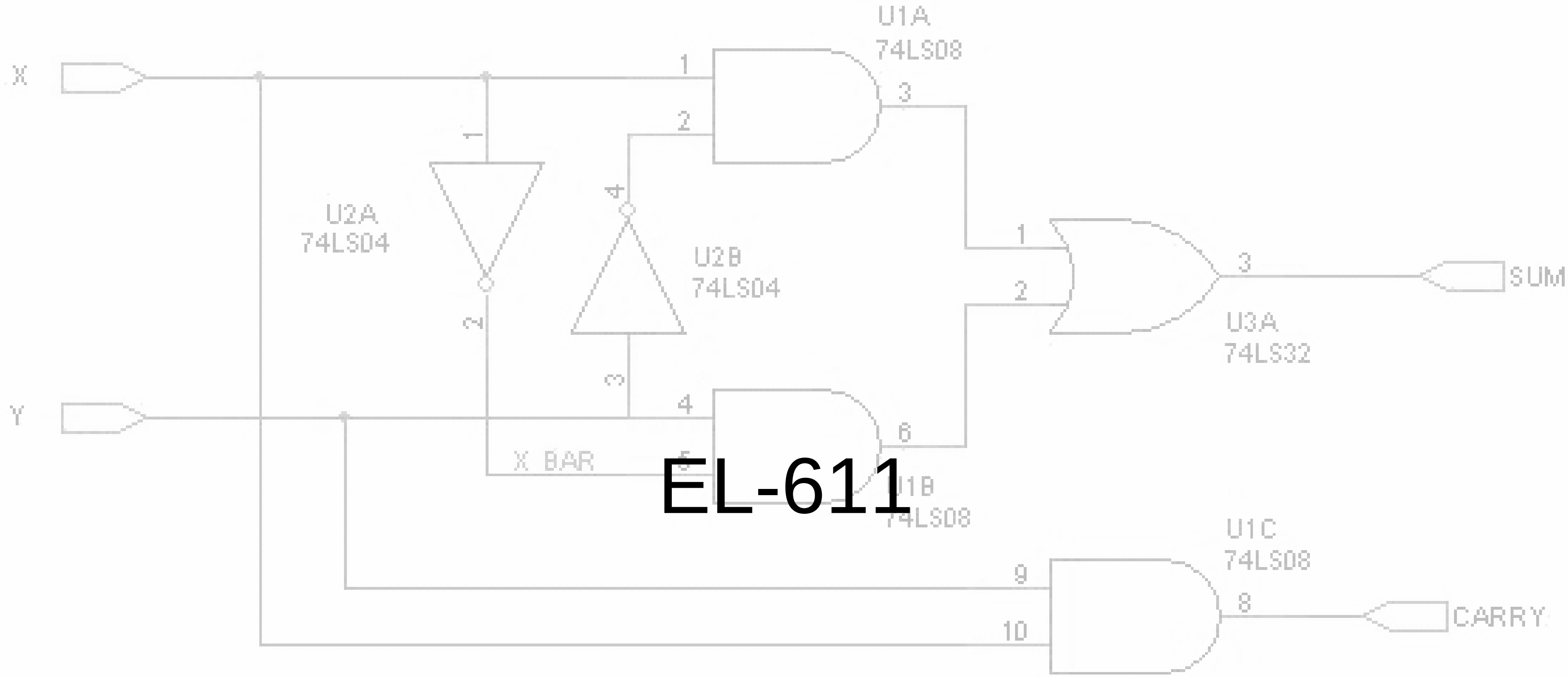


EL-611

Descomposición Funcional

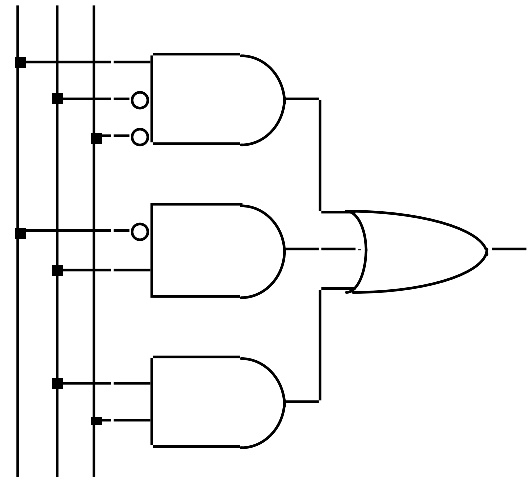
Descomposición Funcional

Idea de utilizar funciones mas simples para implementar funciones complejas

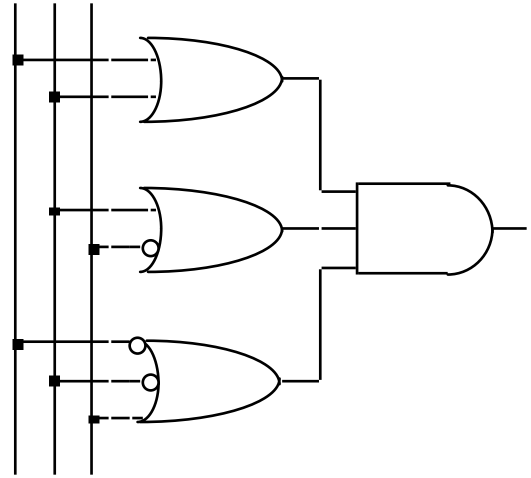
- Funciones NAND/NOR
  - NAND/NOR de dos niveles
  - NAND/NOR de varios niveles
- Descomposición Funcional Disjunta
  - Simple
  - Compleja
    - Multiple
    - Iterativa

