

Fundamentos de Computadores

Subsistemas Combinacionales

Objetivos

- Conceptuales:
 - Escalabilidad de los diseños
 - Diseño modular
 - Diseños alternativos
 - Subsistemas de propósito específico
 - Subsistemas de propósito general y dispositivos lógicos programables

Objetivos

- Procedimentales:
 - Interpretación de hojas de características de subsistemas combinacionales y dispositivos lógicos programables
 - Diseño de subsistemas combinacionales: decodificadores, codificadores, multiplexores, etc.
 - Diseño y análisis de circuitos con subsistemas combinacionales: decodificadores, comparadores, multiplexores, etc.
 - Diseño con dispositivos lógicos programables

Objetivos

- Actitudinales:
 - Importancia del diseño modular y reutilización de diseños.
 - Alternativas y evaluación de costes

Bibliografía

- Básica:
 - [NELS96] Capítulos 4 y 5
 - [GARC92] Capítulo 7
 - [WAKE99] Capítulos 5 y 6
 - [ALMO94] Capítulos 1, 2 y 3
- Complementaria
 - [FLOYD00] Capítulo 6
 - [HAY96] Capítulo 4
 - [MANO98] Capítulos 3 y 6
 - [MAND02] Dispositivos programables. HDL.

Contenidos

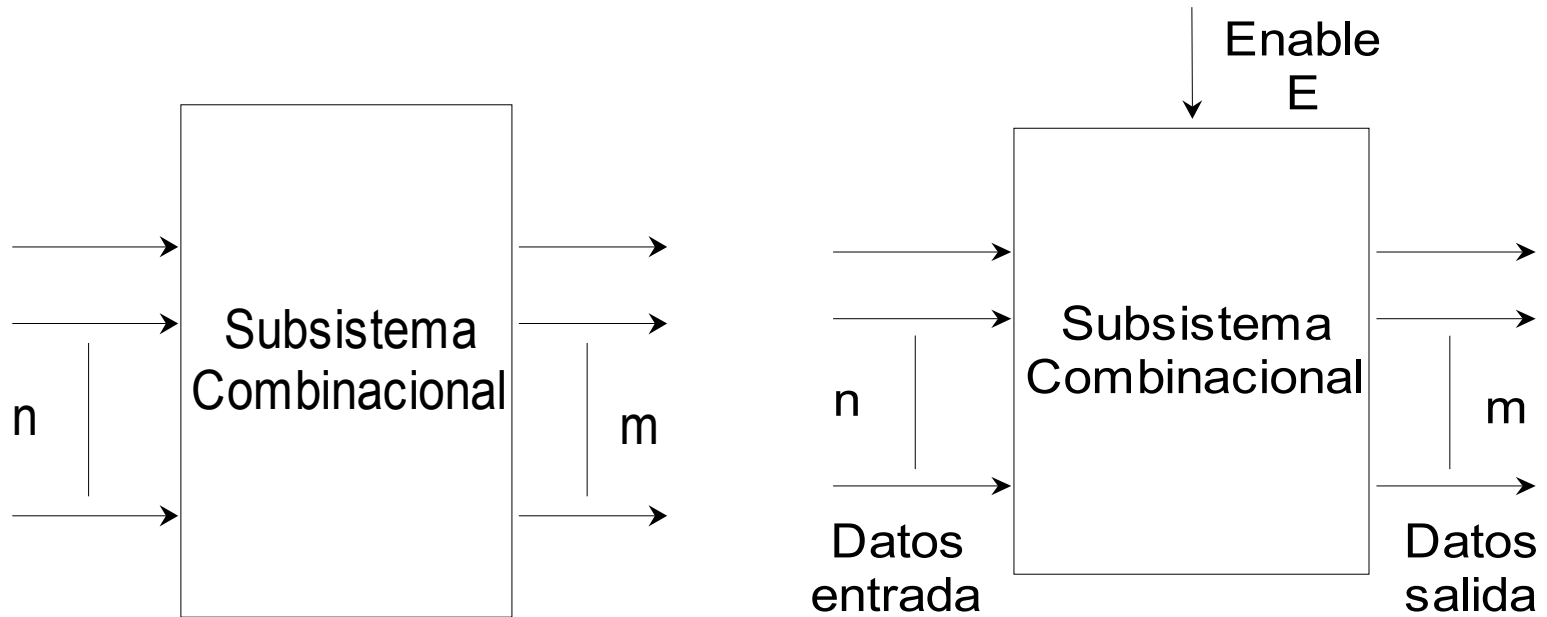
- Generalidades sobre subsistemas combinacionales MSI
- Subsistemas de propósito específico
 - Decodificadores
 - Codificadores
 - Convertidores de código
 - Demultiplexores
 - Comparadores de magnitud
- Subsistemas de propósito general
 - Multiplexores
 - Dispositivos lógicos programables

CIRCUITOS INTEGRADOS MSI/LSI

Estas líneas de entrada y salida pueden clasificarse en:

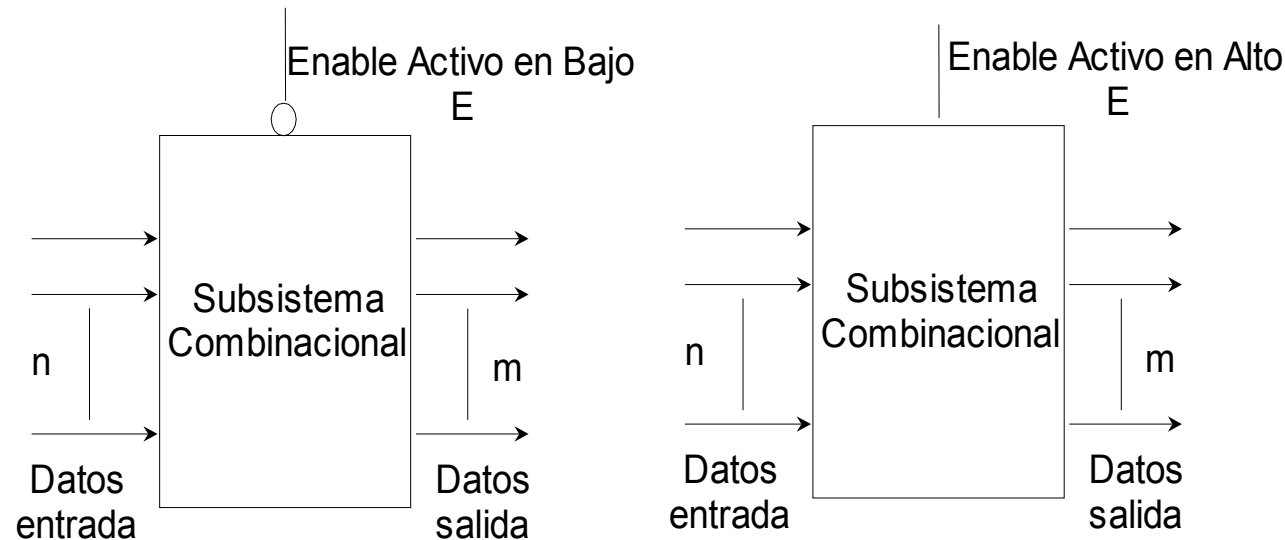
- Líneas de datos
- Líneas de control

Una de las líneas de control más usadas es el Enable (Habilitador).



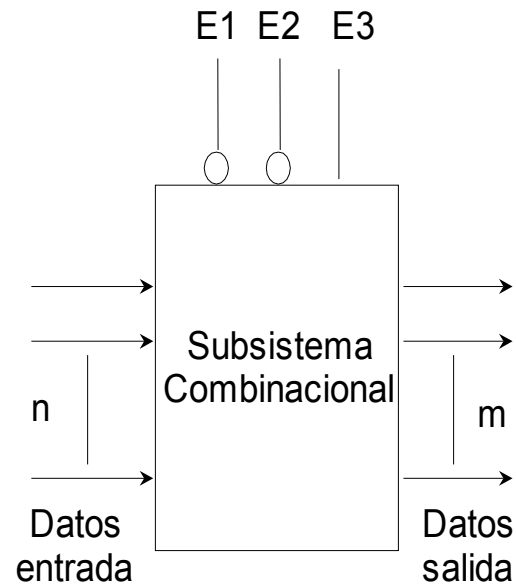
CIRCUITOS INTEGRADOS MSI/LSI

La habilitación de un subsistema por la línea de Enable, puede obtenerse cuando dicha línea de control tenga un nivel alto “1” o cuando tenga un nivel bajo “0”. En el primer caso, se dice que el Enable es activo en alta, mientras que para el segundo caso, el Enable es activo en baja.



CIRCUITOS INTEGRADOS MSI/LSI

Podemos encontrar subsistemas que tengan múltiples entradas de habilitación (enables).

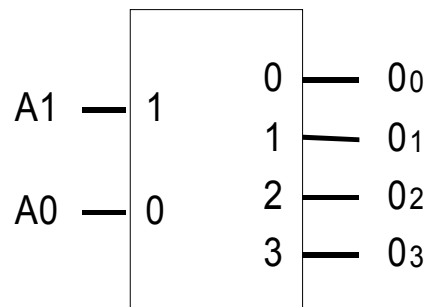


SUBSISTEMAS COMBINACIONES DE PROPÓSITO ESPECÍFICO

Decodificadores

El decodificador es un circuito integrado de n entradas de dato y m salidas de datos, donde $m \leq 2^n$. Cuando se cumple la igualdad, se dice que el decodificador es completo. Se especifican haciendo de la siguiente manera: *DEC $n:m$* o *DEC de n a m* .

Se caracterizan porque sólo se activa una salida por cada combinación de entrada



Entradas		Salidas			
A ₁	A ₀	O ₀	O ₁	O ₂	O 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

$$O_0 = A_1' \bullet A_0'$$

$$O_1 = A_1' \bullet A_0$$

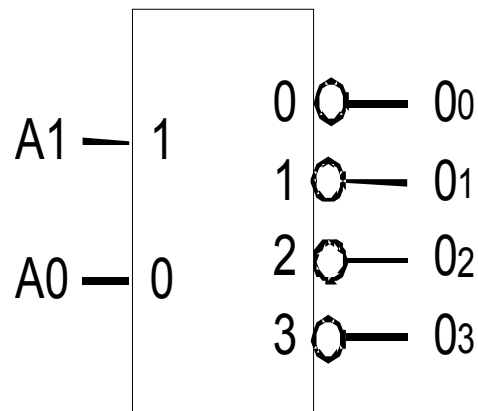
$$O_2 = A_1 \bullet A_0'$$

$$O_3 = A_1 \bullet A_0$$

SUBSISTEMAS COMBINACIONES DE PROPÓSITO ESPECÍFICO

Decodificadores

En el caso de que las salidas del decodificador de 2 a 4 fueran activas en bajo.



Entradas		Salidas			
A_1	A_0	O_0	O_1	O_2	O_3
0	0	0	1	1	1
0	1	1	0	1	1
1	0	1	1	0	1
1	1	1	1	1	0

