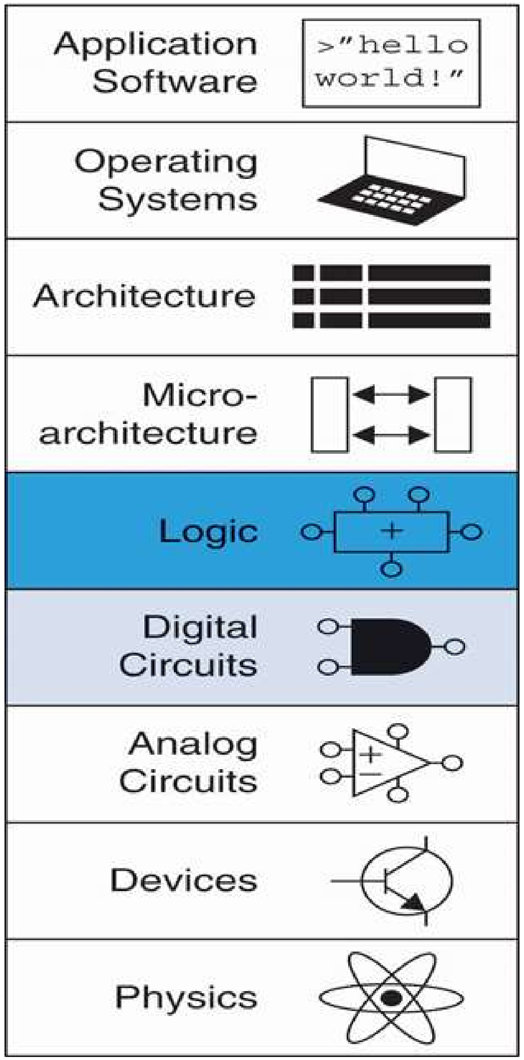
Inicio AOC_02.pptx

Based on the text Digital design and computer architecture,

harris & harris 2nd Edition 2012

Temas

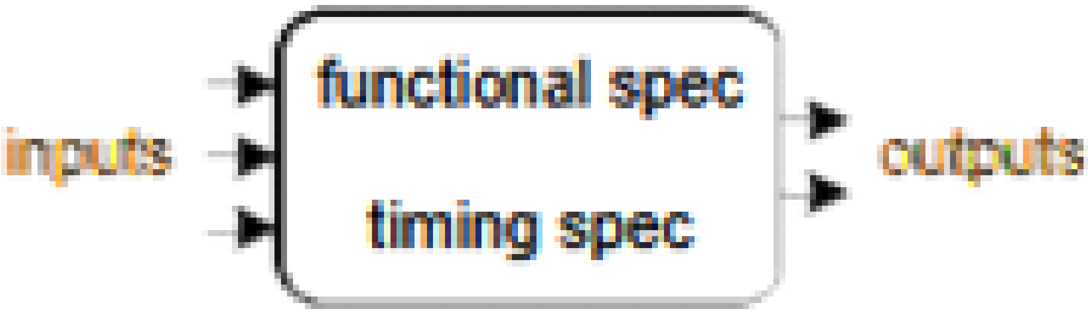
- Introducción
- Ecc Booleaneas
- Algebra Booleana
- Desde la lógica a las compuertas (Gates)
- Logica Combinacional Multinivel
- X's   y   Z's
- Mapas de Karnaugh
- Blocks constructivos Combinacionales
- Sincronización



Introduccción

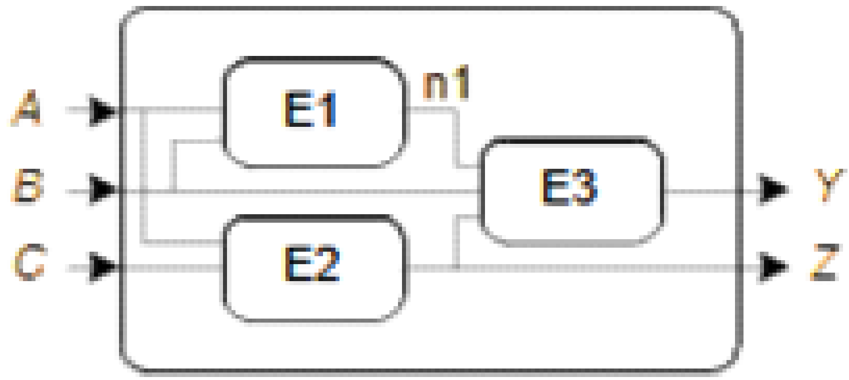
Un circuito lógico está compuesto de:

- Entradas (Inputs)
- Salidas (Outputs)
- Especificación funcional
- Especificación de temporización o sincronización



Circuitos

- Nodos
  - Entradas: *A*, *B*, *C*
  - Salidas: *Y*, *Z*
  - Interno: n1
- Elementos de Circuito
  - E1, E2, E3
  - Cada uno un circuito

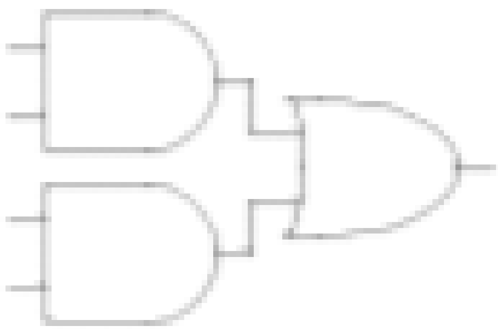


Tipos de circuitos lógicos

- Logica Combinacional
  - Sin Memoria
  - Las salidas están determinadas por los valores actuales de las entradas
- Logica Secuencial
  - Tiene memoria
  - Las salidas están determinadas por los valores previos y actuales de las entradas.

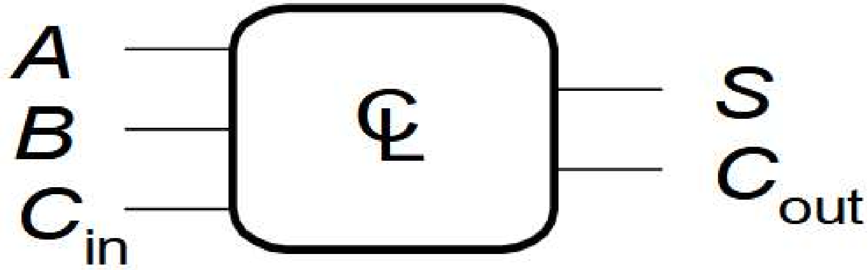
Reglas de creación combinatoria

- Todo elemento es combinacional
- Todo nodo es ya sea una entrada o se conecta *exactamente a una* salida
- El circuito no contiene trayectos ciclicos
- Ejemplo:



Ecuaciones booleanas

- Especificación Funcional de salidas en términos de las entradas
- Ejemplo: $S = \mathrm{F}(A, B, C_{in})$ $C_{out} = \mathrm{F}(A, B, C_{in})$



$$S = A \oplus B \oplus C_{in}$$
$$C_{out} = AB + AC_{in} + BC_{in}$$

Algunas definiciones

- Complemento: variable con una barra sobre él

  $\overline{A}$, $\overline{B}$, $\overline{C}$

- Otro modo de indicar una variable complementada es con ' o con ¬
  A', B', C'   ¬A, ¬B, ¬C
- Literal: variable o su complemento
  A, A', B, B', C', C
  Implicante: producto de literales
  ABC, AC, BC
- Minterm: product that includes all input variables
  ABC, ABC, ABC
- Maxterm: sum that includes all input variables
  (A+B+C), (A+B+C), (A+B+C)

Formato Suma de productos (SOP)

- Todas las ecuaciones pueden ser escritas en la forma SOP form
- Cada fila tiene un **mintérmino**
- Un mintérm is un producto (AND) de literales
- Cada mintérmino es TRUE para esa fila (y solo esa fila)
- Escriba la función haciendo OR con los mintérminos donde la salida es verdadera (TRUE)
- Así se tiene una suma (OR) de productos (términos AND)

A	B	Y	minterm	minterm name
0	0	0	$\overline{A}\ \overline{B}$	m_0
0	1	1	$\overline{A}\ B$	m_1
1	0	0	$A\ \overline{B}$	m_2
1	1	1	$A\ B$	m_3

$$Y = \mathrm{F}(A, B) = \mathrm{A'B+AB}$$

Formato Suma de productos (SOP)

- Todas las ecuaciones se pueden escribir en forma SOP.
- Cada fila tiene un minitérmino.
- Un mintérmino es un producto (AND) de literales.
- Cada mintérmino es TRUE para esa fila (y solo esa fila).
- Forme la función haciendo ORing en mintérminos donde la salida es TRUE.
- Así, se tiene una suma (OR) de productos (Y términos).

