Arquitectura de Computadoras para Ingeniería

(Cód. 7526)
1° Cuatrimestre 2016

Dra. Dana K. Urribarri
DCIC - UNS

Técnicas digitales

Álgebra de Boole

- Un álgebra de Boole es el orden parcial de los subconjuntos definidos por inclusión.
- El álgebra de Boole *b*(*A*) de un conjunto *A* es el conjunto de subconjuntos de *A* que se pueden obtener a través de un número finitos de operaciones de unión (OR), intersección (AND) y complemento (NOT).

Álgebra de Boole

Un álgebra de Boole es un conjunto no vacío A, junto a dos operaciones $\wedge$ y $\vee$ (definidas sobre A), una operación unaria ' y dos elementos particulares 0 y 1 ($0 \in A$ y $1 \in A$), tales que satisfacen los siguientes axiomas $\forall p,q,r \in A$:

- Ley de identidad:

  $p \wedge 1 = p$ y $p \vee 0 = p$

- Existencia de complemento:

  $p \wedge p' = 0$ y $p \vee p' = 1$

- Ley conmutativa:

  $p \wedge q = q \wedge p$ y $p \vee q = q \vee p$

- Leyes distributivas:

  $p \wedge (q \vee r) = (p \wedge q) \vee (p \wedge r)$ y $p \vee (q \wedge r) = (p \vee q) \wedge (p \vee r)$.

Álgebra de Boole

Se pueden probar los siguientes teoremas:

- Ley de idempotencia:

  $p \wedge p = p \vee p = p$

- Ley del doble complemento *(involución)*:

  $(p')' = p$

- Ley del complemento:

  $p \vee p' = 1$ y $p \wedge p' = 0$

- Leyes asociativas:

  $p \wedge (q \wedge r) = (p \wedge q) \wedge r$ y $p \vee (q \vee r) = (p \vee q) \vee r$

- Leyes de De Morgan:

  $(p \wedge q)' = p' \vee q'$ y $(p \vee q)' = p' \wedge q'$

- Ley de Absorción:

  $p \wedge (p \vee q) = p$ y $p \vee (p \wedge q) = p$

- $0' = 1$ y $1' = 0$

Álgebra de Boole

En 1938 Shannon demostró que el álgebra de Boole de dos valores (0 y 1 o *falso* y *verdadero*) podía ser utilizado para el análisis y síntesis de la conmutación y circuitos digitales. Además, cómo la combinación de estos podía representar operaciones aritméticas y lógicas complejas.

Variables y literales

- *Variable*: cada elemento de la expresión con distinto nombre.
  - A . B' + A' C + A (D+E) → 5 variables

  Determinan los grados de libertad de la expresión.

- *Literal*: cada aparición de una variable o de su complemento.
  - A . B' + A' C + A (D+E) → 7 literales

  Determinan la complejidad de la expresión.

Principio de dualidad

Cualquier expresión algebraica deducible de los postulados del álgebra de Boole permanece válida si los operadores binarios y los elementos identidad son intercambiados.

¿Cómo se obtiene el dual?

- Se intercambian ANDs y ORs
- Dual[(a+b') · (c·b+d)] = (a·b') + ((c+b) · d)

Principio de dualidad

$$Dual[\ (a+b') \cdot (c \cdot b+d)\ ] = (a \cdot b') + ((c+b) \cdot d)$$

No hay relación lógica entre una expresión y su dual.

Si dos expresiones son equivalente, entonces sus duales también lo son.

Complemento de una expresión

- A través de la ley de De Morgan

$$F_1 = \overline{X}YZ + \overline{X}\,\overline{Y}Z$$

$$\overline{F}_1 = \overline{\overline{X}YZ + \overline{X}\,\overline{Y}Z} = (\overline{\overline{X}YZ}) \cdot (\overline{\overline{X}\,\overline{Y}Z})$$

$$= (X + \overline{Y} + Z)(X + Y + \overline{Z})$$

Complemento de una expresión

- A través del dual y complementando cada literal

- El dual de F_1 es

$$F_1 = \overline{X}Y\overline{Z} + \overline{X}\,\overline{Y}Z = (\overline{X}Y\overline{Z}) + (\overline{X}\,\overline{Y}Z)$$

- Complementando cada literal

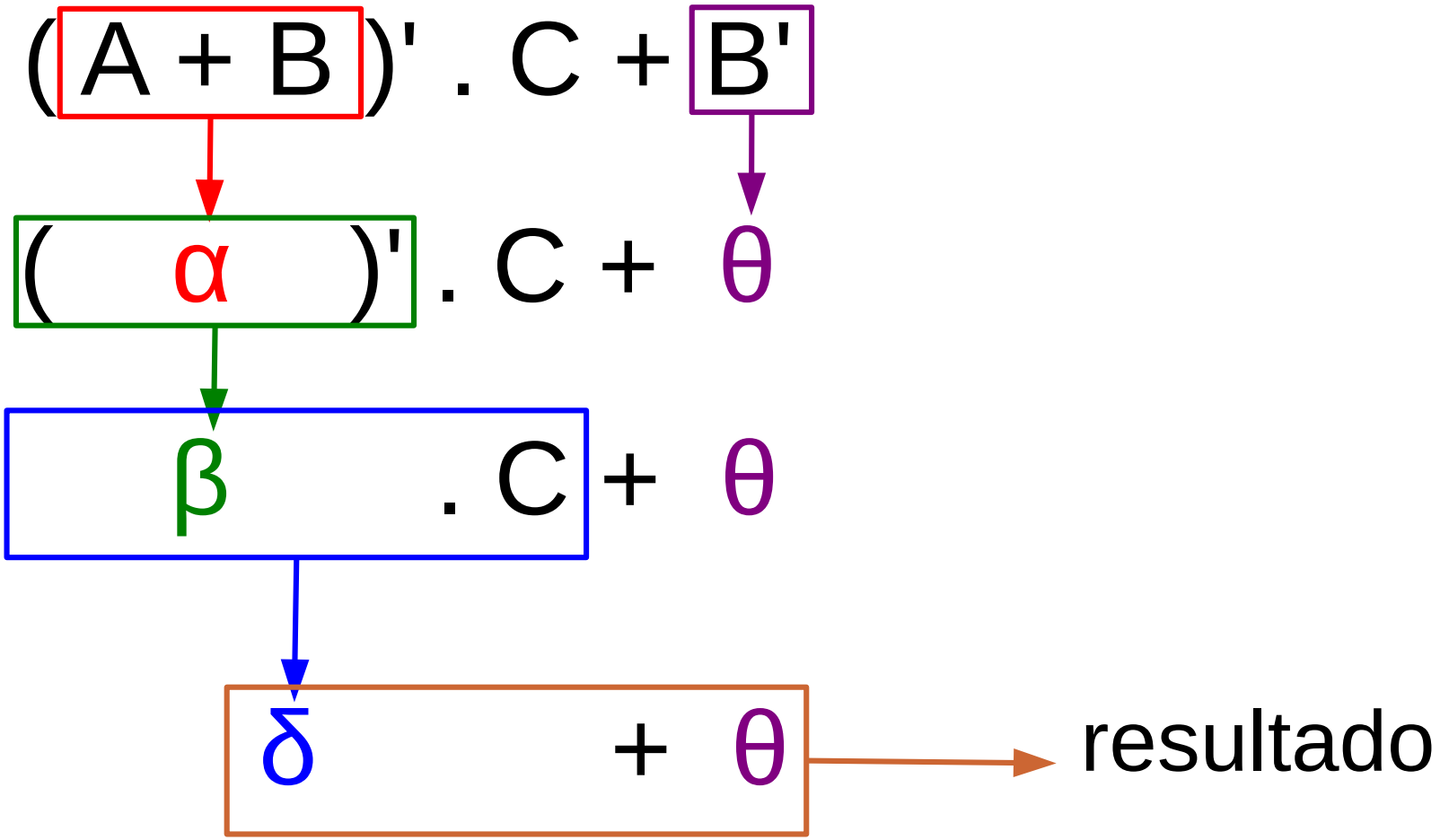
$$(\overline{X} + Y + \overline{Z})(\overline{X} + \overline{Y} + Z)$$

$$(X + \overline{Y} + Z)(X + Y + \overline{Z}) = \overline{F}_1$$

Precedencia de los operadores

- El orden de precedencia de los operadores es:
  1) Paréntesis
  2) Complemento
  3) And
  4) Or

Precedencia de los operadores



Álgebra de Boole bivaluada

- Un álgebra de Boole bivaluada *(en adelante álgebra de Boole)* se define sobre un conjunto de dos elementos B={0,1}.
- Los operadores binarios son · (AND) y **+** (OR).
- El operador unario es **'** (NOT).

- AND: ·, ∧, ∩
- OR: +, ∨, ∪
- NOT: ¯, '

x	y	$x \cdot y$
0	0	0
0	1	0
1	0	0
1	1	1

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

x	x'
0	1
1	0

Tabla de verdad

(a) $F = \overline{X}YZ + \overline{X}Y\overline{Z} + XZ$

(b) $F = \overline{X}Y + XZ$

X	Y	Z	(a) F	(b) F
0	0	0	0	0
0	0	1	0	0
0	1	0	1	1
0	1	1	1	1
1	0	0	0	0
1	0	1	1	1
1	1	0	0	0
1	1	1	1	1

Formas estándares

- Término suma:
  - Suma lógica. Consiste de la operación OR entre literales
  - X + Y + Z'
- Término producto:
  - Producto lógico. Consiste de la operación AND entre los literales
  - XYZ'

Observación: ¡suma y producto NO hacen referencia a las operaciones aritméticas!