$$O_0 = A_1 + A_0$$

$$O_1 = A_1 + A_0'$$

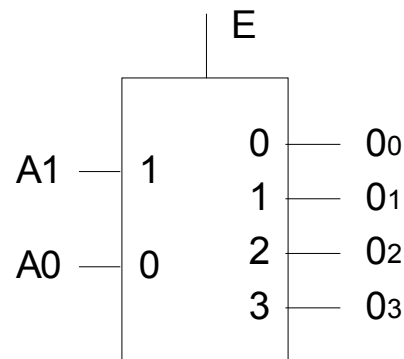
$$O_2 = A_1' + A_0$$

$$O_3 = A_1' + A_0'$$

SUBSISTEMAS COMBINACIONES DE PROPÓSITO ESPECÍFICO

Decodificadores

Señales de habilitación en los decodificadores



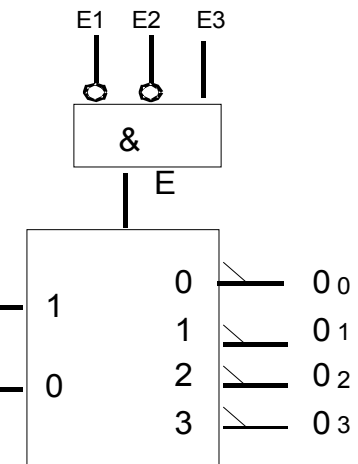
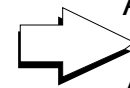
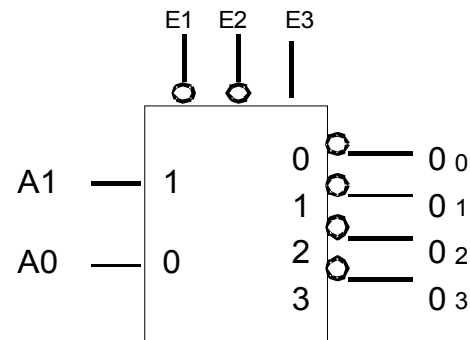
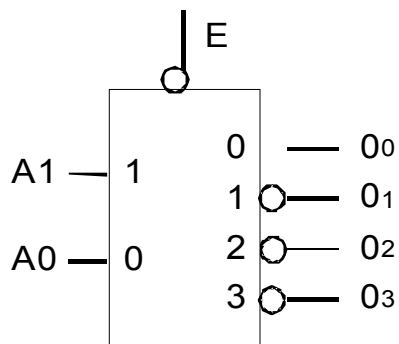
Entradas			Salidas			
E	A ₁	A ₀	O ₀	O ₁	O ₂	O ₃
0	X	X	0	0	0	0
1	0	0	1	0	0	0
1	0	1	0	1	0	0
1	1	0	0	0	1	0
1	1	1	0	0	0	1

$$O_3 = A_1 \bullet A_0 \bullet E$$

$$O_0 = A_1' \bullet A_0' \bullet E$$

$$O_1 = A_1' \bullet A_0 \bullet E$$

$$O_2 = A_1 \bullet A_0' \bullet E$$



SUBSISTEMAS COMBINACIONES DE PROPÓSITO ESPECÍFICO

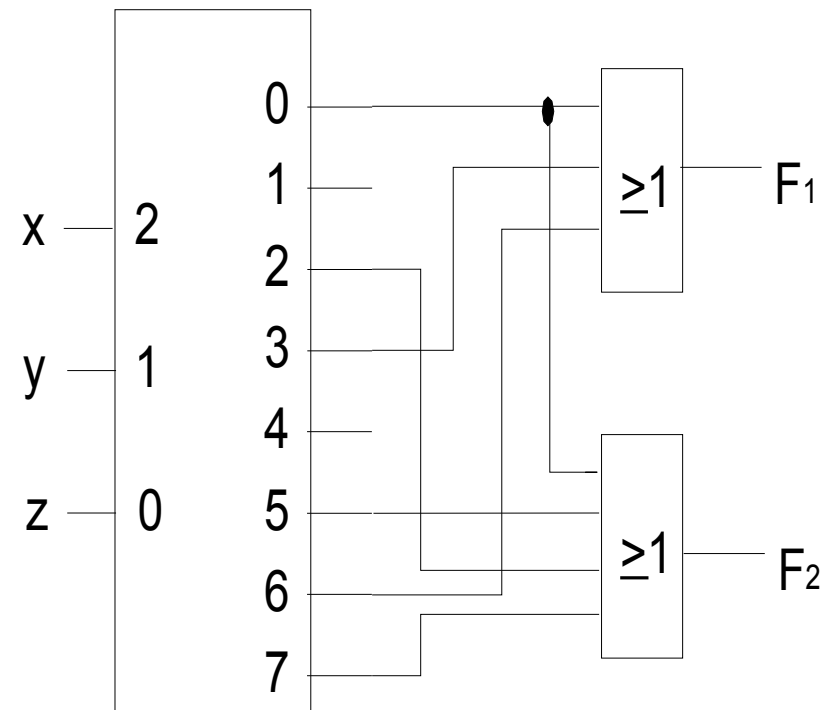
Decodificadores como generadores de funciones

Ejemplo: Implementar con un decodificador las siguientes funciones de conmutación

$$F_1 = \Sigma(0,3,6)$$

$$F_2 = \Pi(1,3,4,6)$$

Solución 1.- El decodificador tiene las salidas activas en alto.



SUBSISTEMAS COMBINACIONES DE PROPÓSITO ESPECÍFICO

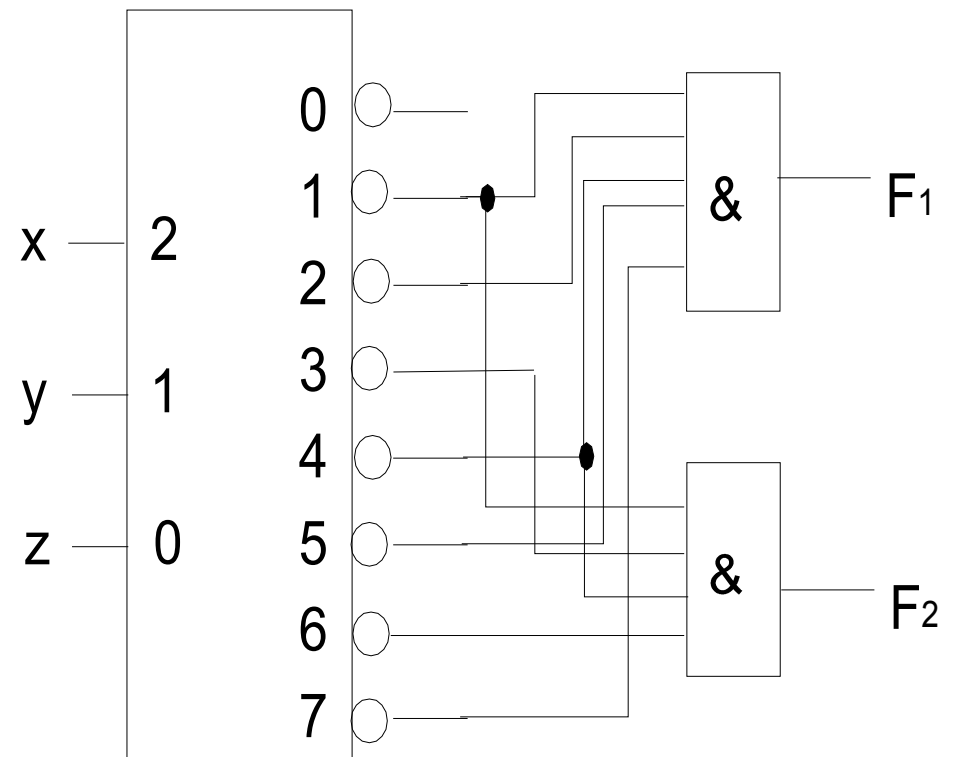
Decodificadores como generadores de funciones