A	**B**	**Y**	**minterm**	**minterm name**
0	0	0	$\overline{A}\ \overline{B}$	m_0
0	1	1	$\overline{A}\ B$	m_1
1	0	0	$A\ \overline{B}$	m_2
1	1	1	$A\ B$	m_3

$$Y = F(A, B) = A'B + AB = \Sigma(1, 3)$$

Formato Productos de Suma (POS)

- Todas las ecuaciones booleanas se pueden escribir en forma POS.
- Cada fila tiene un maxtérmino.
- Un maxtérmino es una suma (OR) de literales.
- Cada maxtérmino es FALSE para esa fila (y solo esa fila).
- Escriba la función haciendo ANDing con los maxtérminos para los cuales la salida es FALSE.
- Así, se tiene un producto (AND) de sumas (términos OR).

A	*B*	*Y*	**maxterm**	**maxterm name**
0	0	0	$A + B$	M_0
0	1	1	$A + \overline{B}$	M_1
1	0	0	$\overline{A} + B$	M_2
1	1	1	$\overline{A} + \overline{B}$	M_3

$$Y = \mathrm{F}(A, B) = (A + B)(A + B') = \Pi(0, 2)$$

Ejemplo de ecuaciones booleanas

- Estás yendo a la cafetería a almorzar:
  - No vas a almorzar (E’)
  - si no está abierto (O’) o
  - si solo sirven hotdogs (C).
- Escriba una tabla de verdad para determinar si almorzarás (E).

O	*C*	*E*
0	0	
0	1	
1	0	
1	1	

Ejemplo de ecuaciones booleanas

- vas a la cafetería a almorzar:
  - No vas a almorzar (E’)
  - Si no está abierto (O’)
  - o Si solo sirven hotdogs(C)
- Escriba una tabla de verdad para determinar si almorzarás (E).

O	C	E
0	0	0
0	1	0
1	0	1
1	1	0

Formas SOP y POS

- SOP – Suma de Productos

O	C	E	minterm
0	0		$\overline{O}\ \overline{C}$
0	1		$\overline{O}\ C$
1	0		$O\ \overline{C}$
1	1		$O\ C$

- POS – Productos de Sumas

O	C	E	maxterm
0	0		$O + C$
0	1		$O + \overline{C}$
1	0		$\overline{O} + C$
1	1		$\overline{O} + \overline{C}$

Formas SOP y POS

- SOP – Suma de Productos

O	C	E	minterm
0	0	0	$\overline{O}\ \overline{C}$
0	1	0	$\overline{O}\ C$
1	0	1	$O\ \overline{C}$
1	1	0	$O\ C$

$$E = OC' = \Sigma(2)$$

- POS – Producto de Sumas

O	C	E	maxterm
0	0	0	$O + C$
0	1	0	$O + \overline{C}$
1	0	1	$\overline{O} + C$
1	1	0	$\overline{O} + \overline{C}$

$$E = (O + C)(O + C')(O' + C') = \Pi(0, 1, 3)$$

Álgebra de boole

- Axiomas y teoremas para simplificar ecuaciones booleanas.
- Como el álgebra regular, pero más simple: las variables tienen solo dos valores (1 o 0).
- Dualidad en axiomas y teoremas: ANDs y ORs, 0 y 1 intercambiados.

Axiomas booleanos

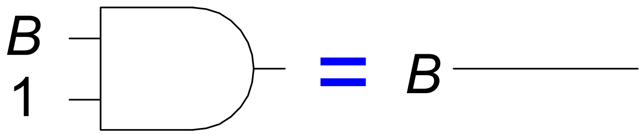
	Axiom		Dual	Name
A1	$B = 0$ if $B \neq 1$	A1′	$B = 1$ if $B \neq 0$	Binary field
A2	$\overline{0} = 1$	A2′	$\overline{1} = 0$	NOT
A3	$0 \bullet 0 = 0$	A3′	$1 + 1 = 1$	AND/OR
A4	$1 \bullet 1 = 1$	A4′	$0 + 0 = 0$	AND/OR
A5	$0 \bullet 1 = 1 \bullet 0 = 0$	A5′	$1 + 0 = 0 + 1 = 1$	AND/OR

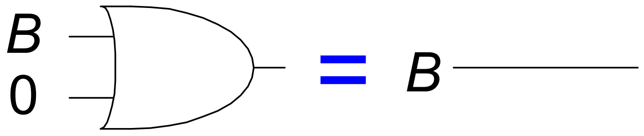
	Theorem		Dual	Name
T1	$B \bullet 1 = B$	T1′	$B + 0 = B$	Identity
T2	$B \bullet 0 = 0$	T2′	$B + 1 = 1$	Null Element
T3	$B \bullet B = B$	T3′	$B + B = B$	Idempotency
T4		$\overline{\overline{B}} = B$		Involution
T5	$B \bullet \overline{B} = 0$	T5′	$B + \overline{B} = 1$	Complements