Implementación Lógica de Dos Niveles

- Suma-de-productos
  - Compuertas AND para formar términos productos (minitérminos)
  - Compuertas OR para formar sumas

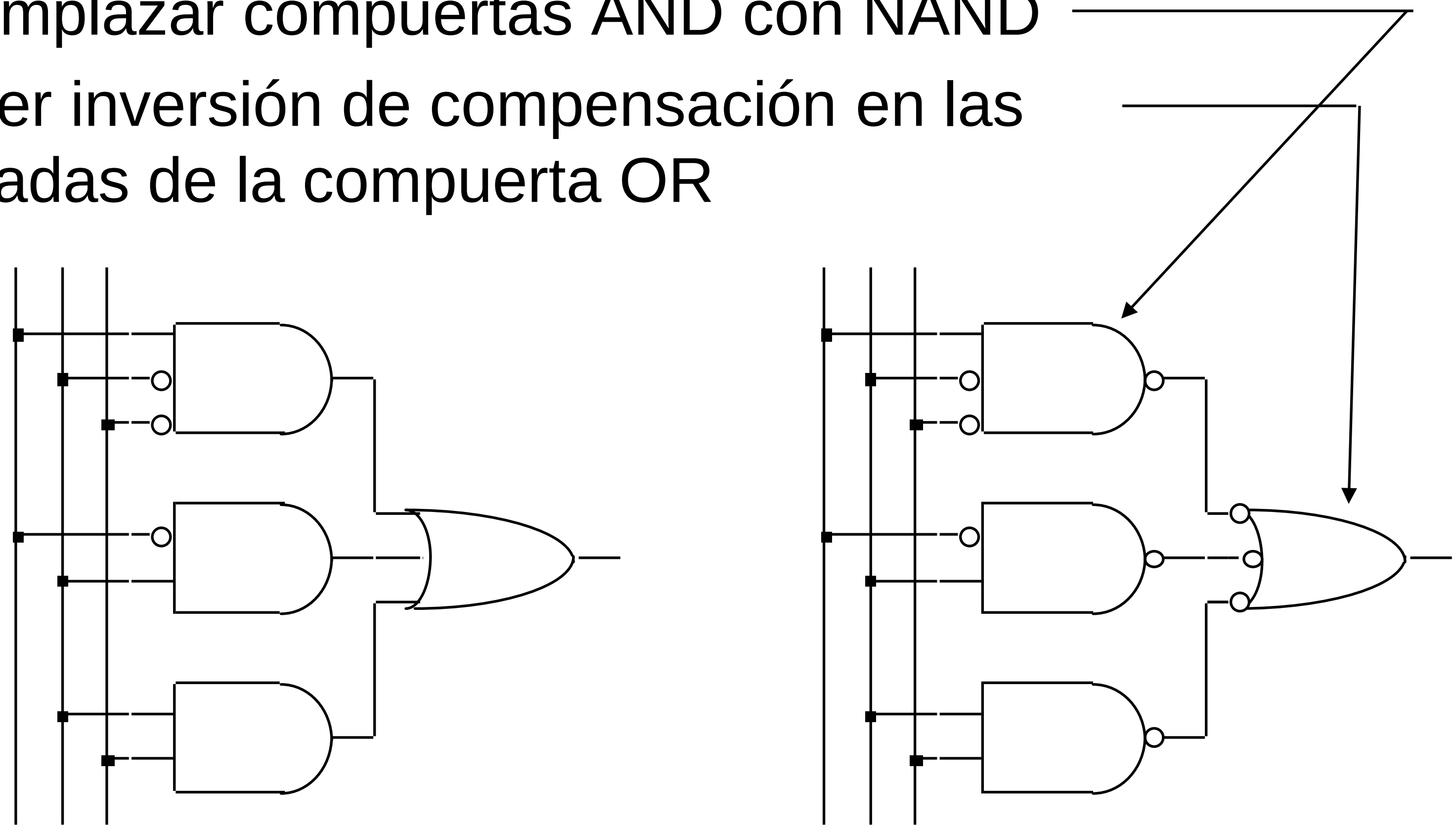


- Productos-de-sumas
  - Compuertas OR para formar términos sumas (maxitérminos)
  - Compuertas AND para formar productos