Formas canónicas y estándares

- Formas canónicas
  - Suma expandida de productos
  - Producto expandido de sumas
- Formas estándares
  - Mínima suma de productos
  - Mínimo producto de sumas

Minitérminos y maxitérminos

- Minitérmino:
  - *Término producto* donde todas las variables aparecen exactamente una vez.
    Pueden estar complementadas o no.

- Maxitérmino
  - *Término suma* donde todas las variables a parecen exactamente una vez.
    Pueden estar complementadas o no.

Minitérminos y maxitérminos

- Para *n* variables hay 2^n minitérminos y 2^n maxitérminos posibles.
- Para 2 variables A e B
  - Los minitérminos posibles son AB, AB', A'B y A'B'
  - Los maxitérminos posibles son A+B, A'+B, A+B' y A'+B'
- M_j denota el maxitérmino para el cual su combinación binaria se corresponde al decimal *j*.
- m_j denota el minitérmino para el cual su combinación binaria se corresponde al decimal *j*.

Minitérminos

- Minitérminos para tres variables

X	Y	Z	Product Term	Symbol	m_0	m_1	m_2	m_3	m_4	m_5	m_6	m_7
0	0	0	$\overline{X}\,\overline{Y}\,\overline{Z}$	m_0	1	0	0	0	0	0	0	0
0	0	1	$\overline{X}\,\overline{Y}Z$	m_1	0	1	0	0	0	0	0	0
0	1	0	$\overline{X}Y\overline{Z}$	m_2	0	0	1	0	0	0	0	0
0	1	1	$\overline{X}YZ$	m_3	0	0	0	1	0	0	0	0
1	0	0	$X\overline{Y}\,\overline{Z}$	m_4	0	0	0	0	1	0	0	0
1	0	1	$X\overline{Y}Z$	m_5	0	0	0	0	0	1	0	0
1	1	0	$XY\overline{Z}$	m_6	0	0	0	0	0	0	1	0
1	1	1	XYZ	m_7	0	0	0	0	0	0	0	1

- Notar que m_j vale 1 para la combinación j y 0 para todas las demás

Minitérminos

Una función booleana puede representarse algebraicamente a través de la tabla de verdad formando la suma de todos los minitérminos que producen un 1 en la función.

X	Y	Z	F	
0	0	0	1	$m_0 = X'Y'Z'$
0	0	1	0	
0	1	0	1	$m_2 = X'YZ'$
0	1	1	0	
1	0	0	0	
1	0	1	1	$m_5 = XY'Z$
1	1	0	0	
1	1	1	1	$m_7 = XYZ$

$$F = \overline{X}\overline{Y}\overline{Z} + \overline{X}Y\overline{Z} + X\overline{Y}Z + XYZ$$
$$= m_0 + m_2 + m_5 + m_7$$

$$F(X, Y, Z) = \Sigma m(0, 2, 5, 7)$$

Suma expandida de productos

Maxitérminos

- Maxitérminos para tres variables

X	Y	Z	Sum Term	Symbol	M_0	M_1	M_2	M_3	M_4	M_5	M_6	M_7
0	0	0	$X + Y + Z$	M_0	0	1	1	1	1	1	1	1
0	0	1	$X + Y + \overline{Z}$	M_1	1	0	1	1	1	1	1	1
0	1	0	$X + \overline{Y} + Z$	M_2	1	1	0	1	1	1	1	1
0	1	1	$X + \overline{Y} + \overline{Z}$	M_3	1	1	1	0	1	1	1	1
1	0	0	$\overline{X} + Y + Z$	M_4	1	1	1	1	0	1	1	1
1	0	1	$\overline{X} + Y + \overline{Z}$	M_5	1	1	1	1	1	0	1	1
1	1	0	$\overline{X} + \overline{Y} + Z$	M_6	1	1	1	1	1	1	0	1
1	1	1	$\overline{X} + \overline{Y} + \overline{Z}$	M_7	1	1	1	1	1	1	1	0

- Notar que M_j vale 0 para la combinación j y 1 para todas las demás

Maxitérminos

También puede representarse a través del producto de todos los maxitérminos que produzcan un 0 en la función.

Para esto hallamos la suma de productos del complemento de F, F'.

Maxitérminos

X	Y	Z	F	$\overline{F}$	
0	0	0	1	0	
0	0	1	0	1	$m_1 = X'Y'Z$
0	1	0	1	0	
0	1	1	0	1	$m_3 = X'YZ$
1	0	0	0	1	$m_4 = XY'Z'$
1	0	1	1	0	
1	1	0	0	1	$m_6 = XYZ'$
1	1	1	1	0	

$$\overline{F}(X,Y,Z) = \overline{X}\,\overline{Y}Z + \overline{X}YZ + X\overline{Y}\,\overline{Z} + XY\overline{Z}$$
$$= m_1 + m_3 + m_4 + m_6$$

$$\overline{F}(X, Y, Z) = \Sigma m(1, 3, 4, 6)$$

Notar la relación entre

$$F(X, Y, Z) = \Sigma m(0, 2, 5, 7)$$

y

$$\overline{F}(X, Y, Z) = \Sigma m(1, 3, 4, 6)$$

Maxitérminos

- Luego F = (F')'. Aplicando De Morgan:

$$F = \overline{m_1 + m_3 + m_4 + m_6} = \overline{m_1} \cdot \overline{m_3} \cdot \overline{m_4} \cdot \overline{m_6}$$
$$= M_1 \cdot M_3 \cdot M_4 \cdot M_6 \,(\text{since } \overline{m_j} = M_j)$$
$$= (X + Y + \overline{Z})(X + \overline{Y} + \overline{Z})(\overline{X} + Y + Z)(\overline{X} + \overline{Y} + Z)$$

$$F(X, Y, Z) = \prod M(1, 3, 4, 6)$$

Producto expandido de sumas

A partir de la tabla de verdad

X	Y	Z	F	
0	0	0	1	$m_0 = X'Y'Z'$
0	0	1	0	$M_1 = X+Y+Z'$
0	1	0	1	$m_2 = X'YZ'$
0	1	1	0	$M_3 = X+Y'+Z'$
1	0	0	0	$M_4 = X'+Y+Z$
1	0	1	1	$m_5 = XY'Z$
1	1	0	0	$M_6 = X'+Y'+Z$
1	1	1	1	$m_7 = XYZ$

Propiedades de los minitérminos

1) Hay 2^n minitérminos para *n* variables Booleanas. Estos minitérminos se pueden generar a partir de los números binarios entre 0 y 2^n-1

2) Cualquier función booleana puede expresarse como suma expandida de productos

3) El complemento de una función contiene los minitérminos no incluidos en la función original.

4) La función que incluye todos los 2^n minitérminos es lógicamente igual a 1.

Suma de productos

- Suma lógica de términos productos. Cada términos producto puede tener cualquier cantidad de literales.
- Dado $E = Y' + X'Z'$ no está expresado como suma expandida de productos.
- Puede expandirse:
  - a través de la tabla de verdad
  - algebraicamente.

Suma de productos

X	Y	Z	E	
0	0	0	1	$m_0 = \bar{X}\,\bar{Y}\,\bar{Z}$
0	0	1	1	$m_1 = \bar{X}\,\bar{Y}Z$
0	1	0	1	$m_2 = X\bar{Y}Z$
0	1	1	0	
1	0	0	1	$m_4 = X\bar{Y}\,\bar{Z}$
1	0	1	1	$m_5 = X\bar{Y}Z$
1	1	0	0	
1	1	1	0	

$$E(X, Y, Z) = \Sigma m(0, 1, 2, 4, 5)$$

Suma de productos

- Algebraicamente:

$$E = Y' + X'Z'$$
$$= (X + X')Y' + X'(Y + Y')Z'$$
$$= XY' + X'Y' + X'YZ' + X'Y'Z'$$
$$= XY'(Z+Z') + X'Y'(Z+Z') + X'YZ' + X'Y'Z'$$
$$= XY'Z+XY'Z' + X'Y'Z+X'Y'Z' + X'YZ' + X'Y'Z'$$

Producto de sumas

- Producto lógico de sumas. Cada suma puede tener cualquier cantidad de literales.

$$F = X(\overline{Y} + Z)(X + Y + \overline{Z})$$

- Se puede expandir utilizando tanto la tabla de verdad como algebraicamente.

Valores de la salida

- Dada una combinación de entrada, hay 3 posibles valores de salida:
  - Que esa combinación valide la salida (valor 1)
  - Que esa combinación invalide la salida (valor 0)
  - Que no importe el valor de salida (opcional, *don't care*)
- La salida podrá ser opcional si:
  - Para ciertas combinaciones de entrada no importe el valor de salida.
  - Ciertas combinaciones de entrada son imposibles.

