0	0	0	1
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	0	1
9	1	0	0	1	1	0
10	1	0	1	0	0	1
11	1	0	1	1	0	0
12	1	1	0	0	1	1
13	1	1	0	1	0	0
14	1	1	1	0	0	1
15	1	1	1	1	1	0

Pasos para resolver cualquier función lógica

- Ejemplo 1:

Nº	A	B	C	D	Z_1	Z_2
0	0	0	0	0	1	1
1	0	0	0	1	0	0
2	0	0	1	0	0	1
3	0	0	1	1	1	0
4	0	1	0	0	0	1
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	0	1
9	1	0	0	1	1	0
10	1	0	1	0	0	1
11	1	0	1	1	0	0
12	1	1	0	0	1	1
13	1	1	0	1	0	0
14	1	1	1	0	0	1
15	1	1	1	1	1	0

AB \ CD

00	01	11	10
(1)		(1)	
			(1)
(1)		(1)	
	(1)		

$$Z_1 = \overline{ABC}\overline{D} + \overline{AB}\overline{C}\overline{D} + \overline{ABC}\overline{D} + A\overline{B}\overline{C}\overline{D} + AB\overline{CD} + A\overline{BC}\overline{D}$$

AB \ CD

00	01	11	10
(1)			(1)
(1)			(1)
(1)			(1)
(1)			(1)

$$Z_2 = \overline{D}$$

Pasos para resolver cualquier función lógica

- Funciones incompletamente especificadas
 - Funciones que no tienen definidos todos sus estados.
 - Los estados no definidos se representan como “X” a la salida.
 - A la hora de resolver la tabla de Karnaugh las “X” se toman como 0 o 1 según convenga.

Pasos para resolver cualquier función lógica

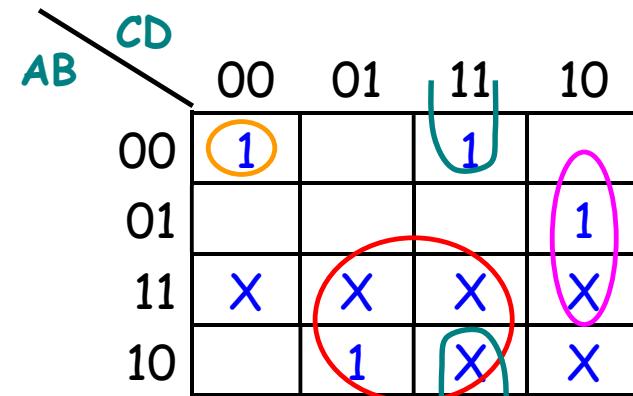
- Funciones incompletamente especificadas
 - Ejemplo2: Ejemplo1 pero definiendo sólo de 0 a 9

Nº	A	B	C	D	Z_1	Z_2
0	0	0	0	0	1	1
1	0	0	0	1	0	0
2	0	0	1	0	0	1
3	0	0	1	1	1	0
4	0	1	0	0	0	1
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	0	1
9	1	0	0	1	1	0
10	1	0	1	0	X	X
11	1	0	1	1	X	X
12	1	1	0	0	X	X
13	1	1	0	1	X	X
14	1	1	1	0	X	X
15	1	1	1	1	X	X

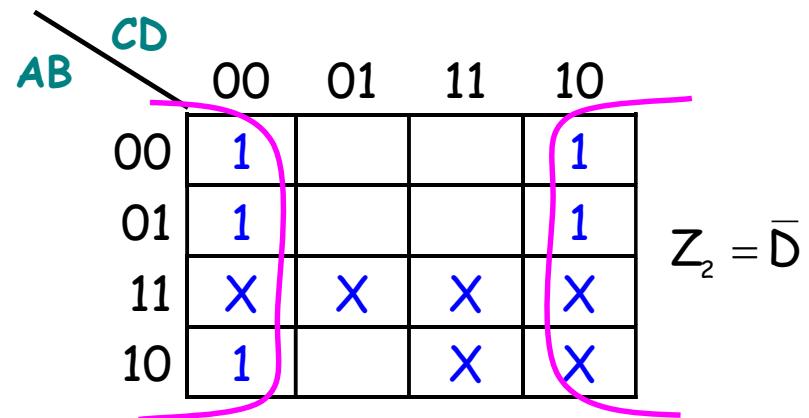
Pasos para resolver cualquier función lógica