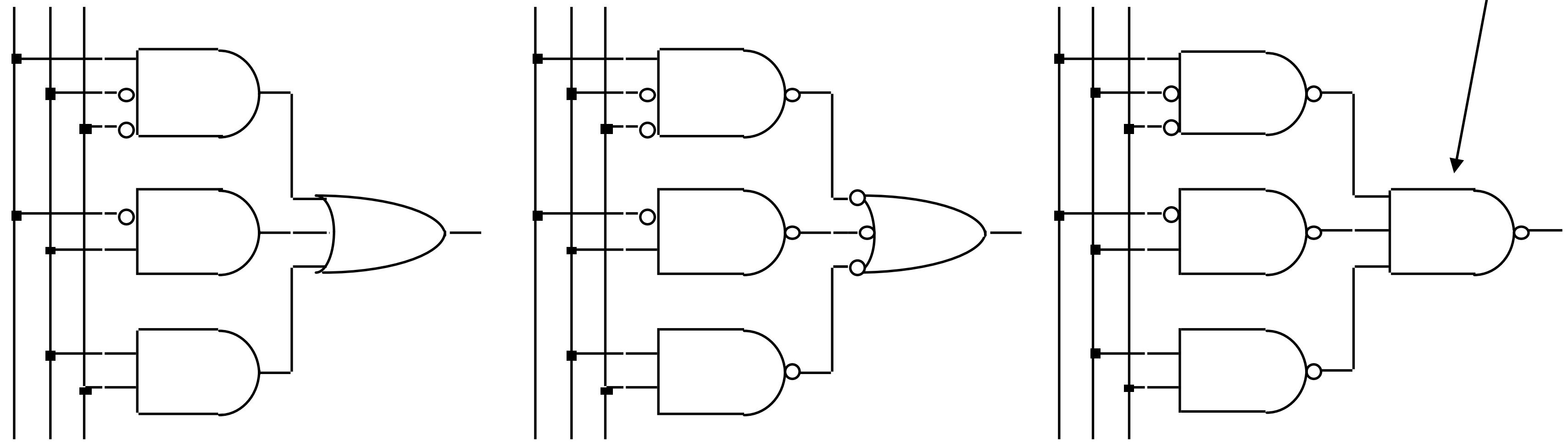
Lógica de dos niveles utilizando compuertas NAND

- Reemplazar compuertas AND con NAND
- Poner inversión de compensación en las entradas de la compuerta OR



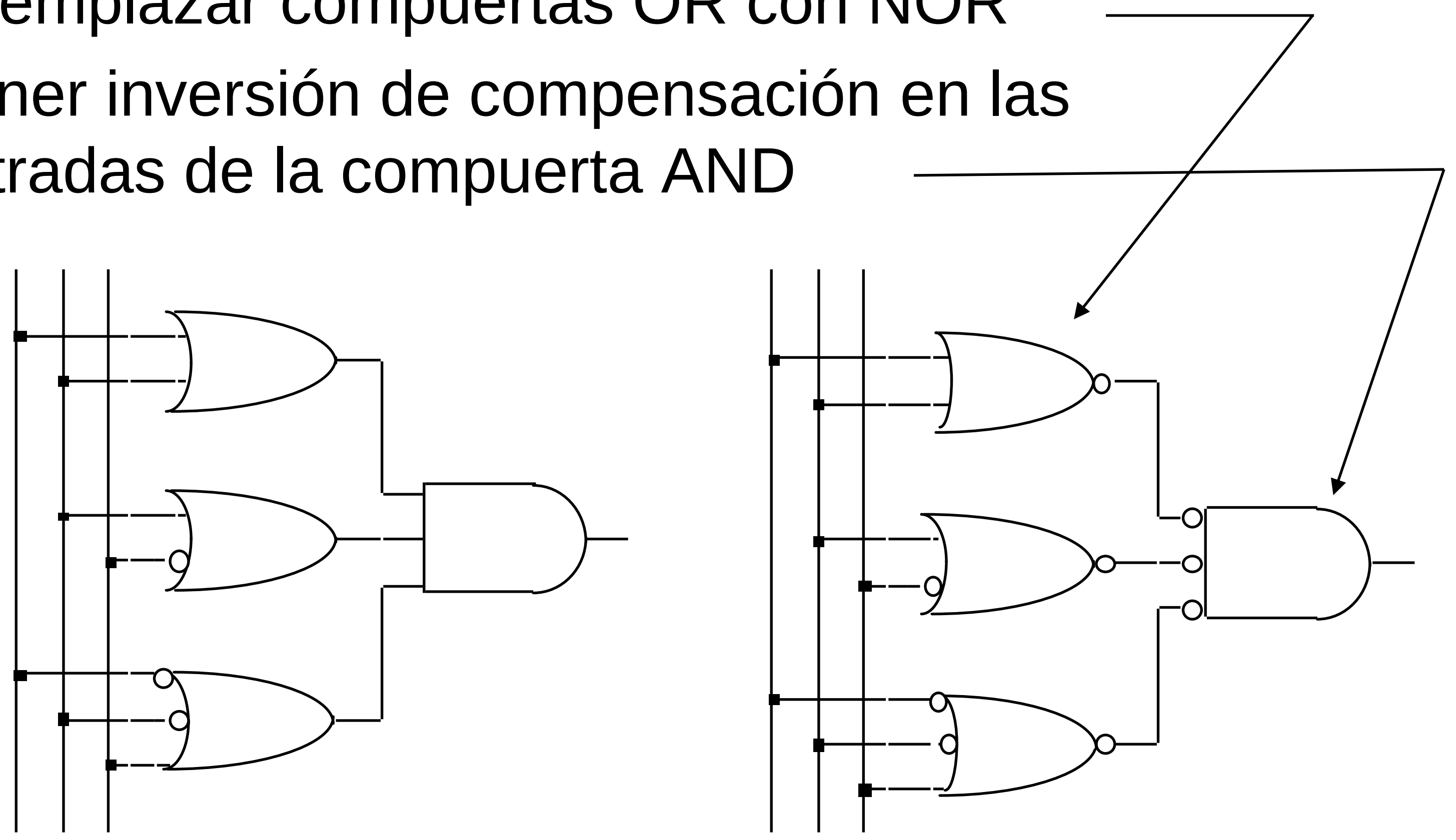
Lógica de dos niveles utilizando compuertas NAND (cont')