Ejemplo: Implementar con un decodificador las siguientes funciones de conmutación

$$F_1 = \Sigma(0,3,6)$$

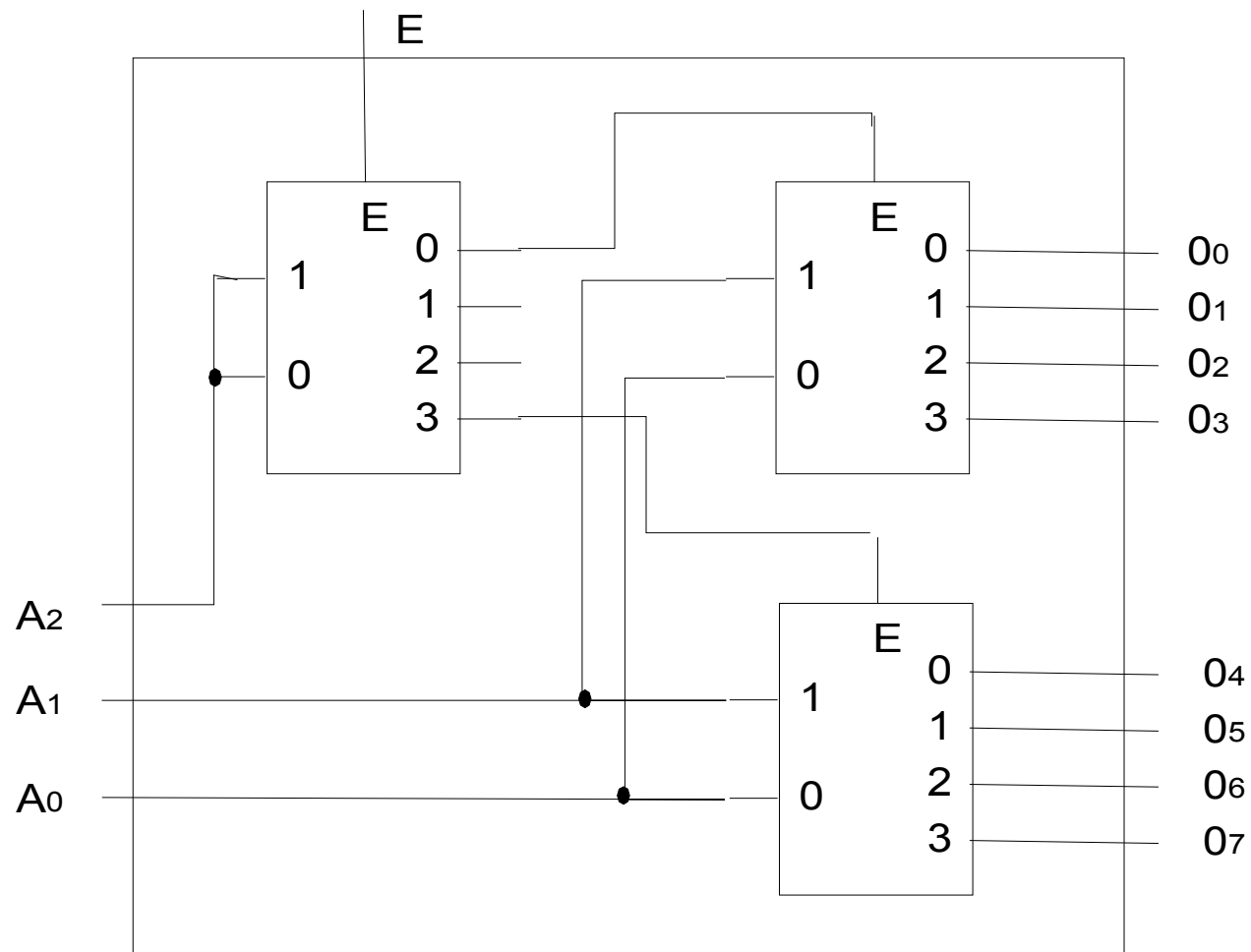
$$F_2 = \Pi(1,3,4,6)$$

Solución 2.- El decodificador tiene las salidas activas en bajo.