Compuertas

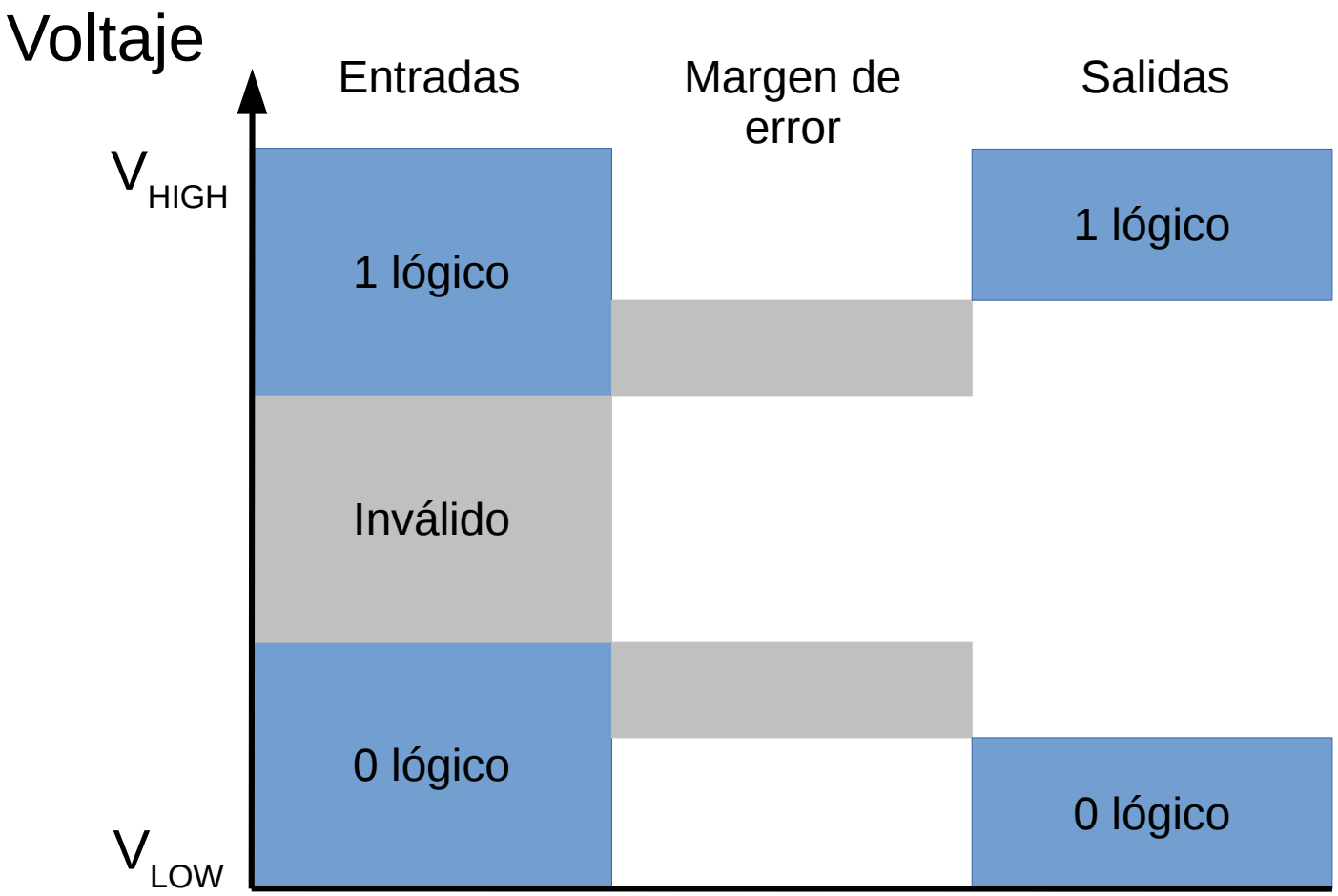
Compuertas

- Compuerta (*gate*): circuito electrónico que opera sobre una o más señales de entrada para proveer una señal de salida.
- El voltaje y la corriente son señales analógicas, toman valores de un rango continuo.
- Los sistemas digitales responden a dos niveles de voltaje bien diferenciados: V_{HIGH} y V_{LOW}. Uno se corresponderá al valor 1 lógico y el otro, al 0 lógico.

Compuertas

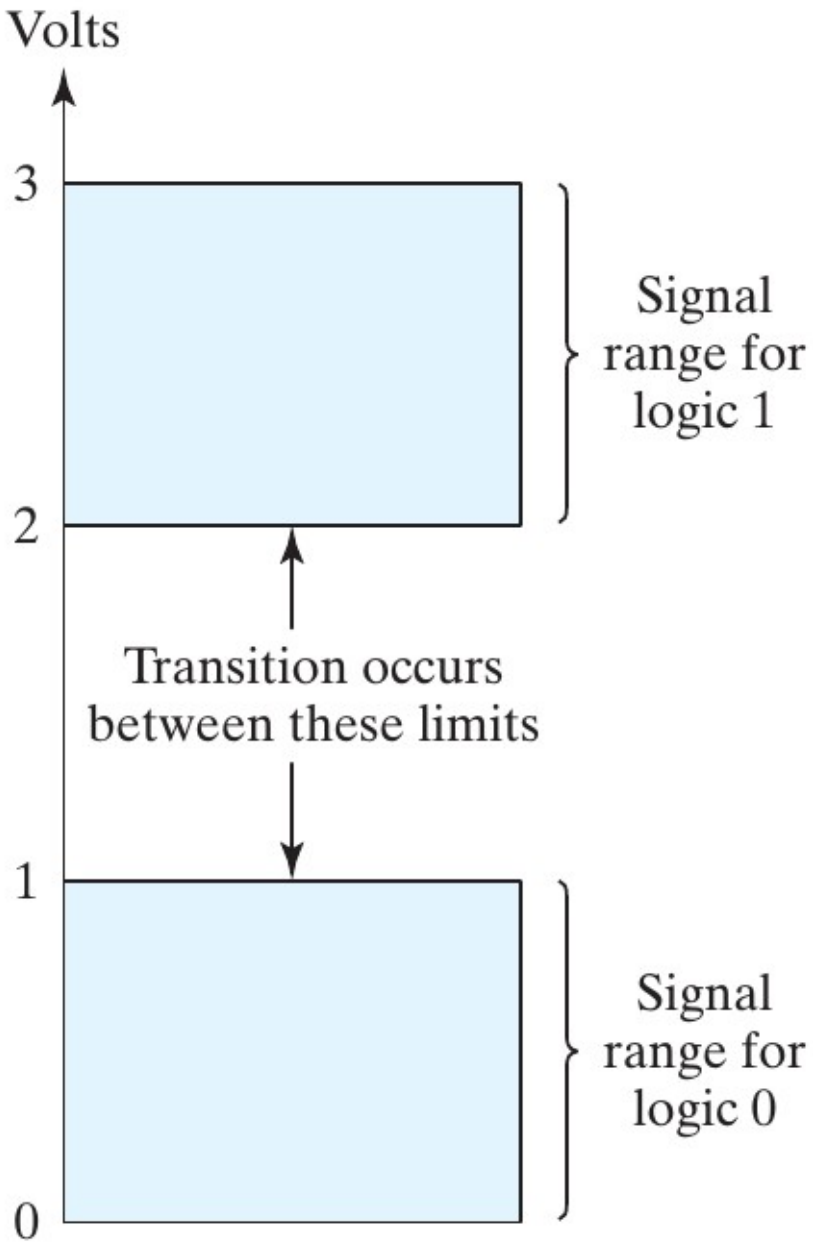
- Se asocia un rango de valores analógicos a cada valor lógico (0 o 1).
- Una compuerta típica no garantiza niveles de voltaje perfectos.
- Puede producir un voltaje dentro de un subrango que garantiza ser reconocido por la compuerta de entrada.
- La diferencia entre los rangos se denomina margen de ruido.

Compuertas


Voltaje
Entradas
Margen de error
Salidas
V HIGH
1 lógico
Inválido
0 lógico
V LOW
1 lógico
0 lógico

Compuertas

- Los cambios de voltaje no son instantáneos.



Lógica de entrada y salida

- Las compuertas AND, OR y NOT entienden tensiones V_{HIGH} y V_{LOW}
- El voltaje de salida en respuesta a los voltajes de entrada están fijados pero no la correspondencia lógica.
- Lógica Positiva (LP):
  - $V_H \rightarrow 1$
  - $V_L \rightarrow 0$
- Lógica Negativa (LN):
  - $V_H \rightarrow 0$
  - $V_L \rightarrow 1$

Lógica de entrada y salida

- Tenemos una compuerta con el siguiente comportamiento eléctrico:

A	B	Salida
V_H	V_H	V_H
V_H	V_L	V_L
V_L	V_H	V_L
V_L	V_L	V_L

Lógica de entrada y salida

- Lógica positiva

A	B	Salida
1	1	1
1	0	0
0	1	0
0	0	0

- AND

La especificación de los fabricantes es siempre en lógica positiva.

- Lógica negativa

A	B	Salida
0	0	0
0	1	1
1	0	1
1	1	1

- OR

Lógica de entrada y salida

- ¿Lógica mixta?
- Entradas y salidas con diferente lógica:
  - Entradas lógica positiva
  - Salidas lógica negativa

    o
  - Entradas lógica negativa
  - Salidas lógica positiva

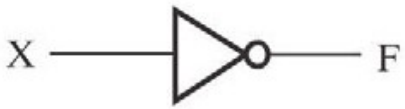
Lógica positiva		LN
A	**B**	**Salida**
1	1	0
1	0	1
0	1	1
0	0	1

NAND

Lógica negativa		LP
A	**B**	**Salida**
0	0	1
0	1	0
1	0	0
1	1	0

NOR

Compuertas más comunes

Name	Distinctive-Shape Graphics Symbol	Algebraic Equation	Truth Table
AND		$F = XY$	X Y | F 0 0 | 0 0 1 | 0 1 0 | 0 1 1 | 1
OR		$F = X + Y$	X Y | F 0 0 | 0 0 1 | 1 1 0 | 1 1 1 | 1
NOT (inverter)		$F = \overline{X}$	X | F 0 | 1 1 | 0

Compuertas más comunes

NAND

$F = \overline{X \cdot Y}$

X	Y	F
0	0	1
0	1	1
1	0	1
1	1	0

NOR

$F = \overline{X + Y}$

X	Y	F
0	0	1
0	1	0
1	0	0
1	1	0

Exclusive-OR (XOR)

$F = X\overline{Y} + \overline{X}Y$
$= X \oplus Y$

X	Y	F
0	0	0
0	1	1
1	0	1
1	1	0

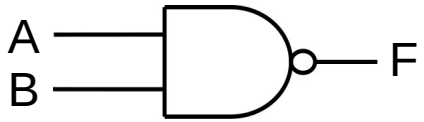
Exclusive-NOR (XNOR)

$F = \overline{XY + \overline{X}\overline{Y}}$
$= X \oplus Y$

X	Y	F
0	0	1
0	1	0
1	0	0
1	1	1

¿Qué compuertas vamos a usar?