- Compuerta OR con entradas invertidas es una NAND
  - de Morgan: A' + B' = (A • B)'
- Circuito NAND-NAND de dos niveles
  - entradas invertidas no se cuentan
  - en un circuito típico, la inversión es efectuada una sola vez y distribuida



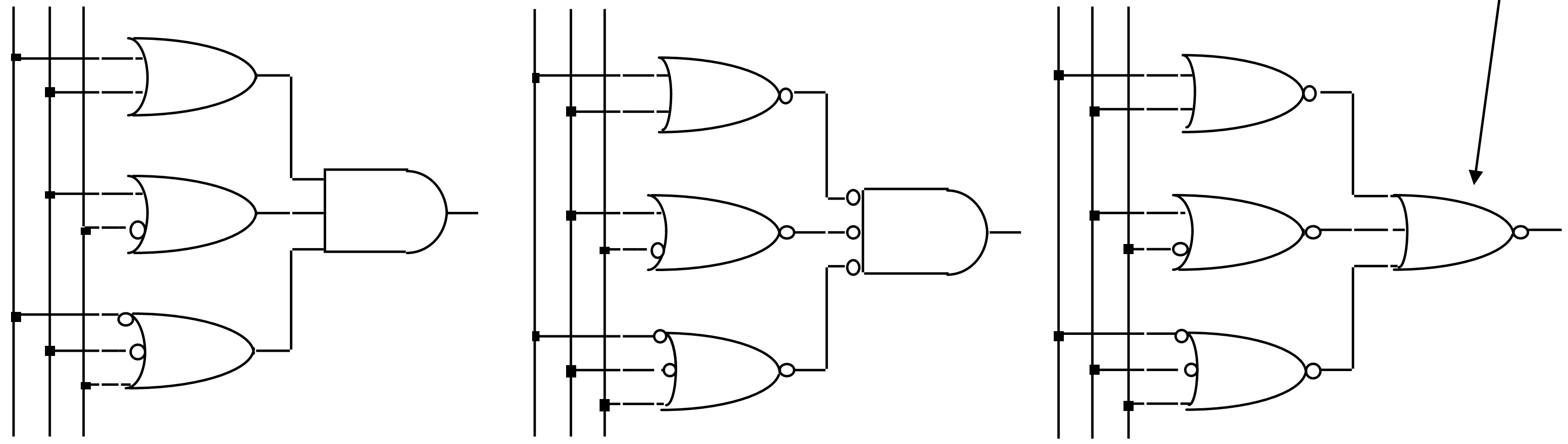
Lógica de dos niveles utilizando NOR

- Reemplazar compuertas OR con NOR
- Poner inversión de compensación en las entradas de la compuerta AND



Lógica de dos niveles utilizando NOR (cont')

- Compuerta AND con entradas invertidas es una NOR
  - de Morgan: $A' \bullet B' = (A + B)'$
- Circuito NOR-NOR de dos niveles
  - entradas invertidas no se cuentan
  - en un circuito típico, la inversión es realizado una sola vez y la señal distribuida