T1: Teorema de identidad

$$B \cdot 1 = B$$

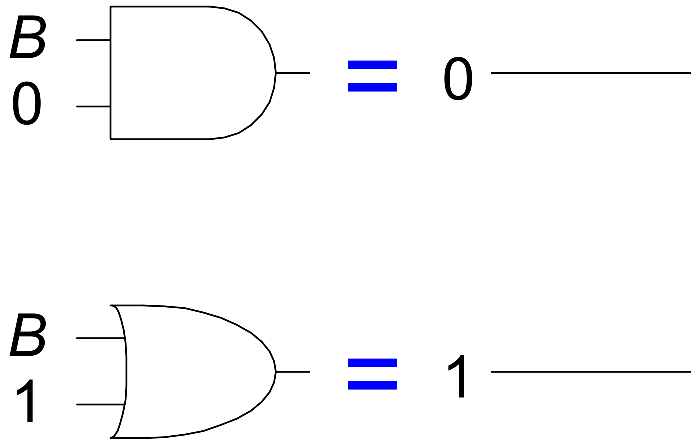
$$B + 0 = B$$





T2: Teorema del elemento nulo

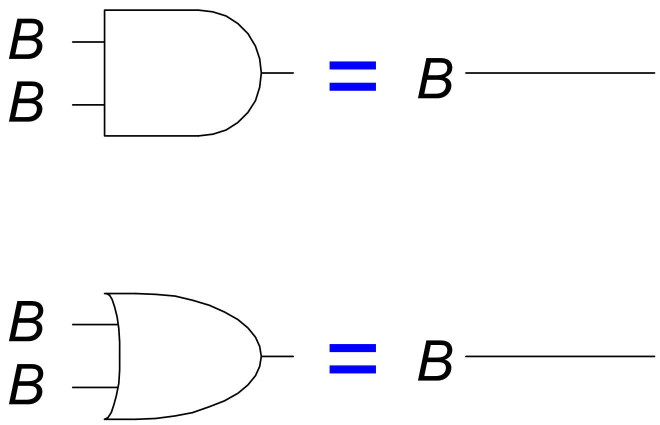
- $B \cdot 0 = 0$
- $B + 1 = 1$



T3: Teorema de la Idempotencia

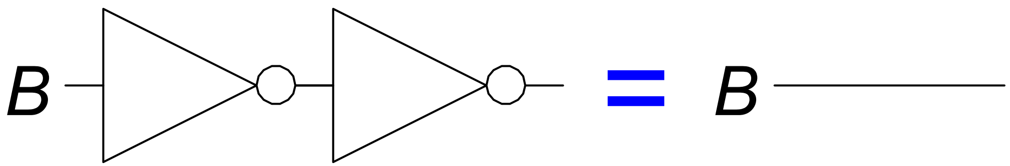
$$B \cdot B = B$$

$$B + B = B$$



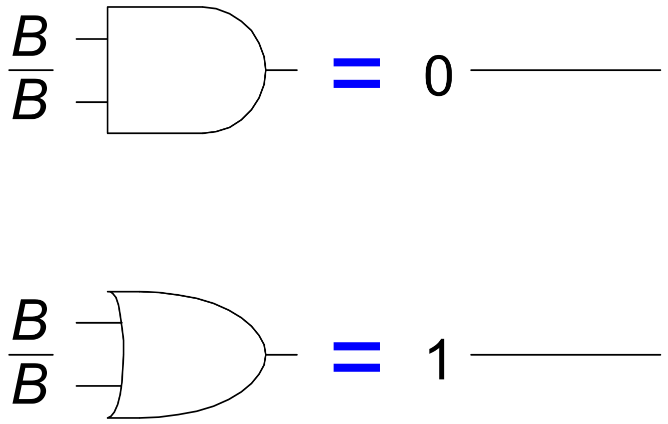
T4: Teorema de la Identidad

$$B = B$$



T5: Teorema del complemento

- $B \cdot B' = 0$
- $B + B' = 1$



Resumen de los teoremas booleanos

	Theorem		Dual	Name
T1	$B \bullet 1 = B$	T1′	$B + 0 = B$	Identity
T2	$B \bullet 0 = 0$	T2′	$B + 1 = 1$	Null Element
T3	$B \bullet B = B$	T3′	$B + B = B$	Idempotency
T4		$\overline{\overline{B}} = B$		Involution
T5	$B \bullet \overline{B} = 0$	T5′	$B + \overline{B} = 1$	Complements

Teoremas booleanos de varias variables

	Theorem		Dual	Name
T6	$B \bullet C = C \bullet B$	T6′	$B + C = C + B$	Commutativity
T7	$(B \bullet C) \bullet D = B \bullet (C \bullet D)$	T7′	$(B + C) + D = B + (C + D)$	Associativity
T8	$(B \bullet C) + B \bullet D = B \bullet (C + D)$	T8′	$(B + C) \bullet (B + D) = B + (C \bullet D)$	Distributivity
T9	$B \bullet (B + C) = B$	T9′	$B + (B \bullet C) = B$	Covering
T10	$(B \bullet C) + (B \bullet \overline{C}) = B$	T10′	$(B + C) \bullet (B + \overline{C}) = B$	Combining
T11	$(B \bullet C) + (\overline{B} \bullet D) + (C \bullet D)$ $= B \bullet C + \overline{B} \bullet D$	T11′	$(B + C) \bullet (\overline{B} + D) \bullet (C + D)$ $= (B + C) \bullet (\overline{B} + D)$	Consensus
T12	$\overline{B_0 \bullet B_1 \bullet B_2 \ldots}$ $= (\overline{B_0} + \overline{B_1} + \overline{B_2} \ldots)$	T12′	$\overline{B_0 + B_1 + B_2 \ldots}$ $= (\overline{B_0} \bullet \overline{B_1} \bullet \overline{B_2})$	De Morgan's Theorem

Simplificando ecuaciones boolenas

Ejemplo 1

$Y = AB + A'B$

$= B(A + A')$ T8

$= B(1)$ T5'

$= B$ T1

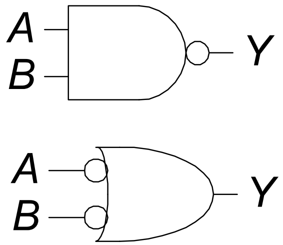
Simplificando ecuaciones boolenas

Ejemplo 2:

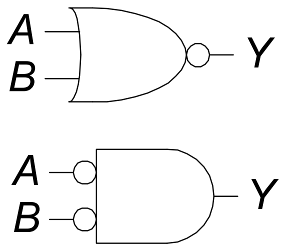
$$
\begin{aligned}
Y &= A(AB + ABC) & \\
&= A(AB(1 + C)) & \text{T8} \\
&= A(AB(1)) & \text{T2'} \\
&= A(AB) & \text{T1} \\
&= (AA)B & \text{T7} \\
&= AB & \text{T3}
\end{aligned}
$$

Teorema de Morgan

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



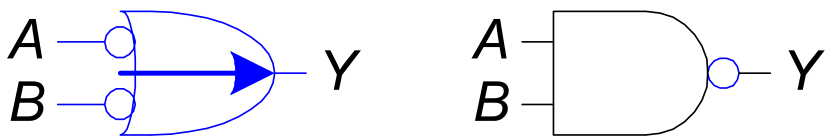
$$Y = (A + B)' = A' \cdot B'$$



Impulsando burbujas

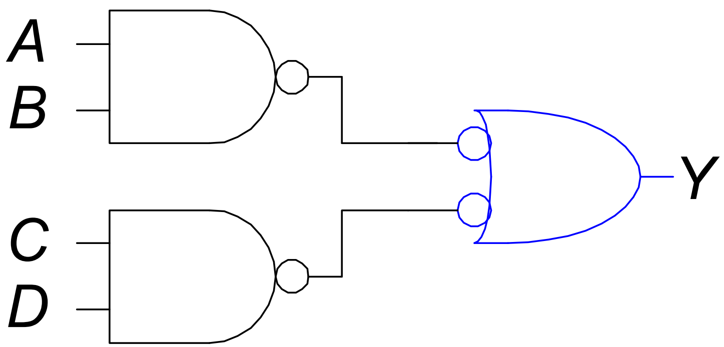
- **Hacia atrás:**
  - Cambia el cuerpo del componente
  - Agrega burbujas a las entradas

- **Hacia Adelante:**
  - Cambia el cuerpo del componente
  - Agrega burbujas a las salidas



Impulsando burbujas

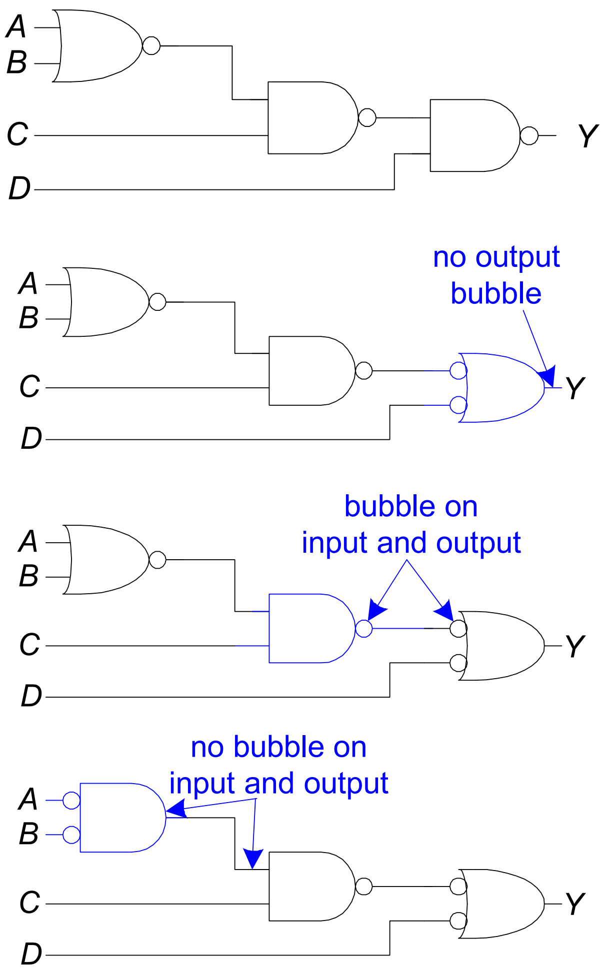
- Cual es la expresión Booleana para este circuito?