SUBSISTEMAS COMBINACIONES DE PROPÓSITO ESPECÍFICO

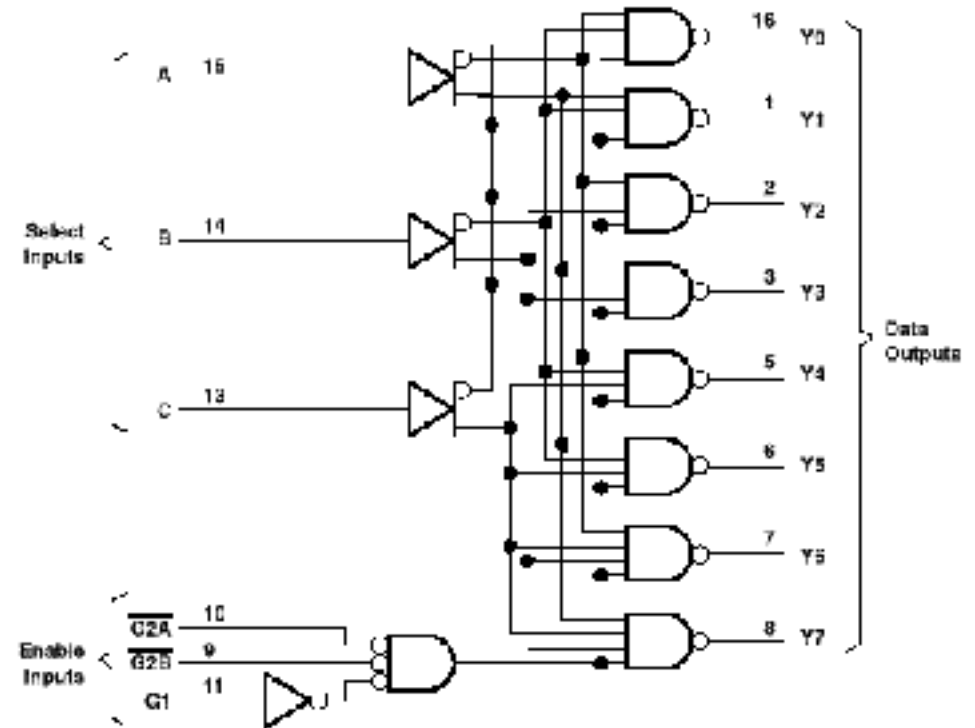
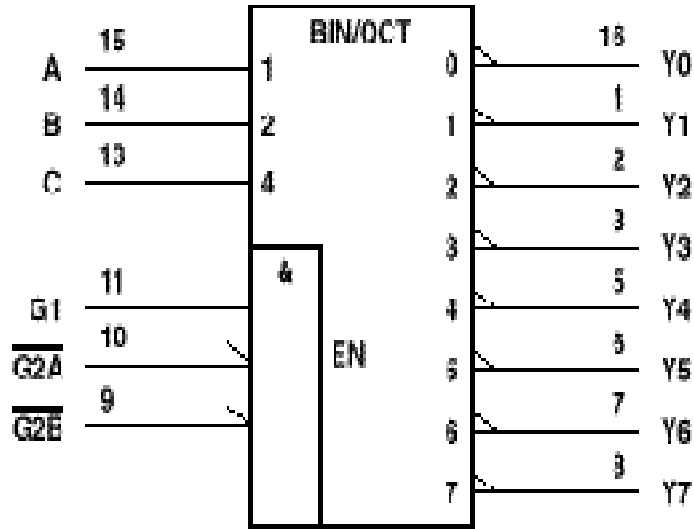
Asociación de Decodificadores



SUBSISTEMAS COMBINACIONES DE PROPÓSITO ESPECÍFICO

Decodificador Comercial

logic symbols (alternatives)[†]



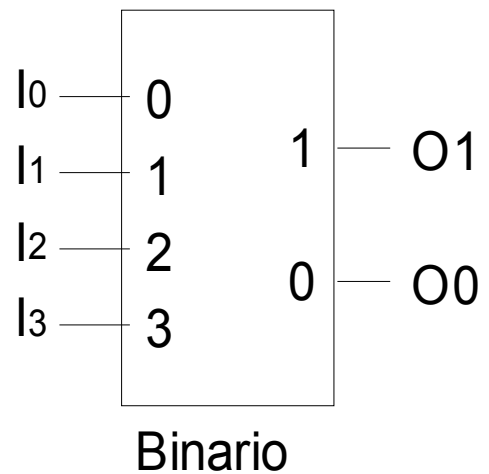
FUNCTION TABLE

[illegible]

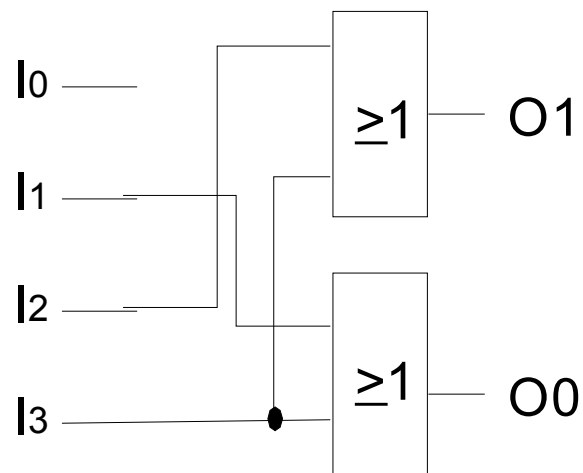
SUBSISTEMAS COMBINACIONES DE PROPÓSITO ESPECÍFICO

Codificadores

Codificador binario

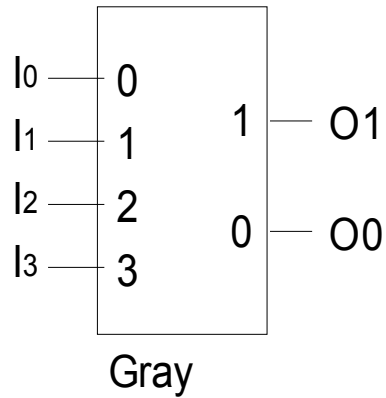


Entradas				Salidas	
I_0	I_1	I_2	I_3	O_1	O_0
1	0	0	0	0	0
0	1	0	0	0	1
0	0	1	0	1	0
0	0	0	1	1	1



SUBSISTEMAS COMBINACIONES DE PROPÓSITO ESPECÍFICO

Codificadores



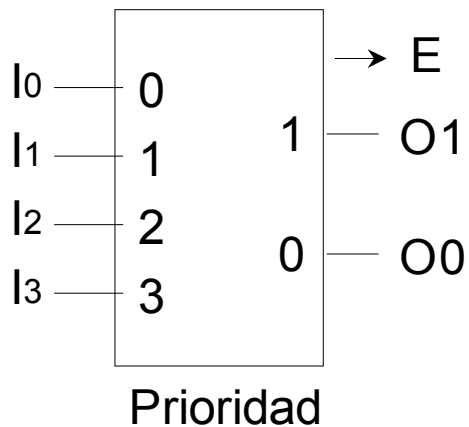
Entradas				Salidas	
I_0	I_1	I_2	I_3	O_1	O_0
1	0	0	0	0	0
0	1	0	0	0	1
0	0	1	0	1	1
0	0	0	1	1	0