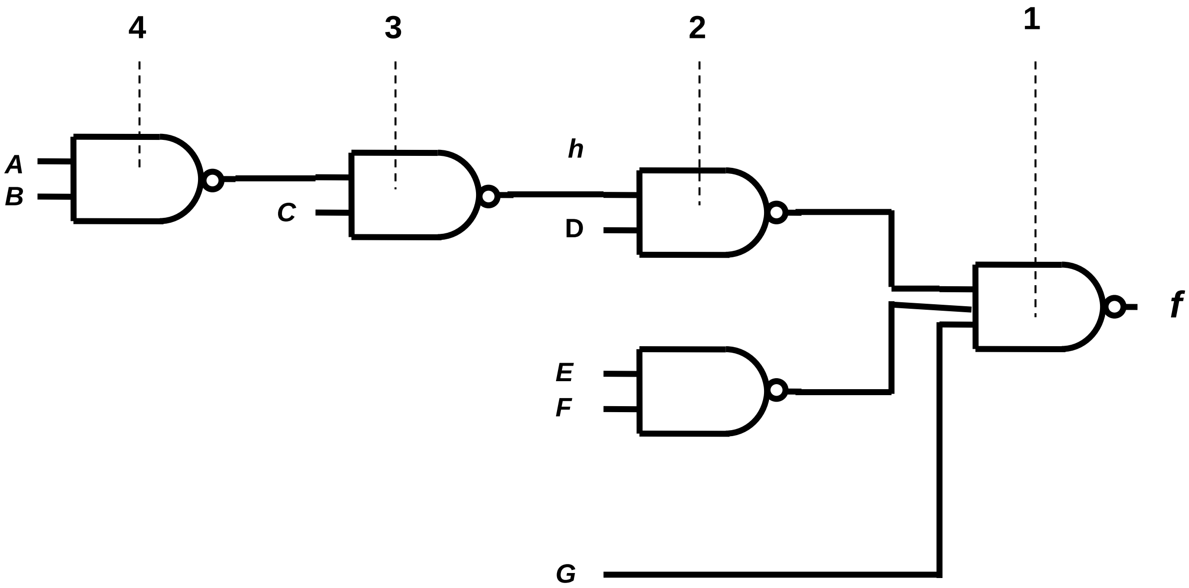
Lógica de dos niveles utilizando compuertas NAND y NOR

- Circuitos NAND-NAND y NOR-NOR
  - Ley de Morgan: $(A + B)' = A' \cdot B'$ $\quad (A \cdot B)' = A' + B'$
  - variante: $A + B = (A' \cdot B')'$ $\quad (A \cdot B) = (A' + B')'$
- En otras palabras:
  - OR es lo mismo que NAND con entradas complementadas
  - AND es lo mismo que NOR con entradas complementadas
  - NAND es lo mismo que OR con entradas complementadas
  - NOR es lo mismo que AND con entradas complementadas

CONVERSION DE AND/OR y OR/AND a NAND y NOR

Para convertir / a	Suma de Productos (AND/OR)	Productos de Suma (OR/AND)
NAND'S	1.- Poner entre paréntesis cada producto (.) separado por sumas (+) 2.- Complementar ($\bar{}$) todo literal que no hace AND con ningún otro literal. 3.- Reemplazar todo los símbolos sumas (+) y productos (.) por el símbolo NAND (|)	1.- Complementar ($\bar{}$) todos los literales de cada factor suma (+), excepto aquellos que no hacen OR con otros literales. 2.- Reemplazar todos los símbolos sumas (+) y productos (.) por el símbolo NAND (|) 3.- Poner una barra de complementación ($\bar{}$) sobre toda la función.
NOR'S	1.- Poner entre paréntesis todos los productos (.) separados por sumas (+) 2.- Complementar ($\bar{}$) todo los literales de cada factor producto (.), excepto aquellos que no hacen AND con otros literales. 3.- Reemplazar todos los símbolos suma (+) y, productos (.) por el símbolo NOR (↓) 4.- Poner una barra de complementación ($\bar{}$) sobre toda la función.	1.- Complementar ($\bar{}$) todos los literales que no hacen OR con otros literales. 2.- Reemplazar todos los símbolos sumas (+) y productos (-) por el símbolo NOR (↓)

NAND/NOR de varios niveles



$$f = \overline{(\overline{h \cdot D}) \cdot (\overline{E \cdot F}) \cdot G} = h \cdot D + E \cdot F + \overline{G}$$

$$h = \overline{(\overline{A \cdot B}) \cdot C} = A \cdot B + \overline{C}$$

$$f(A, B, C, D, E, F, G) = (A \cdot B + \overline{C}) \cdot D + E \cdot F + \overline{G}$$

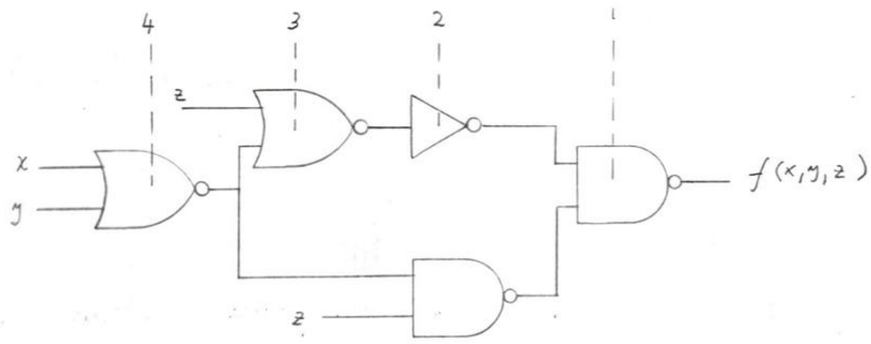
NAND/NOR de varios niveles

- Las compuertas se enumeran desde la salida hacia la entrada
- Cuando la función NAND está ubicada en un nivel impar, realiza una función OR y cuando esta ubicada en un nivel par realiza una operación AND
- Análogamente, para compuertas NOR, realizan la operación OR cuando están ubicadas en niveles pares y la operación AND cuando se ubican en niveles impares
- Las entradas a compuertas de niveles impares aparecen complementadas en la expresión de salida (por ejemplo, C y G), y no complementadas cuando entran en niveles pares (por ejemplo, A, B, D, E y F)