$$Y = AB + CD$$

Impulsando burbujas

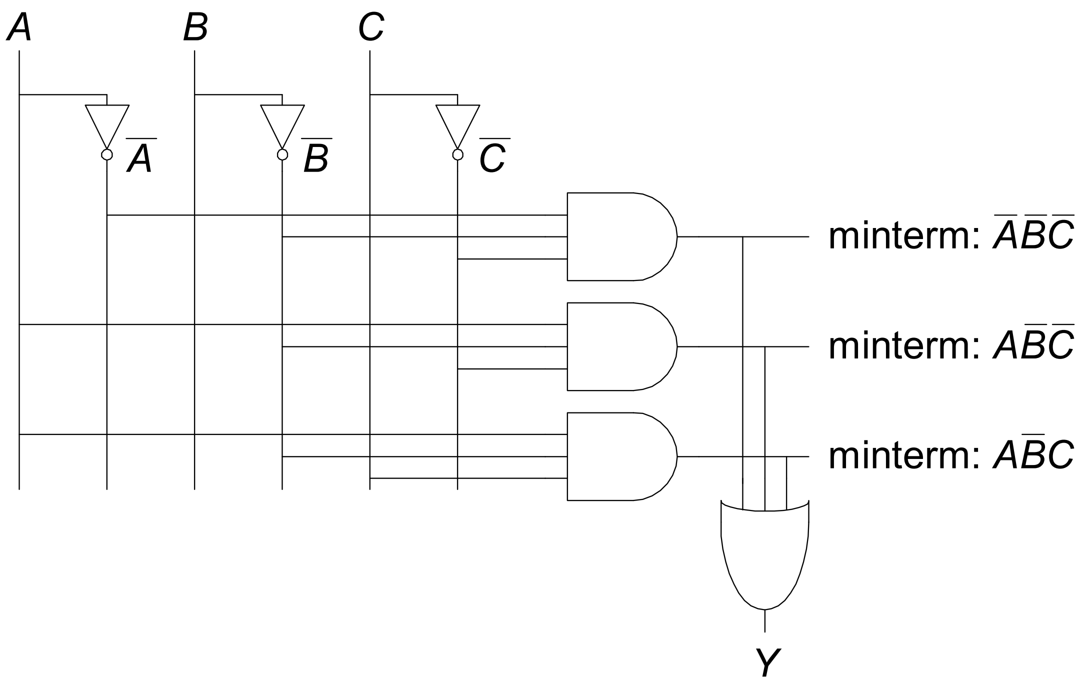
Otro ejemplo:



$$Y = \overline{A}\,\overline{B}C + \overline{D}$$

Desde la lógica a las puertas

- Lógica de dos niveles: ANDs followed by ORs
- Example: $Y = A'B'C' + AB'C' + AB'C$

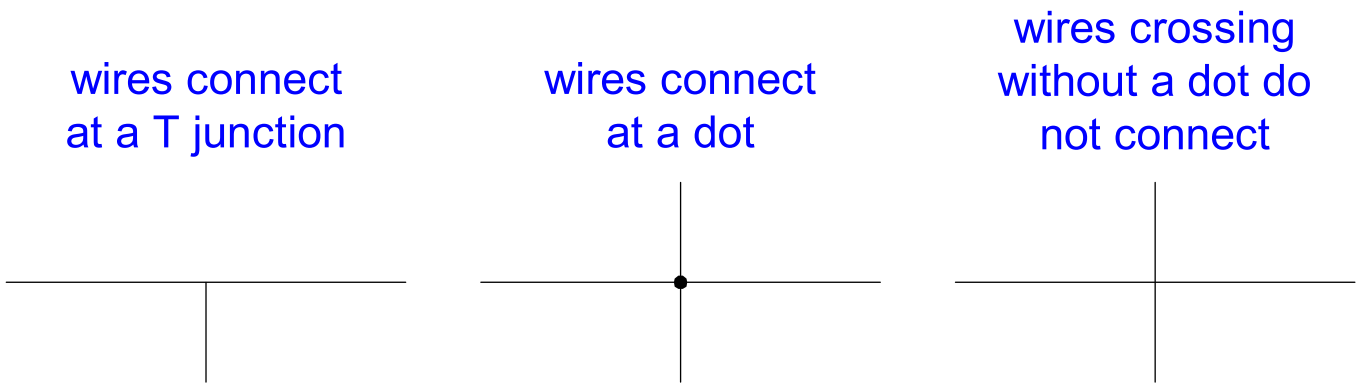


Reglas de los esquemáticos de circuitos

- Entradas a la izquierda (o arriba)
- Salidas a la derecha (o abajo)
- Las puertas fluyen de izquierda a derecha
- Los cables rectos son los mejores

Reglas de los esquemáticos de circuitos

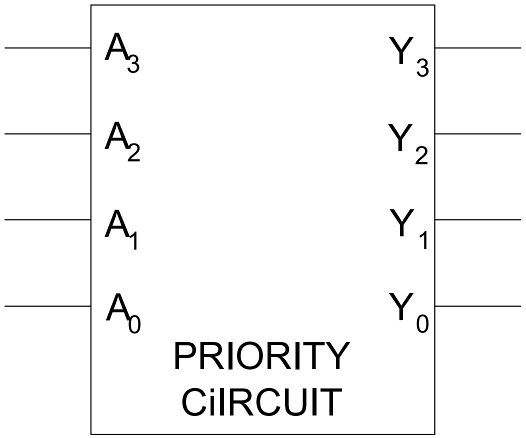
- Los cables siempre se conectan en una unión en T
- Un punto donde se cruzan los cables indica una conexión entre los cables
- Los cables que se cruzan sin un punto no hacen conexión



Circuitos con múltiples salidas

Ejemplo: Circuito Prioritario

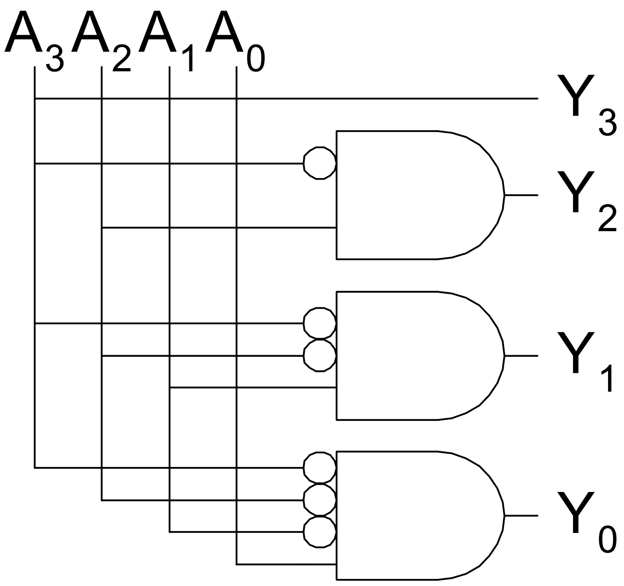
Salida establecida correspondiente a la entrada del bit más significativo de valor TRUE



A_3	A_2	A_1	A_0	Y_3	Y_2	Y_1	Y_0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	0
0	0	1	1	0	0	1	0
0	1	0	0	0	1	0	0
0	1	0	1	0	1	0	0
0	1	1	0	0	1	0	0
0	1	1	1	0	1	0	0
1	0	0	0	1	0	0	0
1	0	0	1	1	0	0	0
1	0	1	0	1	0	0	0
1	0	1	1	1	0	0	0
1	1	0	0	1	0	0	0
1	1	0	1	1	0	0	0
1	1	1	0	1	0	0	0
1	1	1	1	1	0	0	0

Hardware del circuito de prioridad

A_3	A_2	A_1	A_0	Y_3	Y_2	Y_1	Y_0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	0
0	0	1	1	0	0	1	0
0	1	0	0	0	1	0	0
0	1	0	1	0	1	0	0
0	1	1	0	0	1	0	0
0	1	1	1	0	1	0	0
1	0	0	0	1	0	0	0
1	0	0	1	1	0	0	0
1	0	1	0	1	0	0	0
1	0	1	1	1	0	0	0
1	1	0	0	1	0	0	0
1	1	0	1	1	0	0	0
1	1	1	0	1	0	0	0
1	1	1	1	1	0	0	0



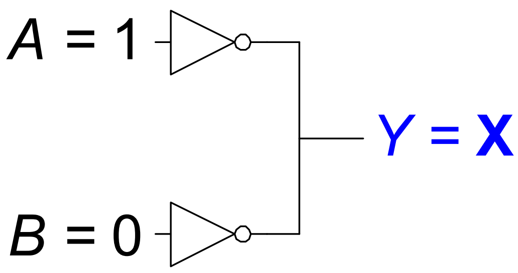
Condiciones DON' T CARES (NO IMPORTA)