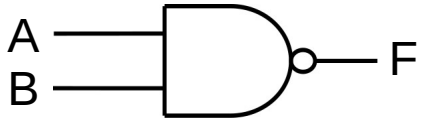
NAND



Comportamiento eléctrico		
A	**B**	**Salida**
V_H	V_H	V_L
V_H	V_L	V_H
V_L	V_H	V_H
V_L	V_L	V_H

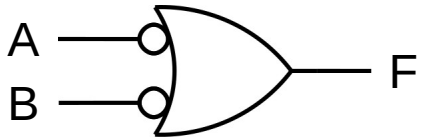
Lógica positiva		LN
A	**B**	**Salida**
1	1	1
1	0	0
0	1	0
0	0	0

AND



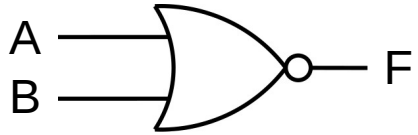
Lógica negativa		LP
A	**B**	**Salida**
0	0	0
0	1	1
1	0	1
1	1	1

OR



¿Qué compuertas vamos a usar?

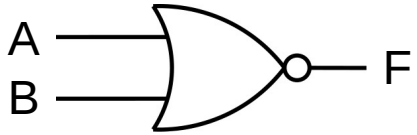
NOR



Comportamiento eléctrico		
A	**B**	**Salida**
V_H	V_H	V_L
V_H	V_L	V_L
V_L	V_H	V_L
V_L	V_L	V_H

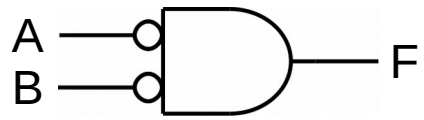
Lógica positiva		LN
A	**B**	**Salida**
1	1	1
1	0	1
0	1	1
0	0	0

OR



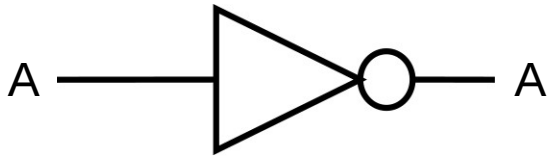
Lógica negativa		LP
A	**B**	**Salida**
0	0	0
0	1	0
1	0	0
1	1	1

AND



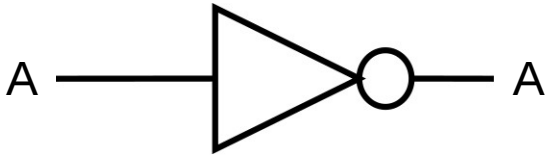
¿Qué compuertas vamos a usar?

INV

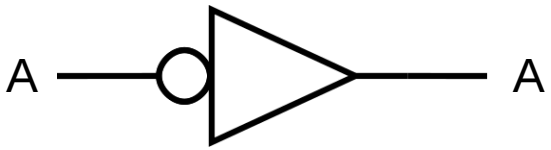


Comportamiento eléctrico	
A	**Salida**
V_H	V_L
V_L	V_H

LP	LN
A	**Salida**
1	1
0	0



LN	LP
A	**Salida**
0	0
1	1



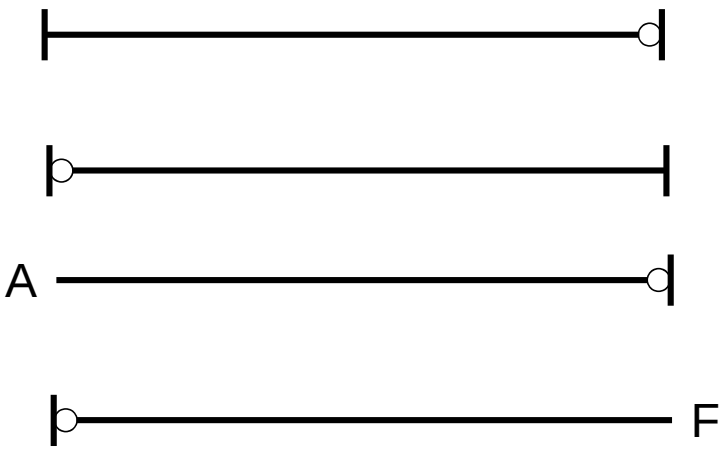
No invierten el valor.

Realizan un cambio de lógica.

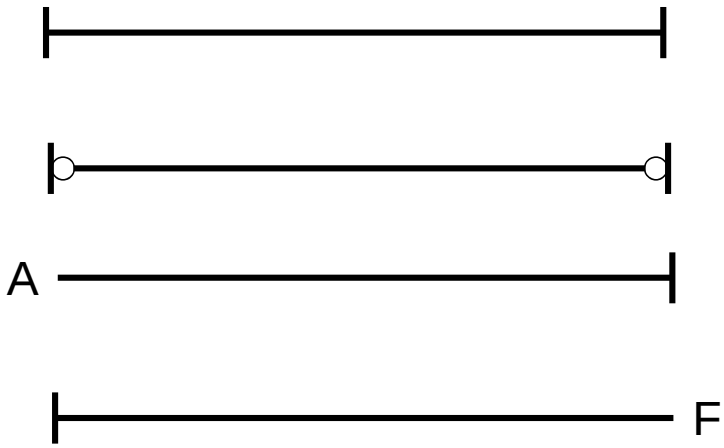
Negación lógica

- Se considera que las entradas y salidas de circuito (salvo especificación en contrario) son en lógica positiva.

- Negación lógica

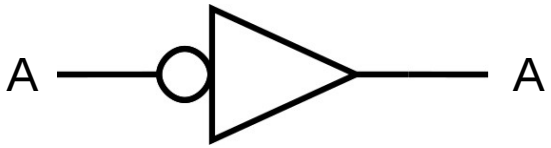


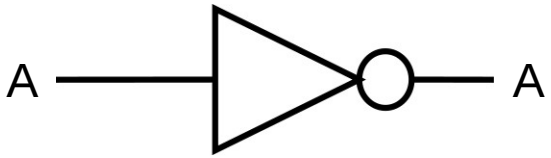
- Identidad lógica



Negación lógica

- ¿Cómo elegir entre ambos inversores? ¿Cuál usar?

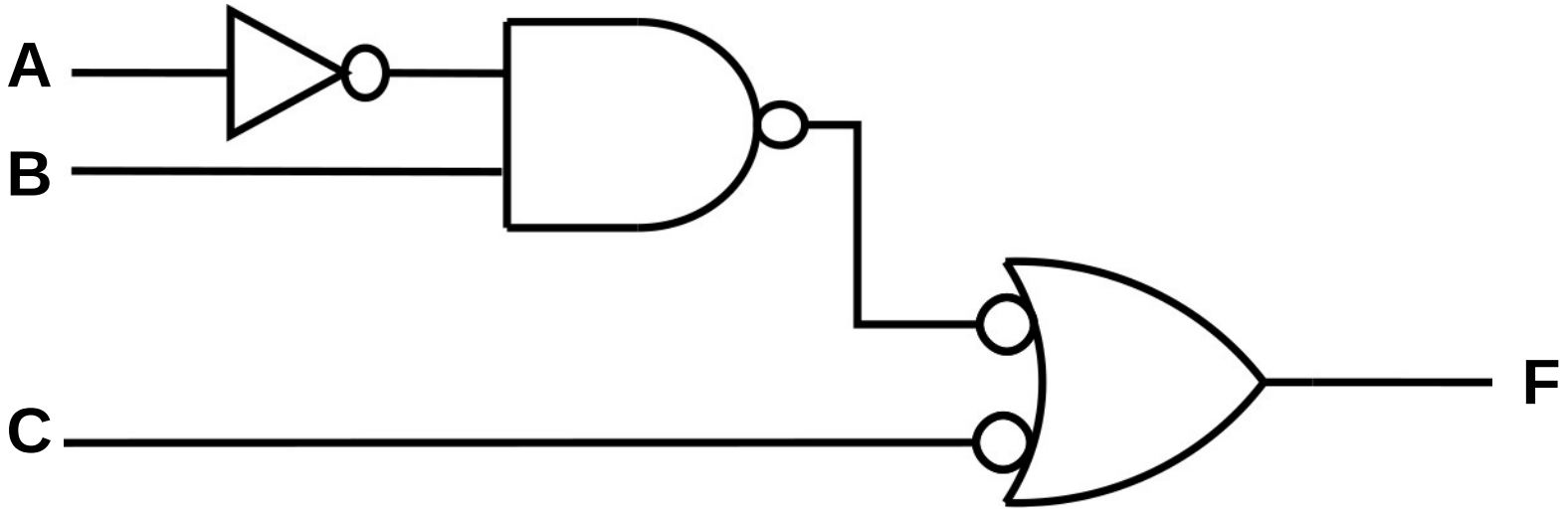




- Cuando el objetivo es compatibilizar señales, usar el que de mayor claridad al circuito.
- Cuando el objetivo es invertir la señal, es indistinto.