$$O_0 = I_1 + I_2$$

$$O_1 = I_2 + I_3$$

SUBSISTEMAS COMBINACIONES DE PROPÓSITO ESPECÍFICO

Codificadores de prioridad



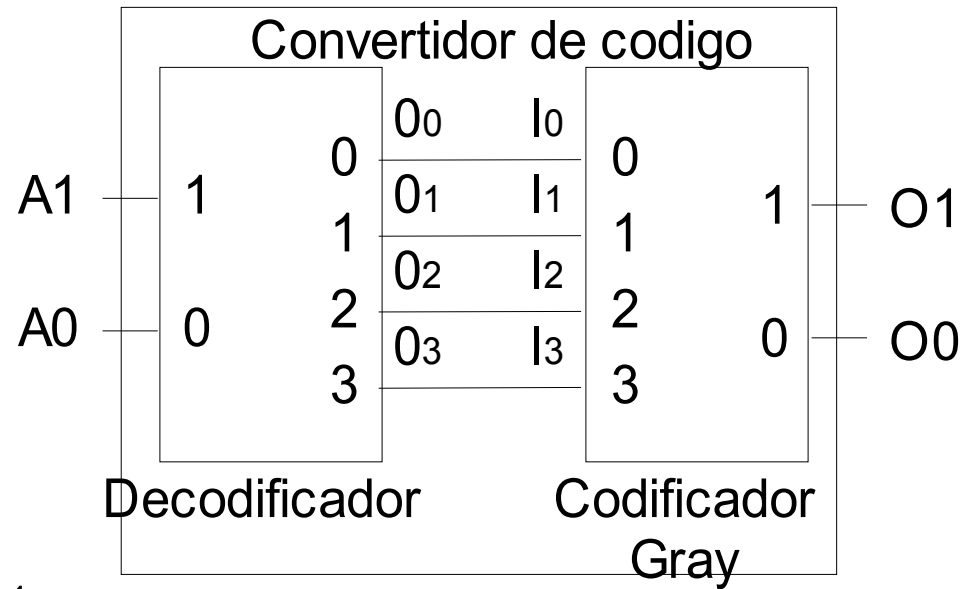
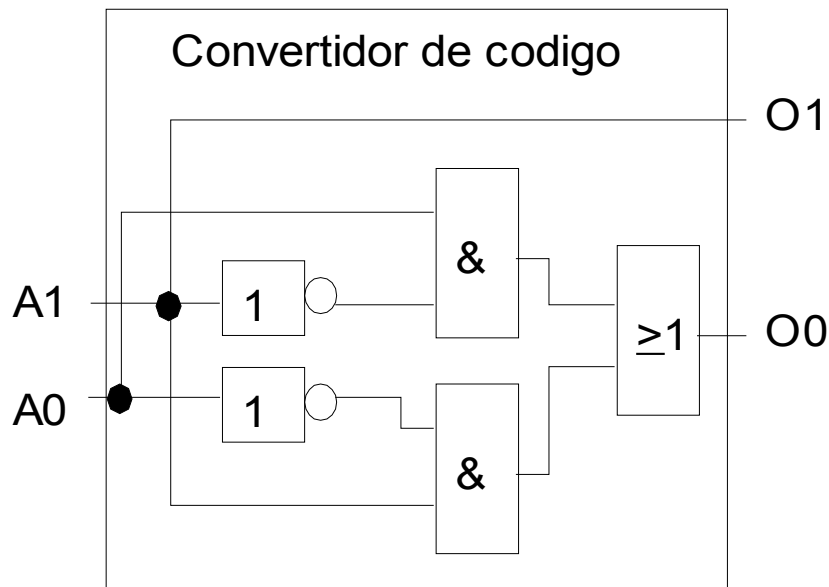
Entradas				Salidas		
I_3	I_2	I_1	I_0	E	O_1	O_0
1	x	x	X	0	1	1
0	1	x	x	0	1	0
0	0	1	x	0	0	1
0	0	0	1	0	0	0
0	0	0	0	1	0	0

[illegible]

SUBSISTEMAS COMBINACIONES DE PROPÓSITO ESPECÍFICO

Convertidor de código

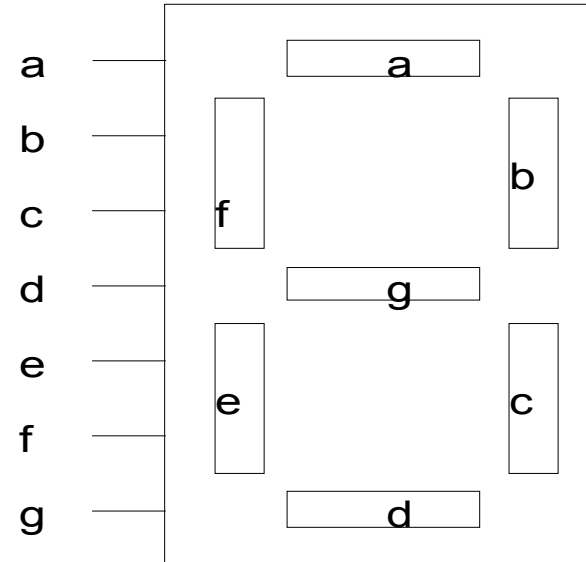
BINARIO/GRAY
GRAY/BINARIO
BCD/GRAY
BCD/7-SEGMENTOS



SUBSISTEMAS COMBINACIONES DE PROPÓSITO ESPECÍFICO

Convertidor de código

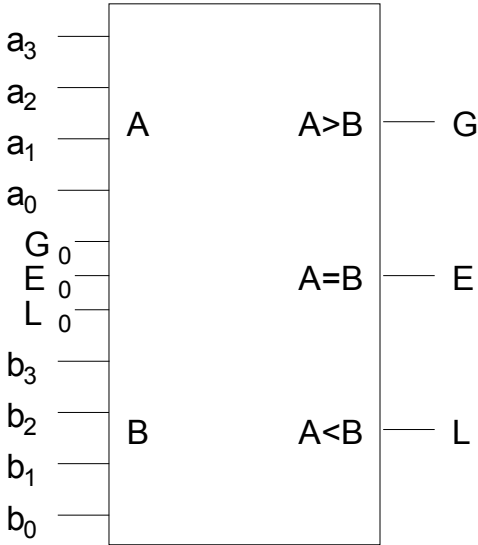
Convertidor BCD/7 segmentos



b_3	b_2	b_1	b_0	a	b	c	d	e	f	g
0	0	0	0	1	1	1	1	1	1	0
0	0	0	1	0	1	1	0	0	0	0
0	0	1	0	1	1	0	1	1	0	1
0	0	1	1	1	1	1	1	0	0	1
0	1	0	0	0	1	1	0	0	1	1
0	1	0	1	1	0	1	1	0	1	1
0	1	1	0	0	0	1	1	1	1	1
0	1	1	1	1	1	1	0	0	0	0
1	0	0	0	1	1	1	1	1	1	1
1	0	0	1	1	1	1	0	0	1	1