- Funciones incompletamente especificadas
 - Ejemplo2: Ejemplo1 pero definiendo sólo de 0 a 9

Nº	A	B	C	D	Z_1	Z_2
0	0	0	0	0	1	1
1	0	0	0	1	0	0
2	0	0	1	0	0	1
3	0	0	1	1	1	0
4	0	1	0	0	0	1
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	0	1
9	1	0	0	1	1	0
10	1	0	1	0	X	X
11	1	0	1	1	X	X
12	1	1	0	0	X	X
13	1	1	0	1	X	X
14	1	1	1	0	X	X
15	1	1	1	1	X	X



$$Z_1 = \overline{AB}\overline{CD} + \overline{BCD} + B\overline{CD} + AD$$

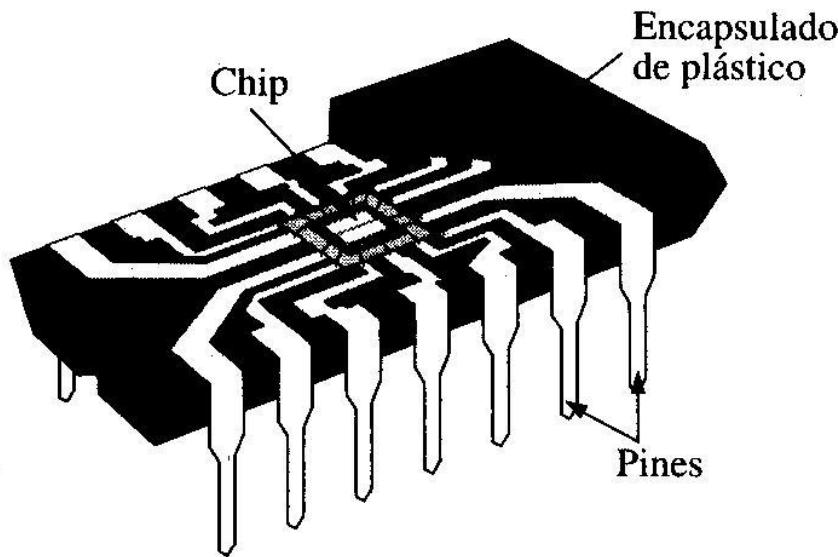


Anexo. Circuitos digitales integrados

- Los elementos y funciones lógicas vistos están disponibles como circuitos integrados (CI)
- Un circuito integrado monolítico es un circuito electrónico construido enteramente sobre un pequeño chip de silicio
- Todos los componentes del circuito (transistores, diodos, resistencias y condensadores) son parte integrante de un único chip

Anexo. Circuitos digitales integrados

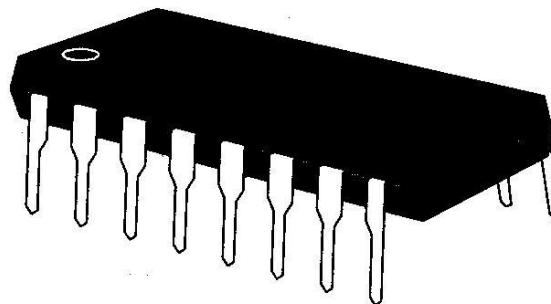
- Sección de un encapsulado de CI:



Anexo. Circuitos digitales integrados

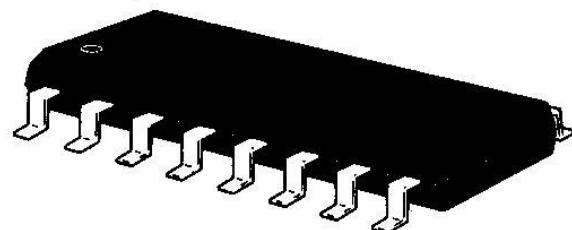
- Encapsulados de los CI

- Se clasifican según la forma en que se montan sobre las tarjetas de circuito impreso
- Pueden ser de inserción o de montaje superficial
- **Encapsulados de inserción:**
 - Los pines (patas) se insertan en los taladros de la tarjeta de circuito impreso y se sueldan a las pistas por la cara opuesta
 - El más típico: **DIP** (dual in-line package)