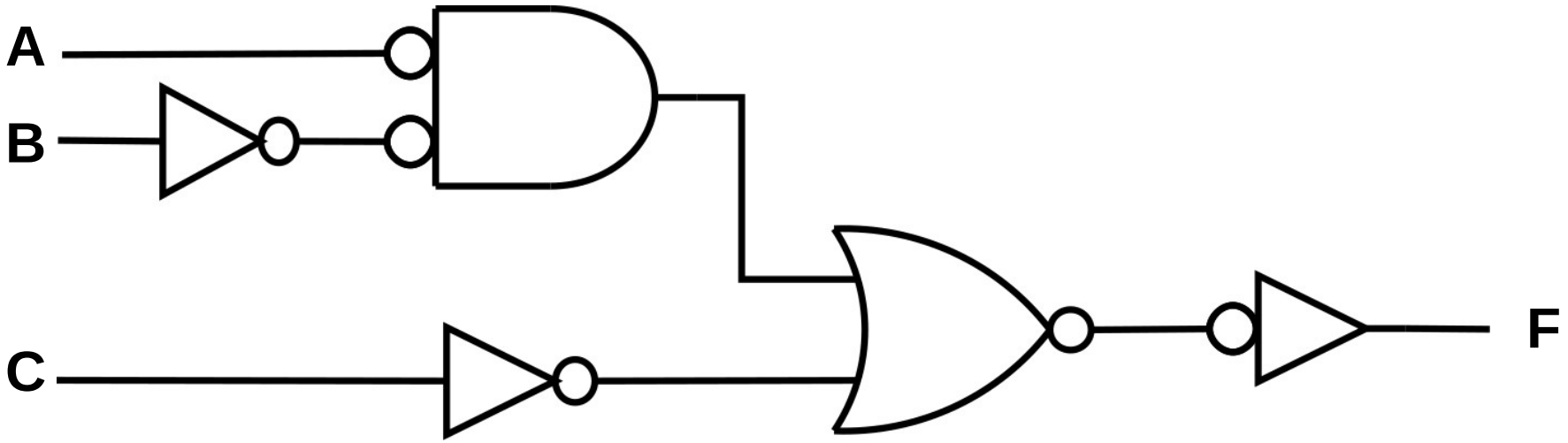
Ejemplo: NAND + INV

- F = A' B + C'
- Implementación con NAND e INV



Ejemplo: NOR + INV

- F = A' B + C'
- Implementación con NOR e INV



Minimización

Criterios de costo

- La complejidad del circuito está directamente relacionada con la expresión que implementa.
- La tabla de verdad (*en ppio*) es única
- Diferentes expresiones algebraica son equivalentes.

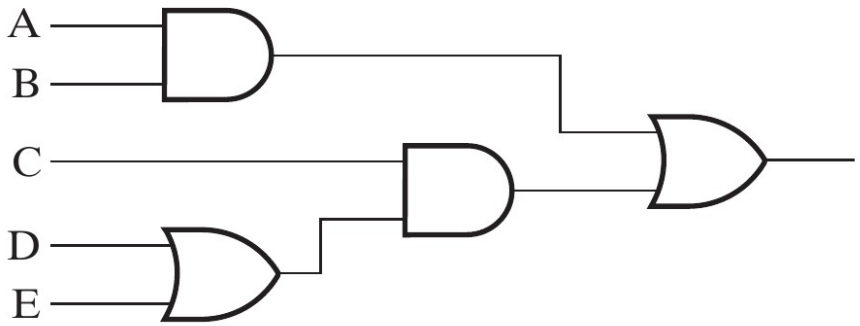
Criterios de evaluación del costo de una implementación

- Literales
- Entradas

Criterios de costo

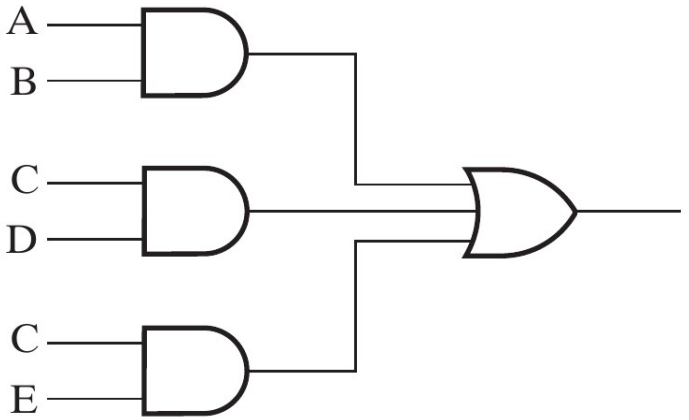
Literales

- Número de literales que aparecen en la expresión y tienen una correspondencia exacta en el diagrama lógico.



(a) $AB + C(D + E)$

5 literales



(b) $AB + CD + CE$

6 literales

Criterios de costo

- El costo en literales es muy simple de evaluar.
- No representa bien la complejidad del circuito en todos lo casos.

$$G = ABCD + \overline{A}\,\overline{B}\,\overline{C}\,\overline{D}$$ and $$G = (\overline{A} + B)(\overline{B} + C)(\overline{C} + D)(\overline{D} + A)$$

8 literales
2 términos

8 literales
4 términos

Criterios de costo

Entradas

- Número de entradas de las compuertas de la implementación que se corresponda exactamente con la expresión lógica.
- A partir del diagrama lógico: contar el total de entradas a las compuertas
- A partir de la función:

  Todas las apariciones de literales

  + El número de términos con 2 o más literales

  + Número de literales distintos complementados

Criterios de costo

$$G = ABCD + \overline{A}\,\overline{B}\,\overline{C}\,\overline{D}$$ and $$G = (\overline{A} + B)(\overline{B} + C)(\overline{C} + D)(\overline{D} + A)$$

8 literales
2 términos
4 literales complementados

8+2+4 = 14

Costo en literales: 8
Costo en entradas: 14

8 literales
4 términos
4 literales complementados

8+4+4 = 16

Costo en literales: 8
Costo en entradas: 16

Minimización

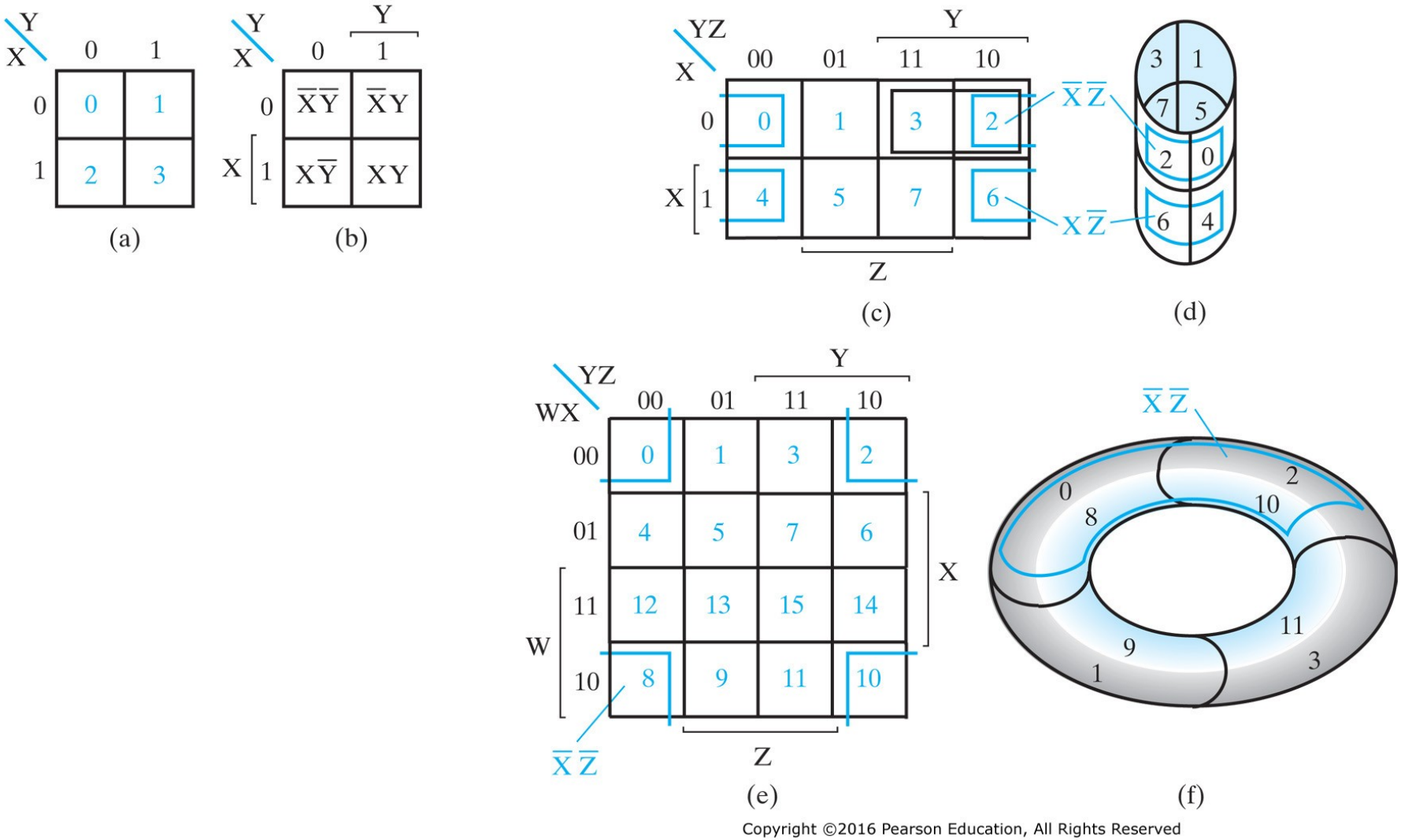
- La *minimización a nivel de compuertas* es la tarea de encontrar **una** implementación óptima de la función booleana que describe el circuito digital.