SUBSISTEMAS COMBINACIONES DE PROPÓSITO ESPECÍFICO

Comparadores de magnitud

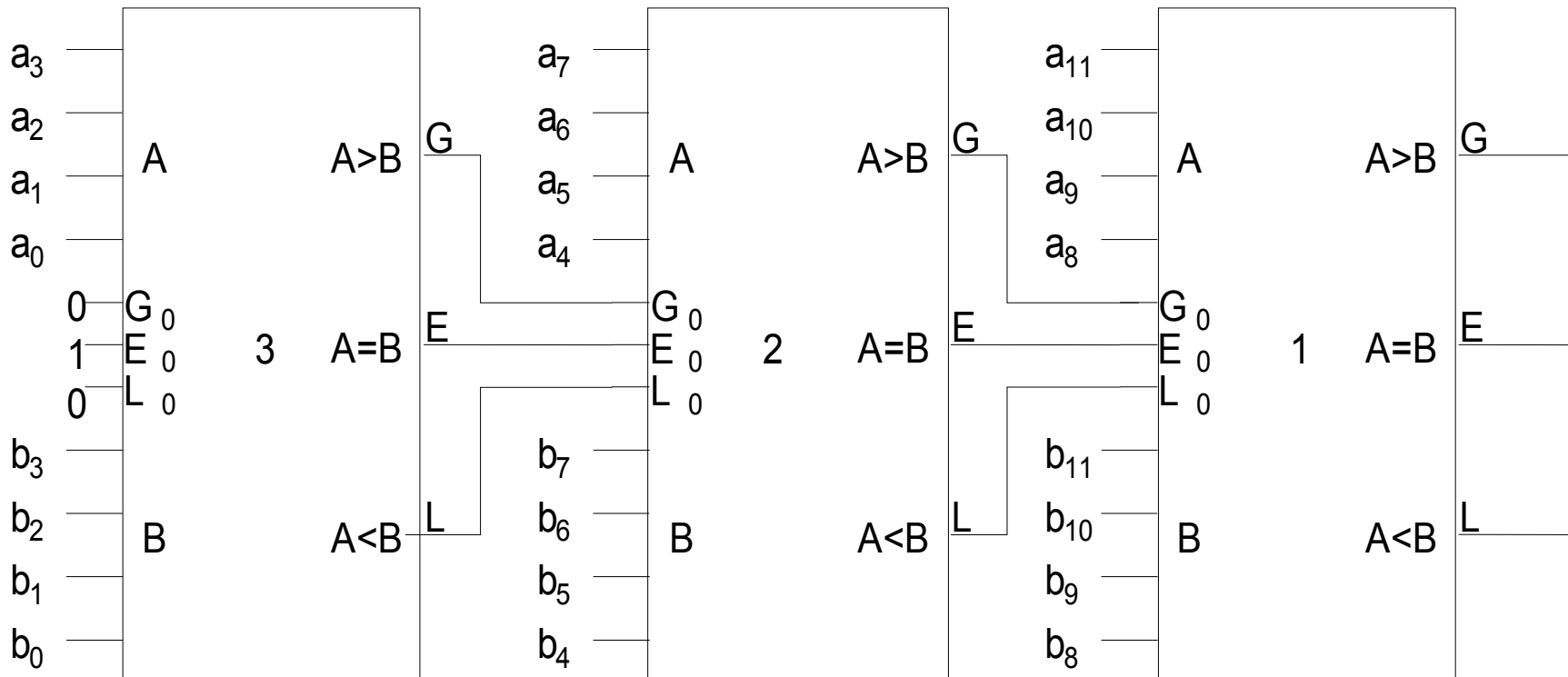


FUNCTION TABLE

COMPARING INPUTS				CASCADING INPUTS			OUTPUTS		
A3 B3	A2 B2	A1 B1	A0 B0	A > B	A < B	A = B	A > B	A < B	A = B
A3 > B3	X	X	X	X	X	X	H	L	L
A3 < B3	X	X	X	X	X	X	L	H	L
A3 = B3	A2 > B2	X	X	X	X	X	H	L	L
A3 = B3	A2 < B2	X	X	X	X	X	L	H	L
A3 = B3	A2 = B2	A1 > B1	X	X	X	X	H	L	L
A3 = B3	A2 = B2	A1 < B1	X	X	X	X	L	H	L
A3 = B3	A2 = B2	A1 = B1	A0 > B0	X	X	X	H	L	L
A3 = B3	A2 = B2	A1 = B1	A0 < B0	X	X	X	L	H	L
A3 = B3	A2 = B2	A1 = B1	A0 = B0	H	L	L	H	L	L
A3 = B3	A2 = B2	A1 = B1	A0 = B0	L	H	L	L	H	L
A3 = B3	A2 = B2	A1 = B1	A0 = B0	X	X	L	L	L	H
A3 = B3	A2 = B2	A1 = B1	A0 = B0	H	H	L	L	L	L
A3 = B3	A2 = B2	A1 = B1	A0 = B0	L	L	L	H	H	L

SUBSISTEMAS COMBINACIONES DE PROPÓSITO ESPECÍFICO

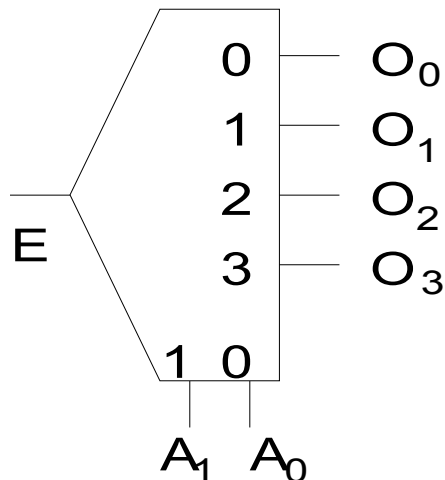
Asociación de comparadores



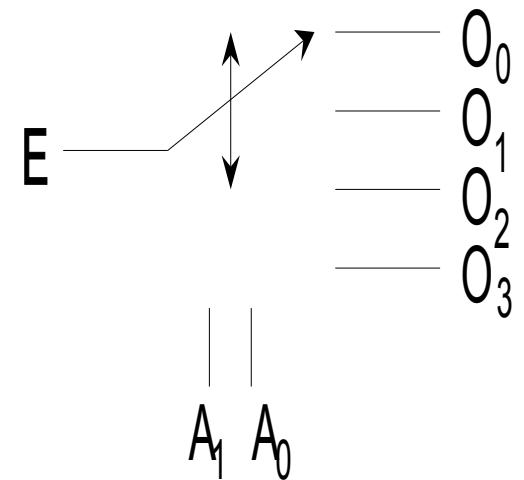
SUBSISTEMAS COMBINACIONES DE PROPÓSITO ESPECÍFICO

Demultiplexor

Es un circuito MSI que dispone de n líneas de control y $2^n + 1$ líneas de datos. 2^n líneas de datos son de salida (también llamadas canales) frente a una única línea de entrada. Se pueden designar mediante DEMUX 1:m, donde m es el número de canales.