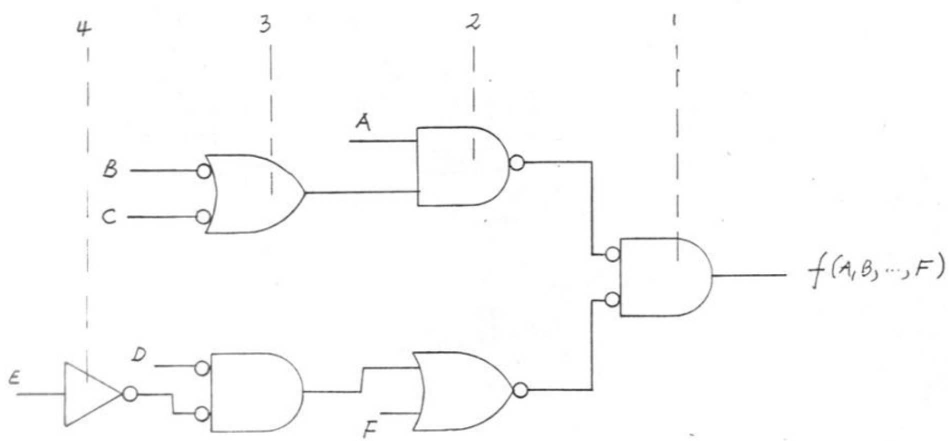
Tabla Diseño NAND/NOR varios niveles

	Nivel Compuerta	
Tipo Lógico	Nivel Impar	Nivel Par
NAND	OR	AND
NOR	AND	OR
Literal Entrada	Complementada	No Complementada