- Método Gráfico
  - Diagrama de Karnaugh
  - Diagrama de Veitch
- Método Tabular

Método gráfico

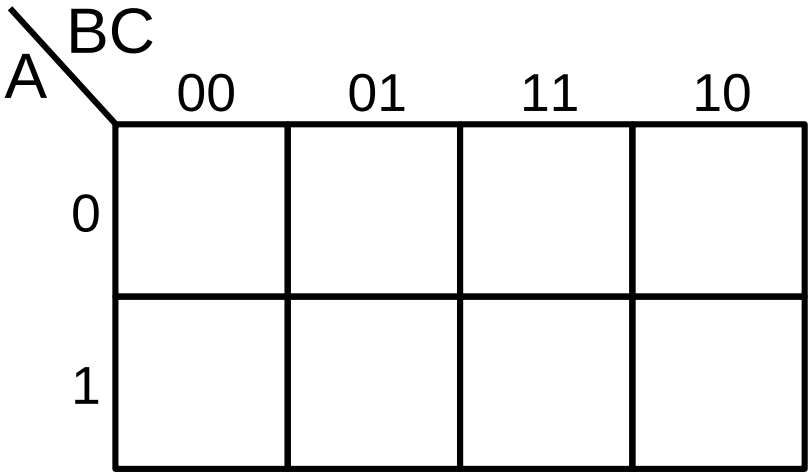
- Consideramos mapas de 2, 3 y 4 variables
- El número de cuadrados en el mapa es igual a la cantidad de posibles minitérminos con esa cantidad de variables.
- Casilleros *adyacentes* difieren en exactamente una variable. (*código Gray*)
- Dos términos productos son *adyacentes* si difieren en exactamente un literal, que aparece complementado en uno y sin complementar en el otro.

Método gráfico

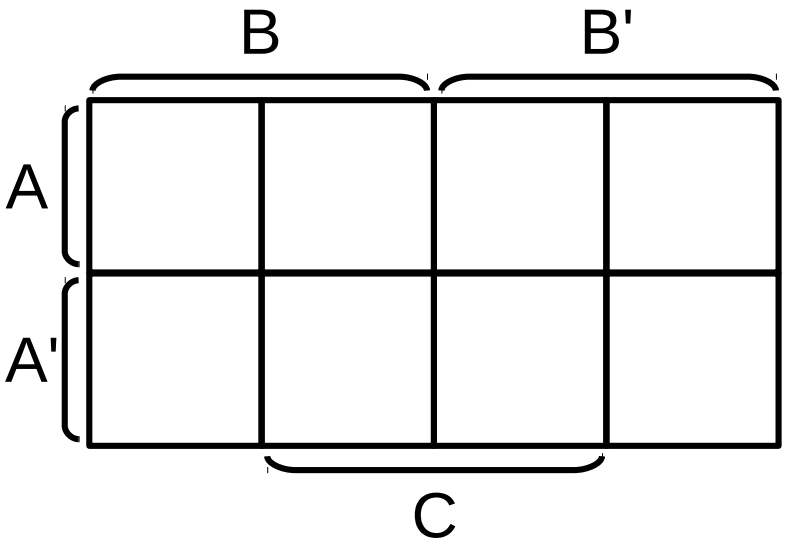


Método gráfico

Karnaugh



Veitch



Método tabular

- El método gráfico se puede extender para 5 y 6 variables usando múltiples mapas de 4 variables.

- ¿Y para más variables?

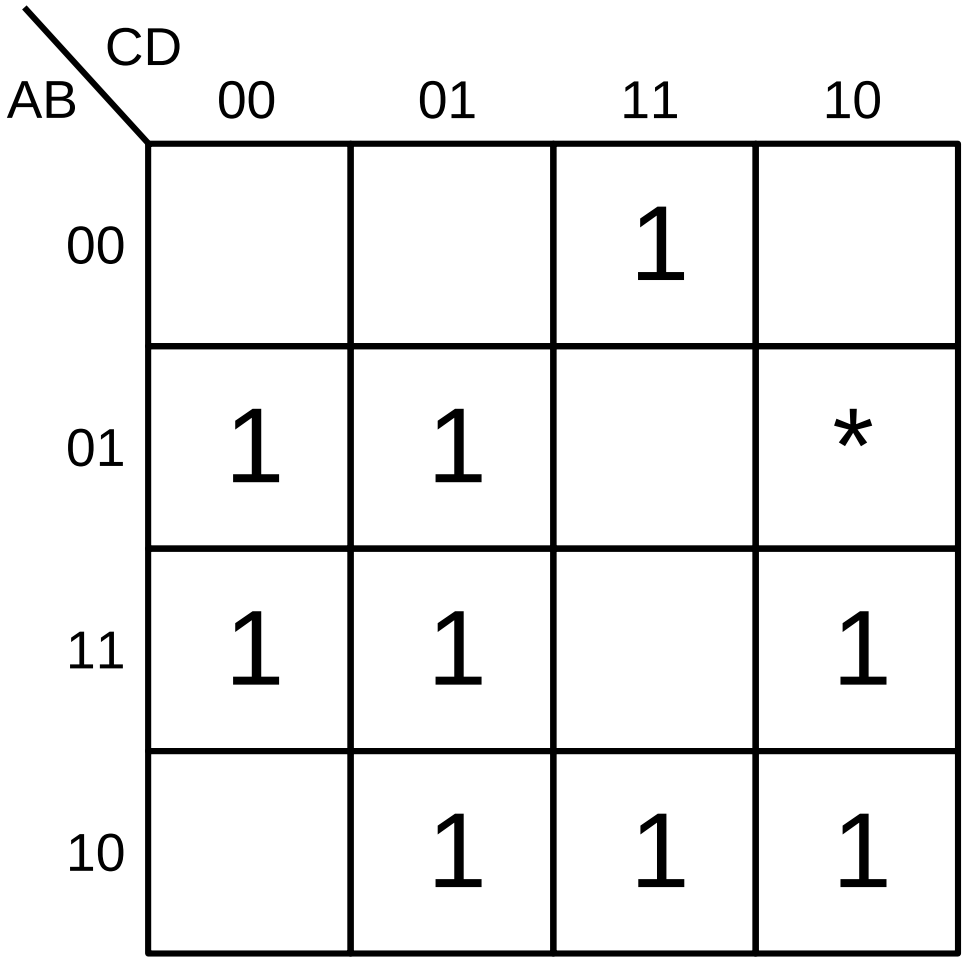
  El método tabular

Método tabular

1) Expandir la función
2) Separar en grupos las combinaciones que validan la función (incluyendo opcionales) según la cantidad de 1s: ninguno, uno, dos …
3) Ordenar los grupos según la cantidad de unos.
4) Comparar los términos en grupos consecutivos y simplificar teniendo en cuenta que difieran en una única variable. Marcar los términos simplificados.
5) Repetir paso 4 hasta que no se pueda más.
6) Los términos que no son cubiertos por otro término son los implicantes primos.

Método tabular

1) Expandir la función



Minitérminos

0011

0100

0101

0110

1001

1010

1011

1100

1101

1110

Método tabular

2) Minitérminos

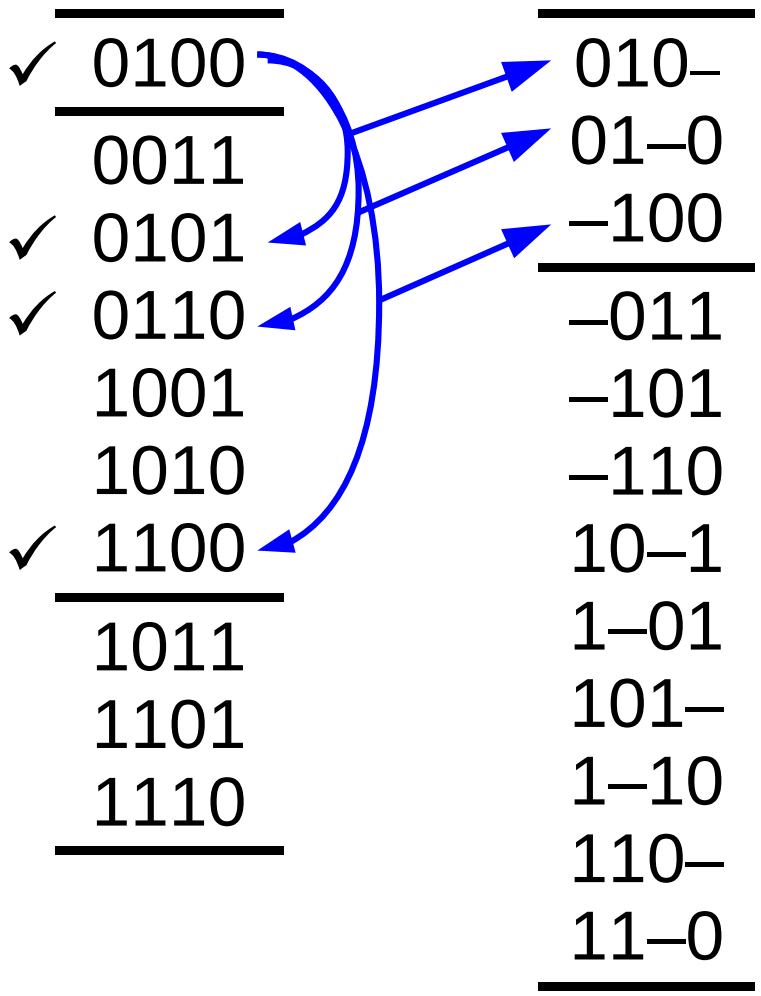
0011
0100
0101
0110
1001
1010
1011
1100
1101
1110