A_3	A_2	A_1	A_0	Y_3	Y_2	Y_1	Y_0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	0
0	0	1	1	0	0	1	0
0	1	0	0	0	1	0	0
0	1	0	1	0	1	0	0
0	1	1	0	0	1	0	0
0	1	1	1	0	1	0	0
1	0	0	0	1	0	0	0
1	0	0	1	1	0	0	0
1	0	1	0	1	0	0	0
1	0	1	1	1	0	0	0
1	1	0	0	1	0	0	0
1	1	0	1	1	0	0	0
1	1	1	0	1	0	0	0
1	1	1	1	1	0	0	0

A_3	A_2	A_1	A_0	Y_3	Y_2	Y_1	Y_0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	X	0	0	1	0
0	1	X	X	0	1	0	0
1	X	X	X	1	0	0	0

Contención: X

- Contención: el circuito intenta llevar la salida a 1 y 0
- Valor real en algún punto intermedio
- Podría ser 0, 1 o en zona prohibida
- Puede cambiar con el voltaje, la temperatura, el tiempo, el ruido
- A menudo causa una disipación de energía excesiva



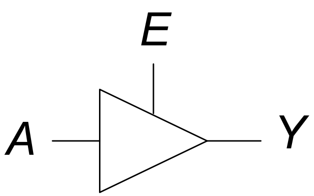
Advertencias: La contención generalmente indica un error.
X se usa para "no importa" y contención: se debe mirar el contexto para distinguirlos

Estado de nodo flotante

- Flotante, alta impedancia, abierto, alto Z
- La salida flotante puede ser 0, 1 o algo intermedio

-- Un voltímetro no indicará si un nodo está flotando, pero si puede observar con osciloscopio

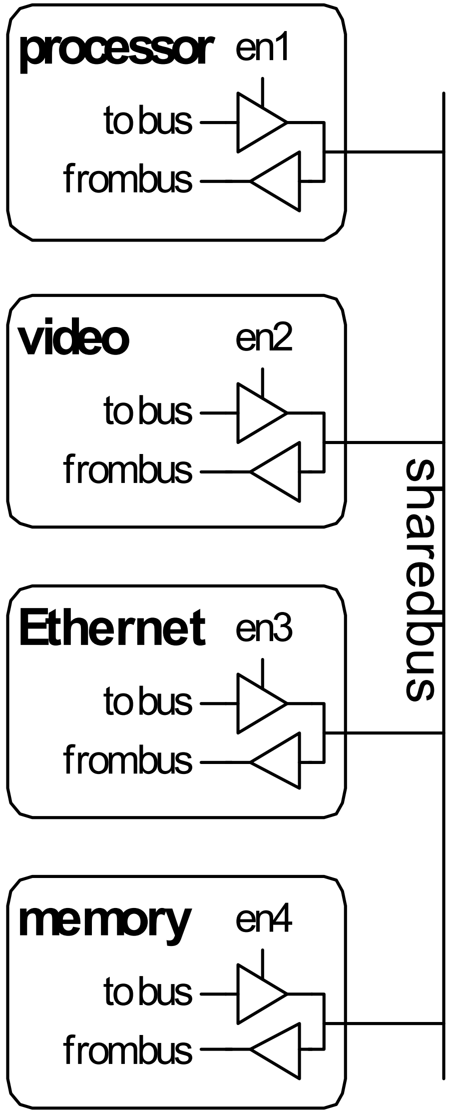
Buffer (separador) Triestado



E	A	Y
0	0	Z
0	1	Z
1	0	0
1	1	1

Aplicación del Triestado en la conexión a buses de un sistema digital

- Los nodos flotantes se utilizan en buses triestado:
  - -- Muchos conductores diferentes
  - -- Exactamente uno está activo a la vez



Mapas de Karnaugh (K-Maps)

- Las expresiones booleanas se pueden minimizar combinando términos
- Los mapas K minimizan las ecuaciones gráficamente

$$PA + PA' = P$$

A	B	C	Y
0	0	0	**1**
0	0	1	**1**
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

Y — C \ AB	00	01	11	10
0	**1**	0	0	0
1	**1**	0	0	0

Y — C \ AB	00	01	11	10
0	$\mathbf{\bar{A}\bar{B}\bar{C}}$	$\bar{A}B\bar{C}$	$AB\bar{C}$	$A\bar{B}\bar{C}$
1	$\mathbf{\bar{A}\bar{B}C}$	$\bar{A}BC$	ABC	$A\bar{B}C$

K-Maps

- Círculo 1 en cuadrados adyacentes
- En la expresión booleana, incluya solo literales cuyo valor TRUE y su complemento no estén en el círculo

A	*B*	*C*	*Y*
0	0	0	**1**
0	0	1	**1**
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

Y — *C* \ *AB*	00	01	11	10
0	1	0	0	0
1	1	0	0	0

$$Y = A'B'$$

K-Maps de tres entradas

Y: C \ AB	00	01	11	10
0	$\bar{A}\bar{B}\bar{C}$	$\bar{A}B\bar{C}$	$AB\bar{C}$	$A\bar{B}\bar{C}$
1	$\bar{A}\bar{B}C$	$\bar{A}BC$	ABC	$A\bar{B}C$

Truth Table

A	B	C	Y
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

K-Map

Y: C \ AB	00	01	11	10
0	0	1	1	0
1	0	1	0	0

Definiciones en K-Maps

- **Complemento:** variable marcada con alguno de los caracteres indicados anteriormente

  A’, B’, C’

- **Literal:** variable o su complemento

  A, A’, B, B’, C, C’

- **Implicante:** producto de literales

  AB’C, A’C, BC

- **Implicant Primo :** implicante correspondiente al círculo más grande en el K-Map

Reglas en K-Map

- Cada 1 debe rodearse (con círculo) al menos una vez
- Cada círculo debe abarcar una potencia de 2 (es decir, 1, 2, 4) cuadrados en cada dirección
- Cada círculo debe ser lo más grande posible.
- Un círculo puede envolver los bordes
- Se marca con un círculo "no me importa" (X) solo si ayuda a minimizar la ecuación

Mapa de 4-entradas

A	*B*	*C*	*D*	*Y*
0	0	0	0	1
0	0	0	1	0
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

Y

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

Mapa de 4-entradas

A	*B*	*C*	*D*	*Y*
0	0	0	0	1
0	0	0	1	0
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

Y

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

$$Y = \overline{A}C + \overline{A}BD + A\overline{B}\,\overline{C} + \overline{B}\,\overline{D}$$

K-Map con Don't cares

A	*B*	*C*	*D*	*Y*
0	0	0	0	1
0	0	0	1	0
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	X
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	X
1	0	1	1	X
1	1	0	0	X
1	1	0	1	X
1	1	1	0	X
1	1	1	1	X

Y

CD \ *AB*	00	01	11	10
00	1	0	X	1
01	0	X	X	1
11	1	1	X	X
10	1	1	X	X

K-Map con Don’t cares

A	B	C	D	Y
0	0	0	0	1
0	0	0	1	0
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	X
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	X
1	0	1	1	X
1	1	0	0	X
1	1	0	1	X
1	1	1	0	X
1	1	1	1	X

Y — CD \ AB	00	01	11	10
00	1	0	X	1
01	0	X	X	1
11	1	1	X	X
10	1	1	X	X

$$Y = A + \overline{B}\,\overline{D} + C$$

Simplificar funciones con 5 o mas variables

- El algoritmo de Quine-McCluskey es funcionalmente idéntico al mapeo de Karnaugh, pero la forma tabular lo hace más eficiente para su uso en algoritmos informáticos y también brinda una forma determinístíca de verificar que se haya alcanzado la forma mínima de una función booleana. A veces se lo denomina método de tabulación.