Ejemplos NAND/NOR de varios niveles

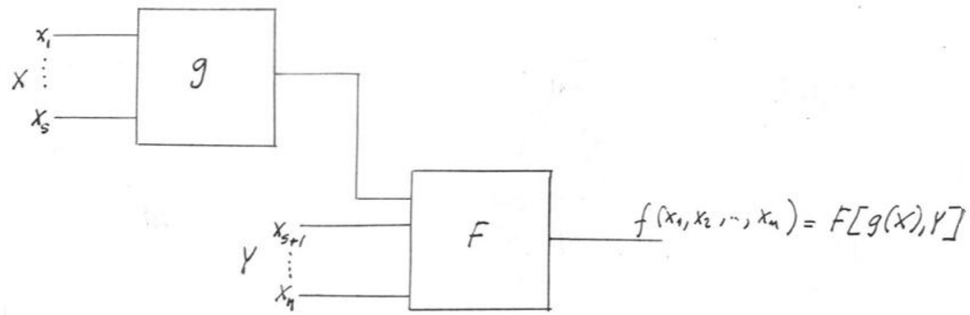


$$f(x,y,z) = (x+y)\cdot\bar{z} + \bar{x}\cdot\bar{y}\cdot z$$

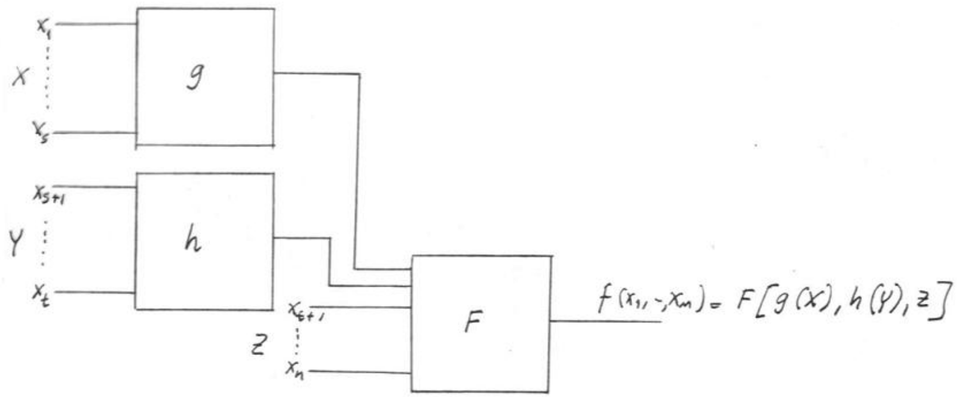


$$f(A,B,\ldots,F) = (\bar{B}+\bar{C})\cdot A\cdot(\bar{D}E+F)$$

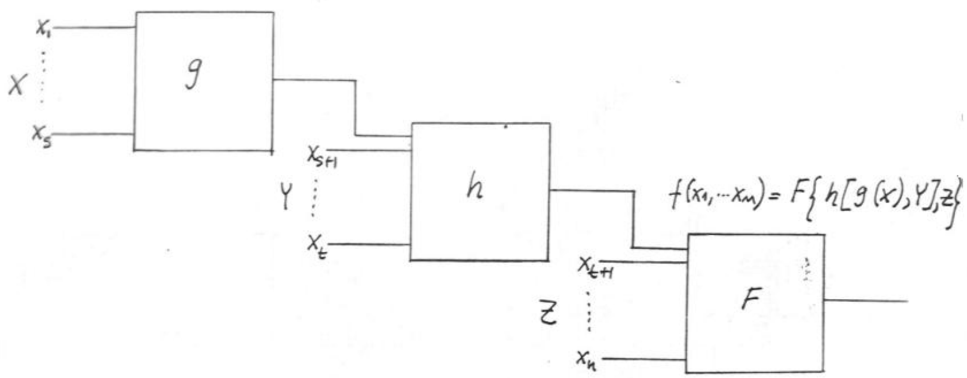
Descomposición Funcional Disjunta



Simple



Compleja Múltiple



Compleja Iterativa

Ejemplo DFDS

$$f(x_1, x_2, x_3, x_4) = x_1\bar{x}_2x_3x_4 + x_1x_2x_3\bar{x}_4 + \bar{x}_1\bar{x}_2\bar{x}_3\bar{x}_4 + \bar{x}_1x_2\bar{x}_3x_4$$
$$= x_1x_3(x_2\bar{x}_4 + \bar{x}_2x_4) + \bar{x}_1\bar{x}_3(\bar{x}_2\bar{x}_4 + x_2x_4)$$
$$= x_1x_3(x_2 \oplus x_4) + \bar{x}_1\bar{x}_3(\overline{x_2 \oplus x_4})$$
$$= x_1x_3\varphi(x_2, x_4) + \bar{x}_1\bar{x}_3\overline{\varphi(x_2, x_4)}$$

$$\therefore\ f\left[\varphi(x_2, x_4), x_1, x_3\right]$$
$$\varphi(x_2, x_4) = x_2 \oplus x_4 = x_2\bar{x}_4 + \bar{x}_2x_4$$

Método para encontrar una DFDS

Dada la función: $f(x_1, x_2, x_3, x_4) = \sum m(4,5,6,7,8,13,14,15)$

Veamos si existe la DFDS: $F[\varphi(x_1, x_4), x_2, x_3]$

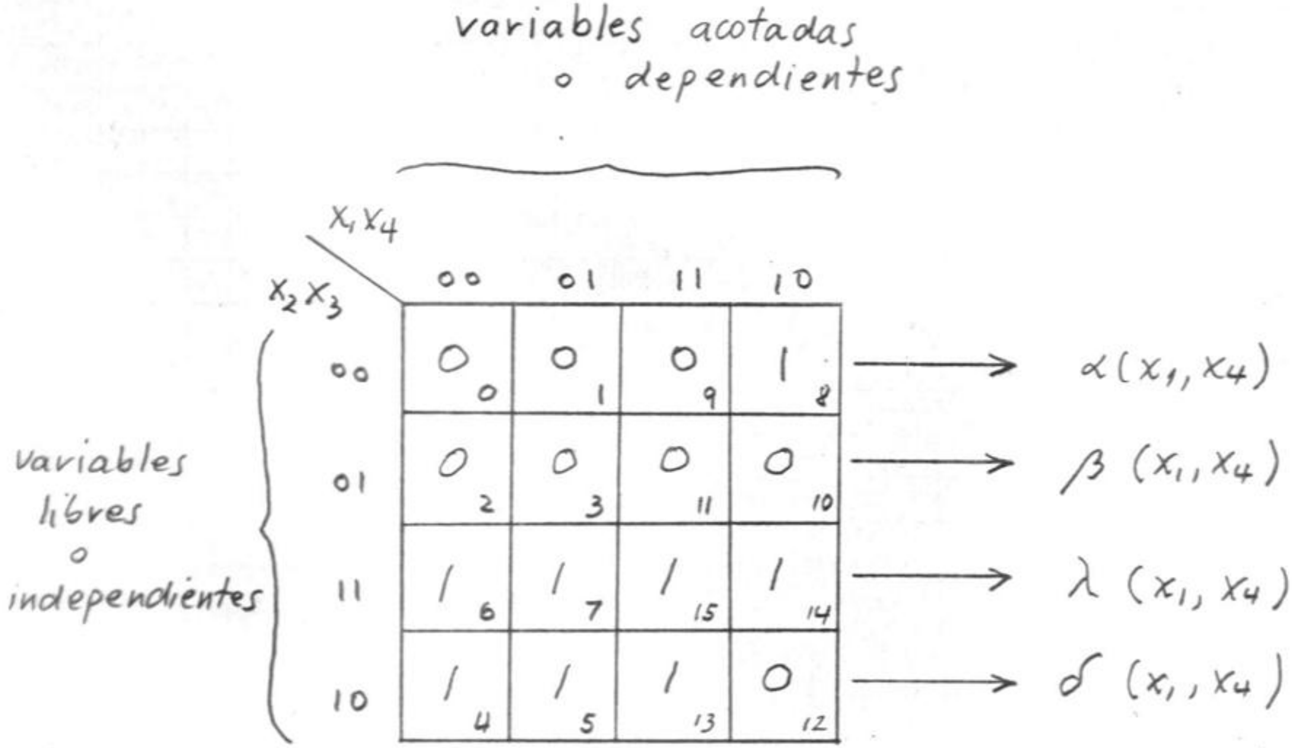
En general se puede escribir:

$$f(x_1, x_2, x_3, x_4) = \bar{x}_2\bar{x}_3\alpha(x_1, x_4) + \bar{x}_2 x_3\beta(x_1, x_4) + x_2 x_3\lambda(x_1, x_4) + x_2\bar{x}_3\delta(x_1, x_4)$$

lo cual es válido para cualquier función, sea o no descomponible

Para encontrar la función φ que debe ser única, utilizamos los Mapas de Particiones