3) Minitérminos en orden

0100
0011
0101
0110
1001
1010
1100
1011
1101
1110

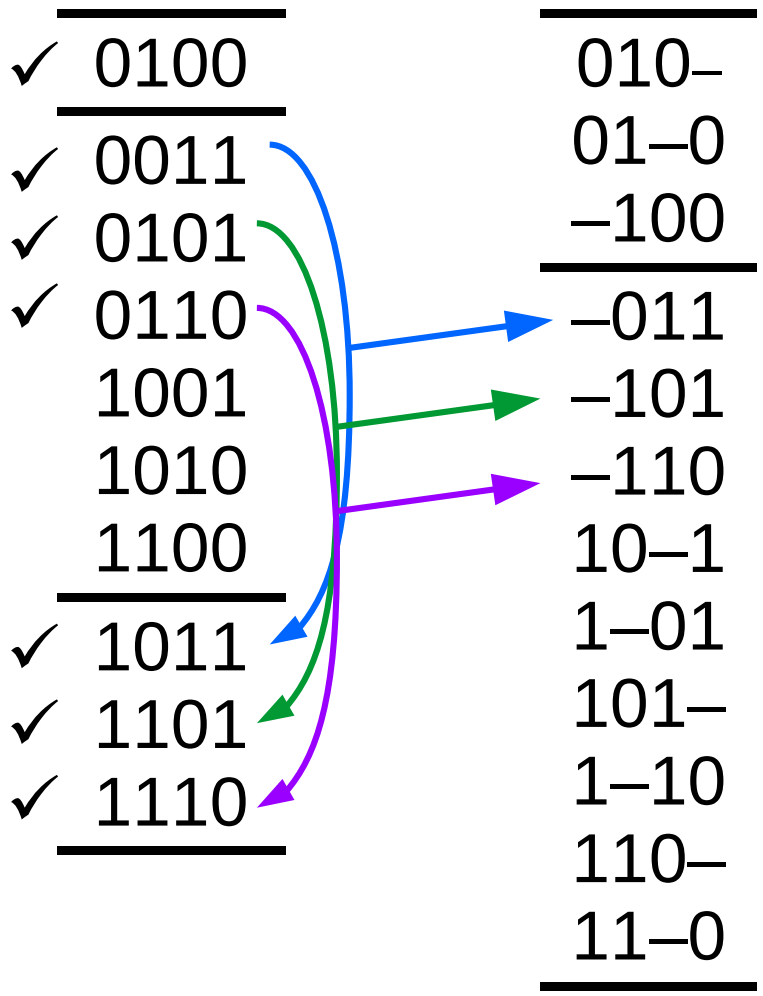
Método tabular

4)



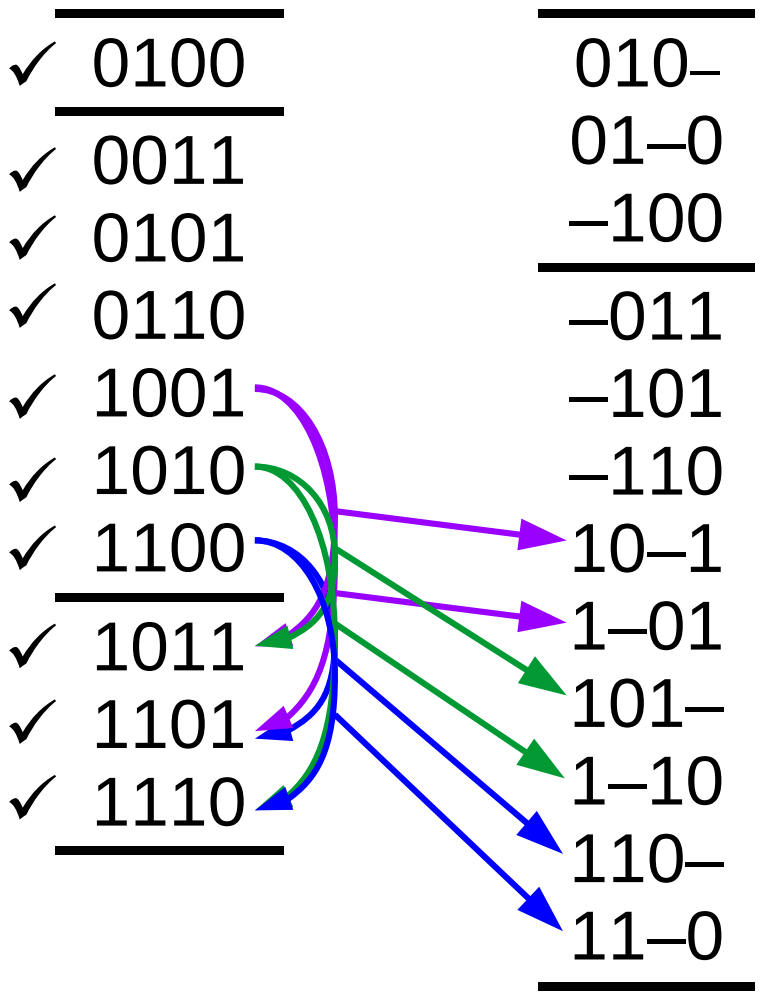
Método tabular

4)



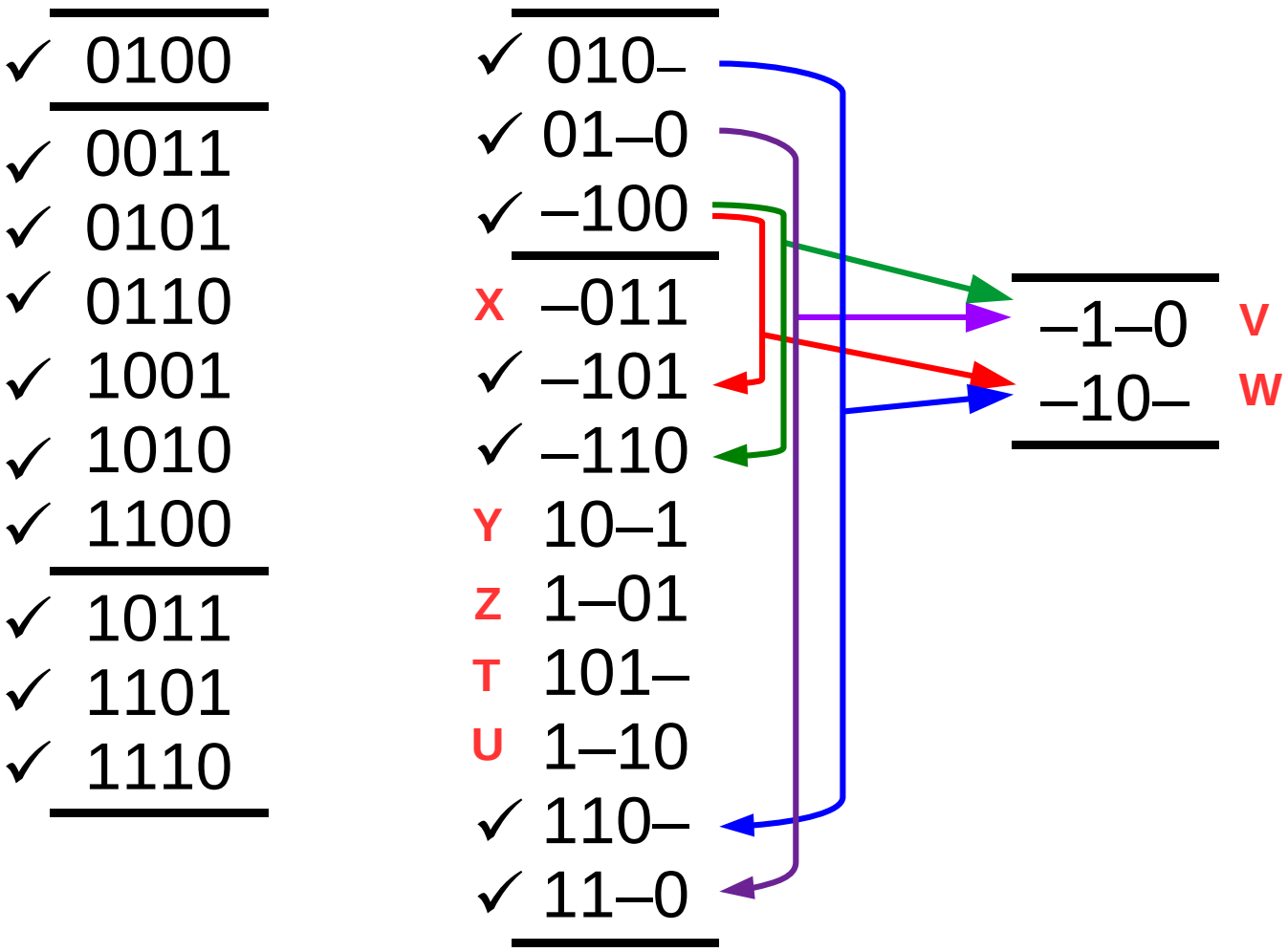
Método tabular

4)



Método tabular

4)



Método tabular

Tabla de implicantes primos

	ABCD	0011	0100	0101	1001	1010	1011	1100	1101	1110
X	–011									
Y	10–1									
Z	1–01									
T	101–									
U	1–10									
V	–1–0									
W	–10–									

Minitérminos 1
No opcionales

Método tabular

Tabla de implicantes primos

	ABCD	**0011**	**0100**	**0101**	**1001**	**1010**	**1011**	**1100**	**1101**	**1110**
X	–011	X					X			
Y	10–1				X		X			
Z	1–01				X				X	
T	101–					X	X			
U	1–10					X				X
V	–1–0		X					X		X
W	–10–		X	X				X	X	

Marcamos con una X qué implicante cubre qué minitérmino.