	Column I	Column II	Column III
group 0	0 0000 ✓	0, 1 000– ✓	0, 1, 8, 9 –00–
group 1	1 0001 ✓	0, 2 00–0 ✓	0, 2, 8, 10 –0–0
	2 0010 ✓	0, 8 –000 ✓	~~0, 8, 1, 9 –00–~~
	8 1000 ✓	1, 5 0–01	~~0, 8, 2, 10 –0–0~~
group 2	5 0101 ✓	1, 9 –001 ✓	2, 6, 10, 14 --10
	6 0110 ✓	2, 6 0–10 ✓	~~2, 10, 6, 14 --10~~
	9 1001 ✓	2, 10 –010 ✓	
	10 1010 ✓	8, 9 100– ✓	
group 3	7 0111 ✓	8, 10 10–0 ✓	
	14 1110 ✓	5, 7 01–1	
		6, 7 011–	
		6, 14 –110 ✓	
		10, 14 1–10 ✓	

Random example

Number of input variables: 4 Allow Don't-Care: no

Truth table:

	x_3	x_2	x_1	x_0	y
0:	0	0	0	0	1
1:	0	0	0	1	1
2:	0	0	1	0	1
3:	0	0	1	1	0
4:	0	1	0	0	0
5:	0	1	0	1	1
6:	0	1	1	0	1
7:	0	1	1	1	1
8:	1	0	0	0	1
9:	1	0	0	1	1
10:	1	0	1	0	1
11:	1	0	1	1	0
12:	1	1	0	0	0
13:	1	1	0	1	0
14:	1	1	1	0	1
15:	1	1	1	1	0

Implicants (Order 0):

	x_3	x_2	x_1	x_0	
0:	0	0	0	0	$\rightarrow$
1:	0	0	0	1	$\rightarrow$
2:	0	0	1	0	$\rightarrow$
5:	0	1	0	1	$\rightarrow$
6:	0	1	1	0	$\rightarrow$
7:	0	1	1	1	$\rightarrow$
8:	1	0	0	0	$\rightarrow$
9:	1	0	0	1	$\rightarrow$
10:	1	0	1	0	$\rightarrow$
14:	1	1	1	0	$\rightarrow$

Implicants (Order 1):

	x_3	x_2	x_1	x_0	
0, 1:	0	0	0	-	$\rightarrow$
0, 2:	0	0	-	0	$\rightarrow$
0, 8:	-	0	0	0	$\rightarrow$
1, 5:	0	-	0	1	✓
1, 9:	-	0	0	1	$\rightarrow$
2, 6:	0	-	1	0	$\rightarrow$
2, 10:	-	0	1	0	$\rightarrow$
5, 7:	0	1	-	1	✓
6, 7:	0	1	1	-	✓
6, 14:	-	1	1	0	$\rightarrow$
8, 9:	1	0	0	-	$\rightarrow$
8, 10:	1	0	-	0	$\rightarrow$
10, 14:	1	-	1	0	$\rightarrow$

Implicants (Order 2):

	x_3	x_2	x_1	x_0	
0, 1, 8, 9:	-	0	0	-	✓
0, 2, 8, 10:	-	0	-	0	✓
2, 6, 10, 14:	-	-	1	0	✓

Prime implicant chart:

	x_3	x_2	x_1	x_0	0	1	2	5	6	7	8	9	10	14	
0, 1, 8, 9:	-	0	0	-	○	○					○	●			$(\bar{x}_2\bar{x}_1)$
0, 2, 8, 10:	-	0	-	0	○		○				○		○		$(\bar{x}_2\bar{x}_0)$
2, 6, 10, 14:	-	-	1	0			○		○				○	●	$(x_1\bar{x}_0)$
1, 5:	0	-	0	1		○		○							$(\bar{x}_3\bar{x}_1x_0)$
5, 7:	0	1	-	1				○		○					$(\bar{x}_3x_2x_0)$
6, 7:	0	1	1	-					○	○					$(\bar{x}_3x_2x_1)$

Extracted essential prime implicants: $(\bar{x}_2\bar{x}_1)$, $(x_1\bar{x}_0)$

Reduced prime implicant chart (Iteration 0):

	x_3	x_2	x_1	x_0	5	7	
5, 7:	0	1	-	1	●	●	$(\bar{x}_3x_2x_0)$

Extracted essential prime implicants: $(\bar{x}_3x_2x_0)$

Minimal boolean expression:

$$y = (\bar{x}_2\bar{x}_1) \vee (x_1\bar{x}_0) \vee (\bar{x}_3x_2x_0)$$

Simplificar funciones con 5 o mas variables

- En el sitio

https://www.mathematik.uni-marburg.de/~thormae/lectures/ti1/code/qmc/

La función que se minimiza se puede ingresar mediante una tabla de verdad que representa la función y = f(xn,...,x1, x0). Puede editar manualmente esta función haciendo clic en los elementos grises en la columna y, alternativamente, puede generar una función aleatoria presionando el botón "Ejemplo aleatorio".

Random example

Number of input variables: 3   Allow Don't-Care: no

Truth table:

	x_2	x_1	x_0	y
0:	0	0	0	0
1:	0	0	1	0
2:	0	1	0	0
3:	0	1	1	0
4:	1	0	0	0
5:	1	0	1	0
6:	1	1	0	0
7:	1	1	1	0

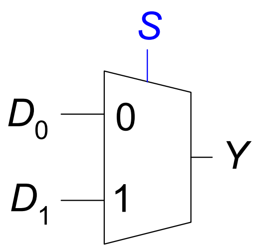
Bloques constructivos combinacionales

- Multiplexores
- Decodificadores

Multiplexor (Mux)

- **Selecciona** entre una de N entradas para conectar a la salida
- Hay 2^N líneas de entrada y N líneas de selección cuyas combinaciones de bits determinan cual entrada se selecciona

Ejemplo



S	D_1	D_0	Y
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

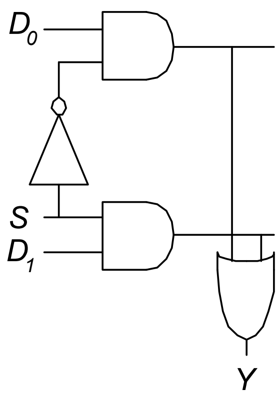
S	Y
0	D_0
1	D_1

Implementar Mux’s

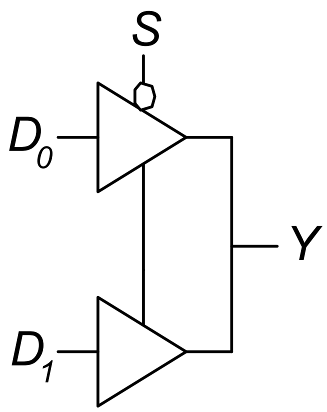
- **Puertas Lógicas**
  - Forma SOP

S \ D_0D_1	00	01	11	10
0	0	0	1	1
1	0	1	1	0

$$Y = D_0\overline{S} + D_1S$$



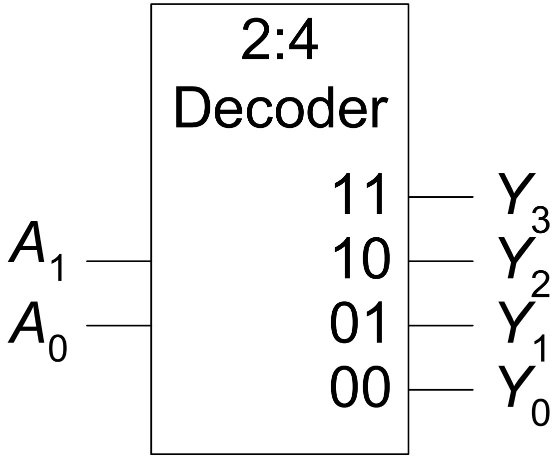
- **Triestados**
  – Para un Mux de N entradas, usa N triestados
  – Encienda exactamente uno para seleccionar la entrada adecuada