Mapa de Particiones



$$f(x_1, x_2, x_3, x_4) = \sum m(4,5,6,7,8,13,14,15)$$

Multiplicidad de Columna

- Para definir en que casos se puede obtener una DFDS se define la Multiplicidad de Columnas (MC)
- La MC son las diferentes distribuciones de 0's y 1's vistas por columnas
- **Teorema**: Un Mapa de Particiones corresponde a una DFDS sii la Multiplicidad de Columnas es menor o igual a 2

Otros Mapas de Particiones

x_1x_2 \ x_3x_4	00	01	11	10
00	0	1	3	2
01	4	5	7	6
11	12	13	15	14
10	8	9	11	10

x_1x_3 \ x_2x_4	00	01	11	10
00	0	1	5	4
01	2	3	7	6
11	10	11	15	14
10	8	9	13	12

x_1x_4 \ x_2x_3	00	01	11	10
00	0	2	6	4
01	1	3	7	5
11	9	11	15	13
10	8	10	14	12

$$f(x_1, x_2, x_3, x_4) = \sum m(4,5,6,7,8,13,14,15)$$

Funciones Parcialmente Especificadas

	x_1 = 0				x_1 = 1				
x_3x_4 \ x_2x_5	00	01	11	10	00	01	11	10	
00	~~0~~	1	9	8	16	17	25	24	$\longrightarrow \varphi$
01	2	3	11	10	18	19	27	~~26~~	$\longrightarrow \bar{\varphi}$
11	6	7	~~15~~	14	22	~~23~~	31	30	$\longrightarrow \varphi$
10	4	5	13	12	~~20~~	21	29	28	$\longrightarrow 0$

$$f(x_1, x_2, x_3, x_4, x_5) = \sum m(1,2,7,9,10,18,19,25,31) +$$
$$+ \phi\, m(0,15,20,23,26)$$

Descomposición Funcional Disjunta Compleja

Existen dos tipos

- Iterativa $f(x) = F[\lambda[\varphi(A), B, C]$
- Múltiple $f(x) = F[\lambda(A), \gamma(B), \delta(C), D]$

Teorema 1: Sea f una función para la cual existen dos DFDS:

$$f(x) = F[\lambda(A, B), C]$$

$$f(x) = G[\varphi(A), B, C]$$

Entonces existe una DFDC **Iterativa** de la forma:

$$f(x) = F[E[h(A), B], C]$$

$$donde,$$

$$E[h(A), B] = \lambda(A, B)$$

Ejemplo

$$f(x_1, x_2, x_3, x_4, x_5) = \sum m(5, 10, 11, 14, 17, 21, 26, 30)$$

$$f(x) = F[\lambda(x_1, x_3, x_5), x_2, x_4]$$

$$f(x) = G[\varphi(x_1, x_3), x_2, x_4, x_5]$$

x_2x_4 \ x_3x_5	x_1 = 0: 00	01	11	10	x_1 = 1: 00	01	11	10	
00	0	1	5	4	16	17	21	20	$\rightarrow \lambda(x_1, x_3, x_5)$
01	2	3	7	6	18	19	23	22	
11	10	11	15	14	26	27	31	30	$\rightarrow \overline{\lambda(x_1, x_3, x_5)}$
10	8	9	13	12	24	25	29	28	

x_1x_3 \ x_4x_5	x_2 = 0: 00	01	11	10	x_2 = 1: 00	01	11	10
00	0	1	2	3	8	9	11	10
01	4	5	7	6	12	13	15	14
11	20	21	23	22	28	29	31	30
10	16	17	19	18	24	25	27	26
		$\uparrow$ $\varphi(x_1, x_3)$					$\uparrow$ $\overline{\varphi(x_1, x_3)}$	

Descomposición Funcional Disjunta Compleja

Teorema 2:

Sea *f(x)* una función para la cual existen dos DFDS de la forma:

$$f(x) = F[\lambda(A), B], y$$

$$f(x) = G[\varphi(B), A]$$

Entonces existe una DFDC Múltiple de la forma:

$$f(x) = H[\lambda(A), \varphi(B)]$$

Descomposición Funcional Disjunta Compleja

Teorema 3:

Sea *f(x)* una función para la cual existen dos DFDS de la forma:

$$f(x) = F[\lambda(A), B, C], y$$

$$f(x) = G[\varphi(B), A, C]$$

Entonces existe una DFDC Múltiple de la forma:

$$f(x) = H[\lambda(A), \varphi(B), C]$$