Método tabular

Tabla de implicantes primos

	ABCD	**0011**	**0100**	**0101**	**1001**	**1010**	**1011**	**1100**	**1101**	**1110**
X	–011	X					X			
Y	10–1				X		X			
Z	1–01				X				X	
T	101–					X	X			
U	1–10					X				X
V	–1–0		X					X		X
W	–10–		X	**X**				X	X	
		✓	✓	✓			✓	✓	✓	

Minitérminos 1
No opcionales

Esenciales

Identificamos los minitérminos cubiertos por un único implicante.
Esos implicantes son esenciales.
Tildamos los minitérminos cubiertos por los implicantes esenciales.

Método tabular

Tabla de implicantes primos

	ABCD	0011	0100	0101	1001	1010	1011	1100	1101	1110	
X	–011	X					X				B'CD
Y	10–1				X		X				AB'D
Z	1–01				X				X		AC'D
T	101–					X	X				AB'C
U	1–10					X				X	ACD'
V	–1–0		X					X		X	BD'
W	–10–		X	X				X	X		BC'
		✓	✓	✓			✓	✓	✓		

La función final será:
H = X . W . (Y+Z) . (T+U) . (U+V)

Método tabular

X,Y,Z,T,U: 3 literales
V,W: 2 literales

$$H = X \cdot W \cdot (Y + Z) \cdot (T + U) \cdot (U + V)$$
$$= XW \cdot (Y + Z) \cdot (TU + TV + U + UV)$$
$$= XW \cdot (Y + Z) \cdot (U + TV)$$
$$= XW \cdot (\underbrace{YU}_{6} + \underbrace{YTV}_{8} + \underbrace{ZU}_{6} + \underbrace{ZTV}_{8}) \quad \text{literales}$$

Soluciones equivalentes

XWYU: B'CD+BC'+ AB'D + ACD'

XWZU: B'CD+BC'+ AC'D + ACD'

Bibliografía

- *Capítulo 2.* Mano, Kime & Martin. *Logic and computer design fundamentals.* Prentice Hall (5ta Ed, 2015)
- *Capítulo 1.* Mano & Ciletti. *Digital Design* (5ta Ed, 2013)
- *Capítulo suplementario "More Optimization".* Mano, Kime & Martin. *Logic and computer design fundamentals.*
  http://wps.pearsoned.com/ecs_mano_lcdf_5/248/63706/16308896.cw/index.html

Circuitos integrados

Circuitos integrados

- Circuito intergrado (IC) o *chip*:

  Estructura de silicio (material semiconductor) que contiene los componentes electrónicos de compuertas digitales y elementos de almacenamiento.

  Se monta en un contenedor de plástico o cerámica.

Niveles de integración

A medida que mejor la tecnología, se incrementa la cantidad de compuertas que puede haber en un chip.

Small-scale integration (SSI)

- *V*arias compuertas independientes en un chip. Menos de 10 compuertas.

Medium-scale integration (MSI)

- Realizan alguna función específica elemental (sumar). Entre 10 y 100 compuertas.

Large-scale integration (LSI)

- Incluye procesadores pequeños, memorias pequeñas y módulos programables. Entre 100 y 100.000 compuertas.

Very-large-scale integration (VLSI)

- Microprocesadores complejos o para procesamiento digital de señales (DSP). Más de 100.000 compuertas (cientos de millones).

SSI: Bloques funcionales

- Decoder
- Encoder
- Multiplexor

Decoder

- Un código de *n* bits puede representar 2^n elementos distintos.
- *Decodificador de n a m*

  Convierte una entrada *n* bits en una salida *m* bits: $n \leq m \leq 2^n$

  - Genera 2^n (o menos) minitérminos a partir de los n bits de entrada.
  - La salida D_i es igual a 1 ($D_j = 0\ \forall j \neq i$) si la entrada codifica el valor *i.*

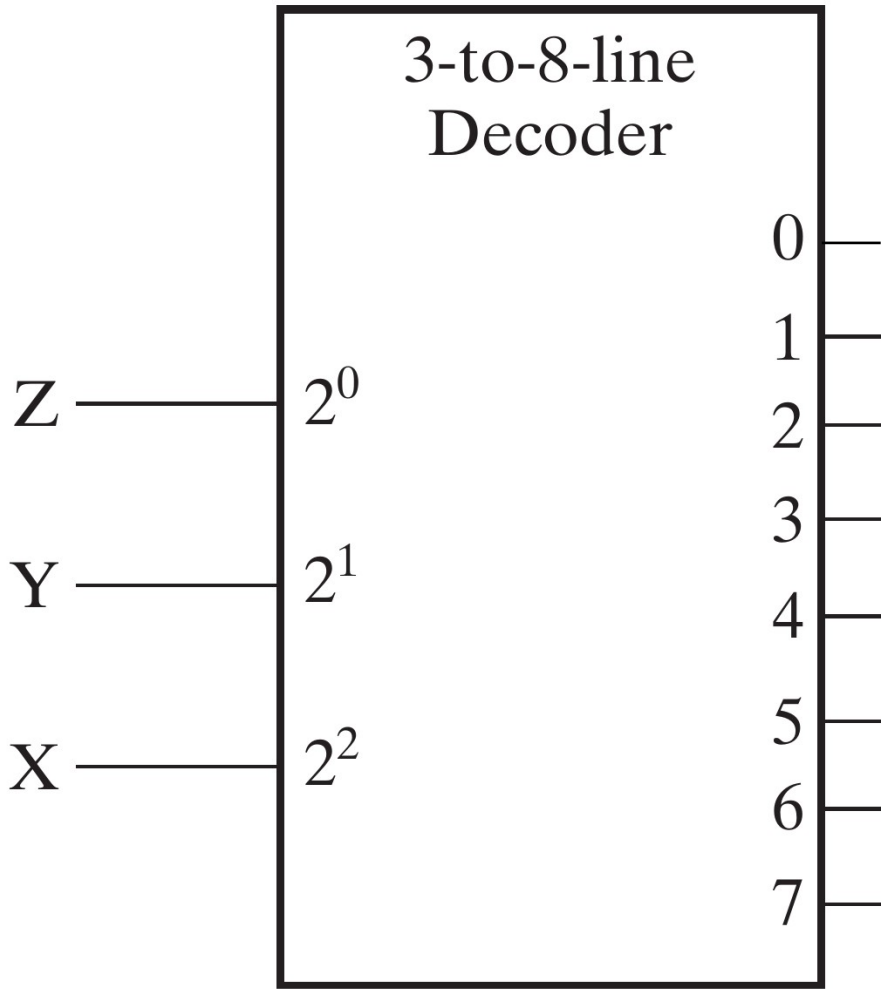
Decoder

Decoder 1 a 2

A	D_0	D_1
0	1	0
1	0	1

Decoder 2 a 4

A_1	A_0	D_0	D_1	D_2	D_3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1



Encoder

- Realiza la función inversa del decoder.
- Tiene 2^n entradas (o menos) y *n* salidas.

Encoder 8 a 3

Inputs								Outputs		
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	A_2	A_1	A_0
0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	1	0	0	0	1
0	0	0	0	0	1	0	0	0	1	0
0	0	0	0	1	0	0	0	0	1	1
0	0	0	1	0	0	0	0	1	0	0
0	0	1	0	0	0	0	0	1	0	1
0	1	0	0	0	0	0	0	1	1	0
1	0	0	0	0	0	0	0	1	1	1

Encoder

- Ambigüedades:
  - Solo una entrada puede tener valor 1.
    ¿Qué pasa cuando hay más de una entrada en 1? La salida dependerá de la implementación.
  - ¿Qué pasa cuando todas las entradas valen 0?

- Se establecen prioridades.

Priority encoder

- Las entradas tienen diferentes prioridades.
- Agrega una salida que indica validez de la demás salidas.

Inputs				Outputs		
D_3	D_2	D_1	D_0	A_1	A_0	V
0	0	0	0	X	X	0
0	0	0	1	0	0	1
0	0	1	X	0	1	1
0	1	X	X	1	0	1
1	X	X	X	1	1	1

Multiplexor

- Circuito combinacional que selecciona una de las entradas la mapea a la (única) salida.
- Generalmente, tiene 2^n entradas y *n* líneas de selección que determinan cuál de las entradas será la salida.

S	I_0	I_1	Y
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

Tabla condensada

S	Y
0	I_0
1	I_1

Multiplexor

S_1	S_0	Y
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3