Anexo. Circuitos digitales integrados

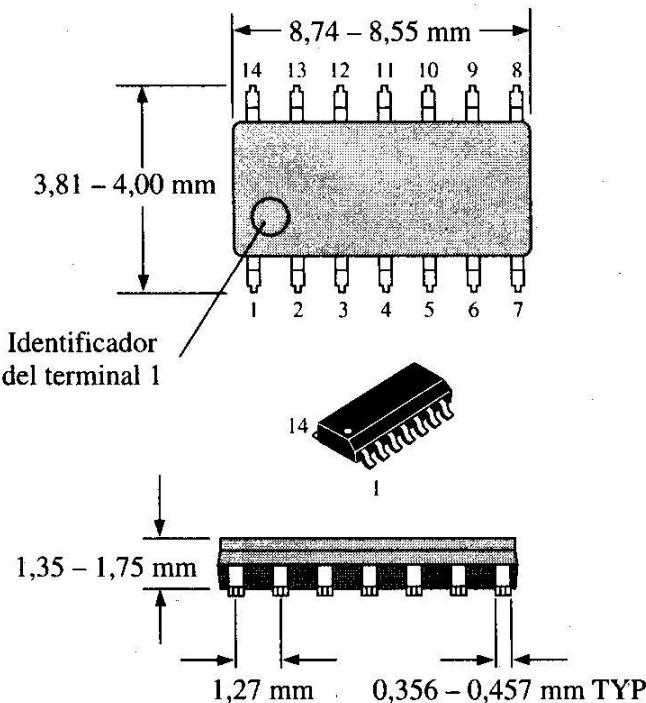
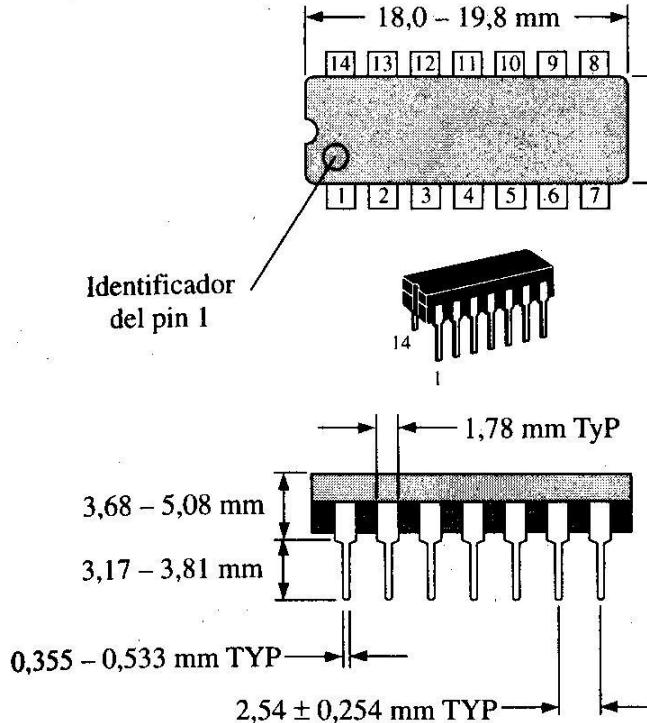
- Encapsulados de los CI
 - Encapsulados de montaje superficial:
 - Método más moderno, ahorra espacio
 - No necesita taladros en las tarjetas de circuito impreso. Los pines se sueldan directamente a las pistas de una cara de la tarjeta, dejando la otra libre para añadir circuitos adicionales
 - Menos tamaño que el DIP, los pines están más cercanos
 - El más típico: SOIC (small-outline IC)



Anexo. Circuitos digitales integrados

- Encapsulados de los CI

- Encapsulados típicos DIP y SOIC con sus dimensiones básicas y la numeración de los pines



Anexo. Circuitos digitales integrados

- Diagramas de configuración de los pines para las puertas lógicas más comunes