Ejemplos de circuitos MUX

Decodificadores

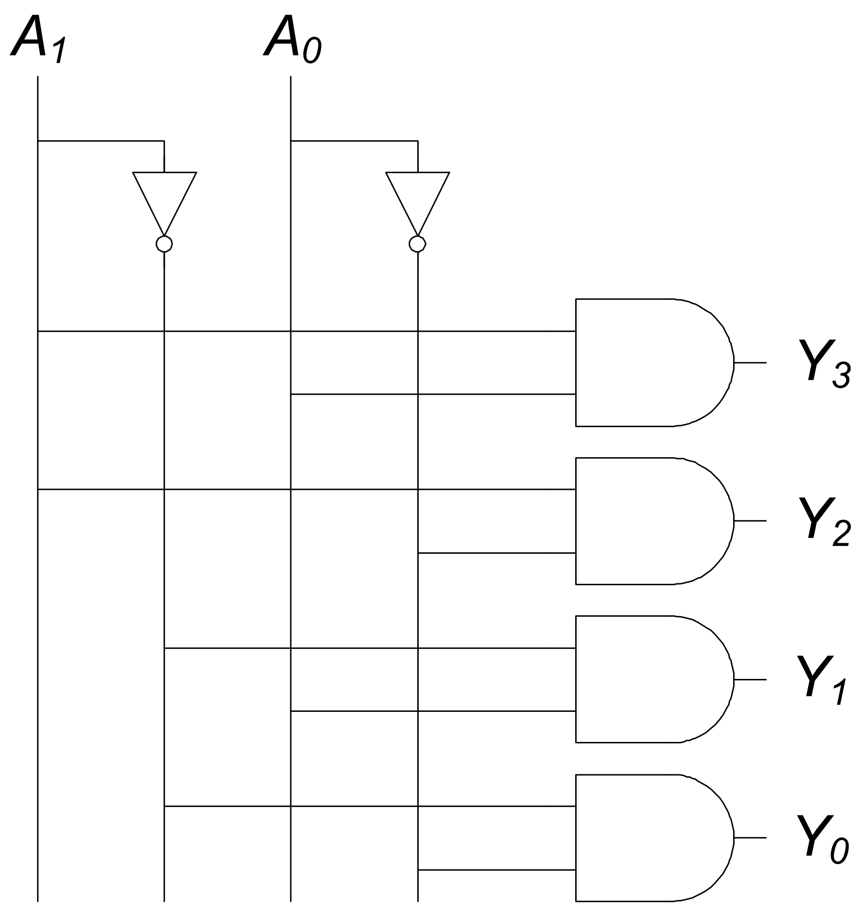
- N entradas, 2N salidas
- Solo una salida ALTA a la vez



A_1	A_0	Y_3	Y_2	Y_1	Y_0
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	0	0	0

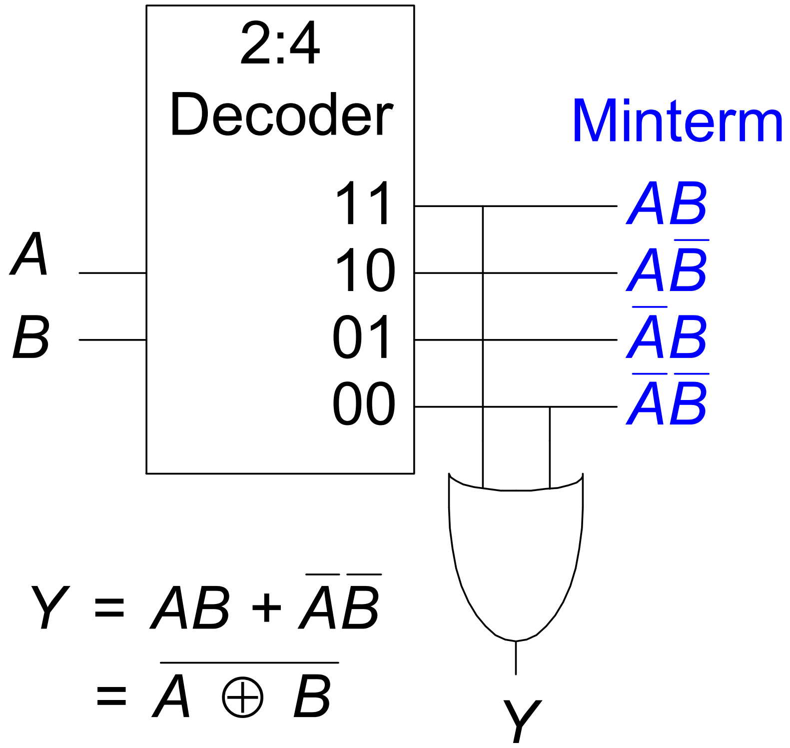
Ejemplos de decodificadores

Implementación de un decodificador



Lógica usando decodificadores

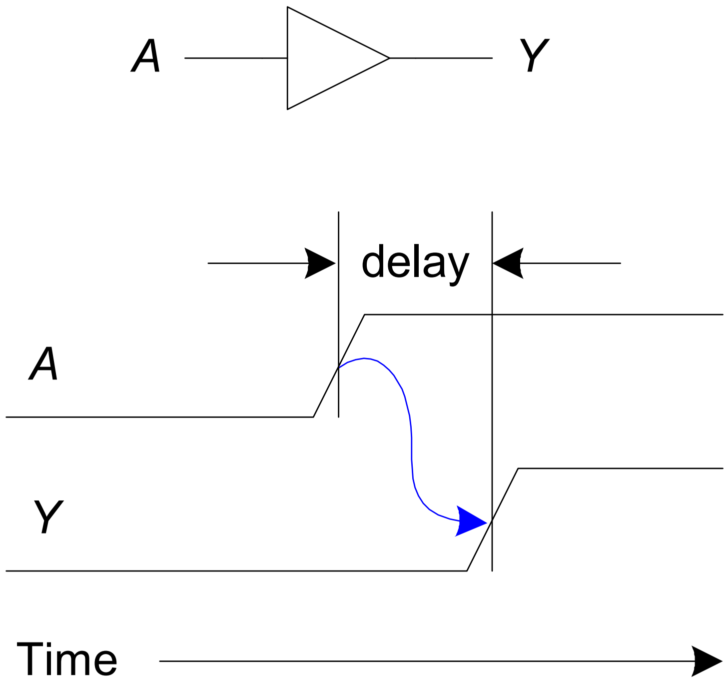
Mintérmino OR



$$Y = AB + \overline{A}\,\overline{B} = \overline{A \oplus B}$$

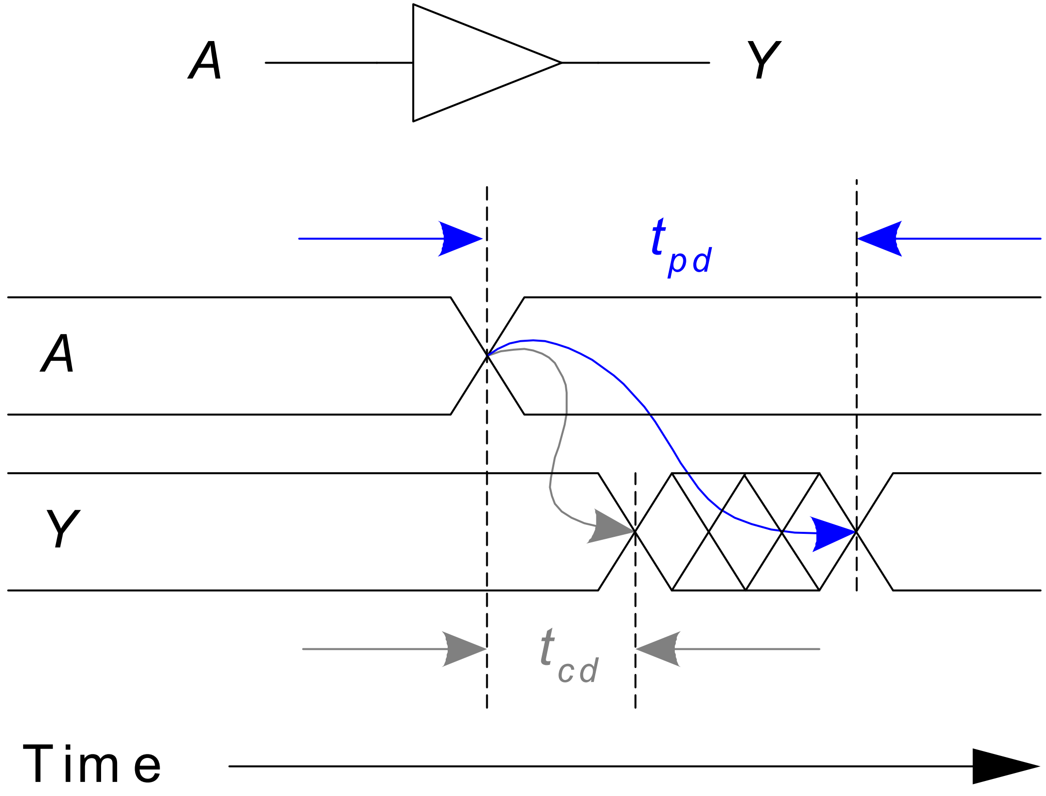
Sincronización

- Retardo entre cambio de entrada y cambio de salida¿
- Cómo construir circuitos rápidos?



Retraso de propagación y contaminación

- Demora de Propagación: t_{pd} = maxima demora desde entrada a salida
- **Demora de Contaminación:** t_{cd} = min delay from input to output



Retraso de propagación y contaminación

- El retraso es causado por

-- Capacitancia y resistencia en un circuito

-- Limitación de la velocidad de la luz

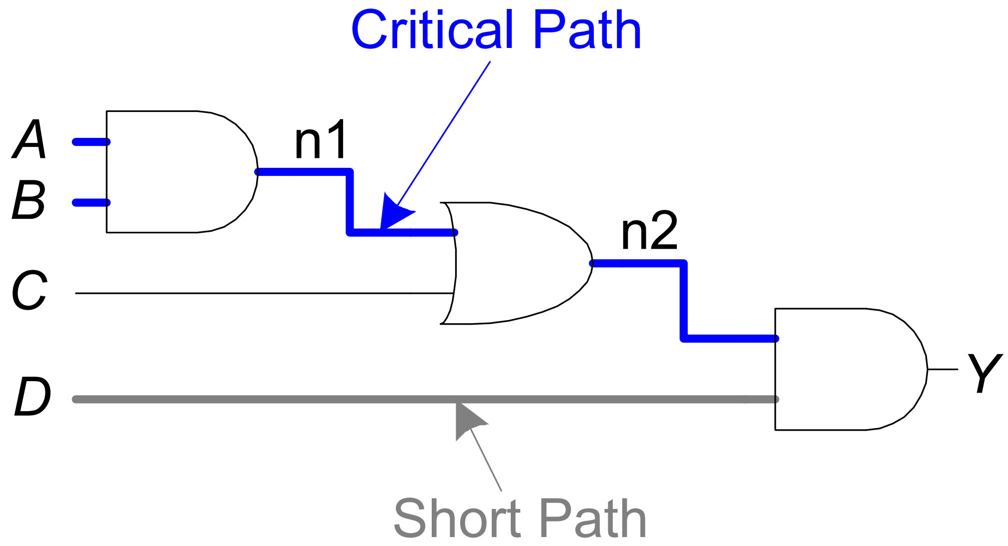
- Razones por las que tpd y tcd pueden ser diferentes:

-- Diferentes retrasos ascendentes y descendentes

-- Múltiples entradas y salidas, algunas de las cuales son más rápidas que otras

-- Los circuitos se ralentizan cuando están calientes y se aceleran cuando están fríos.

Rutas críticas (largas) y cortas



Ruta (Path) crítica (Larga): $t_{pd} = 2t_{pd\_AND} + t_{pd\_OR}$

Trayecto corto: $t_{cd} = t_{cd\_AND}$

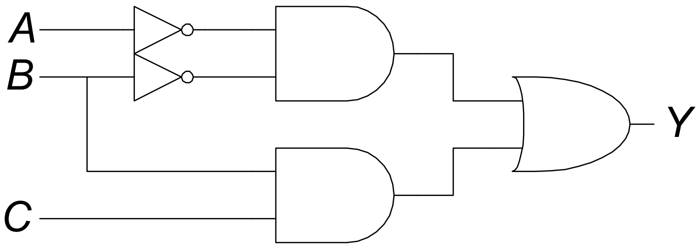
Fallas (glitches)

Cuando un solo cambio de entrada hace que

una salida cambie varias veces

Ejemplo de una falla (glitch)

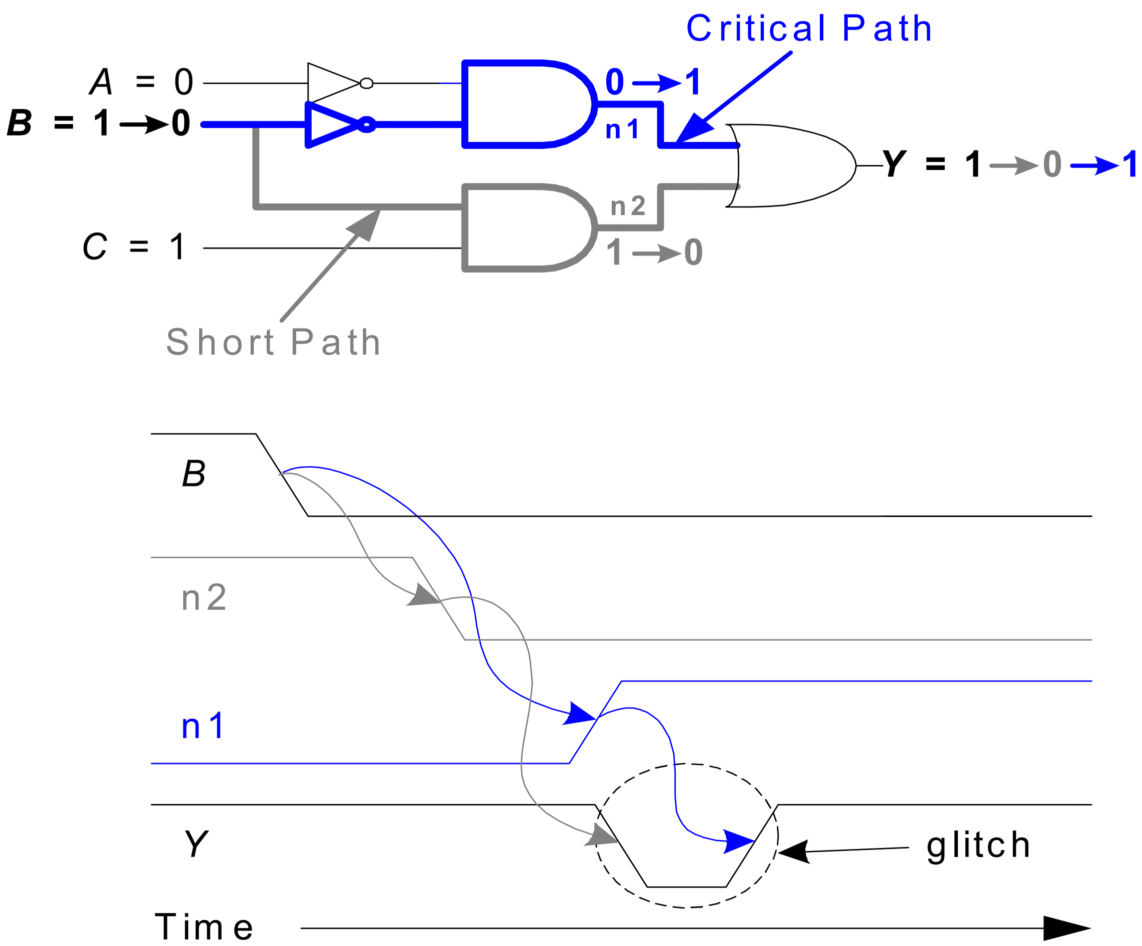
¿Que sucede cuando A=0, C=1, B cae?



Y: C \ AB	00	01	11	10
0	1	0	0	0
1	1	1	1	0

$$Y = \bar{A}\bar{B} + BC$$

Ejemplo de una falla (glitch)



¿Por qué entender los fallos?

- Los fallos no causan problemas debido a las convenciones de diseño síncrono.
- Es importante reconocer un fallo: en simulaciones o en osciloscopio.
- No se pueden deshacer de todos los problemas técnicos: las transiciones simultáneas en múltiples entradas también pueden causar problemas técnicos.

Fin AOC_02.pptx