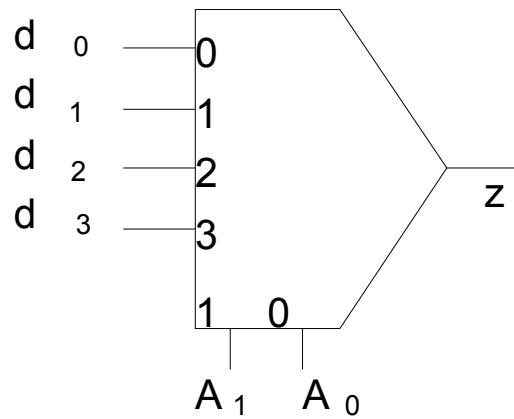


Entradas			Salidas			
E	A_1	A_0	O_0	O_1	O_2	O_3
0	X	X	0	0	0	0
1	0	0	1	0	0	0
1	0	1	0	1	0	0
1	1	0	0	0	1	0
1	1	1	0	0	0	1



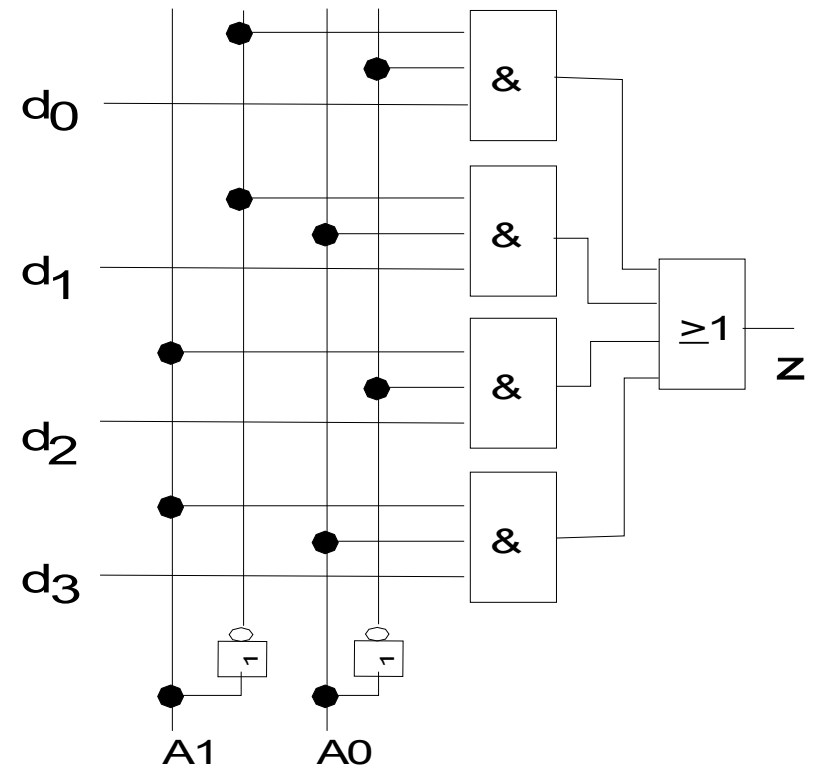
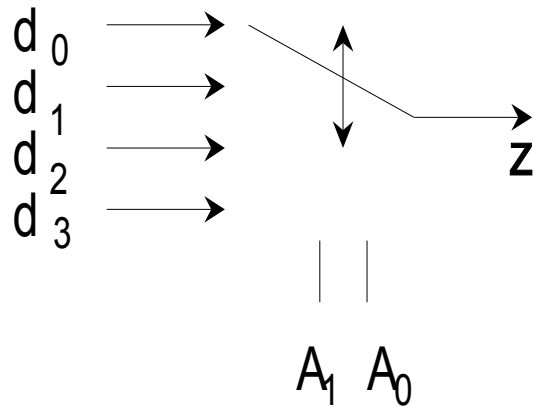
SUBSISTEMAS COMBINACIONES DE PROPÓSITO GENERAL

Multiplexores



A_1	A_0	Z
0	0	d_0
0	1	d_1
1	0	d_2
1	1	d_3

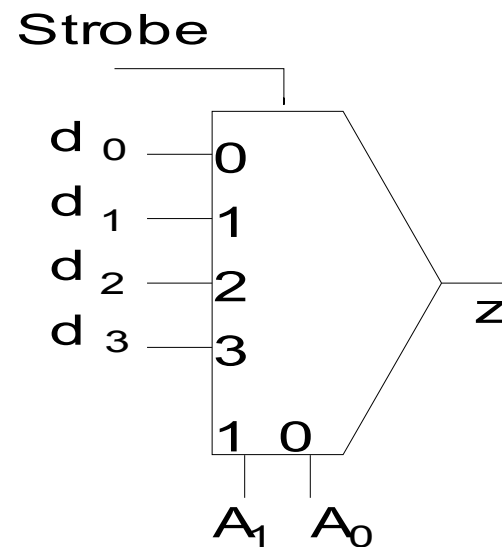
$$Z = d_0 A_1' A_0' + d_1 A_1' A_0 + d_2 A_1 A_0' + d_3 A_1 A_0$$



SUBSISTEMAS COMBINACIONES DE PROPÓSITO GENERAL

Multiplexores

Los multiplexores pueden tener señales de habilitación



SUBSISTEMAS COMBINACIONES DE PROPÓSITO GENERAL

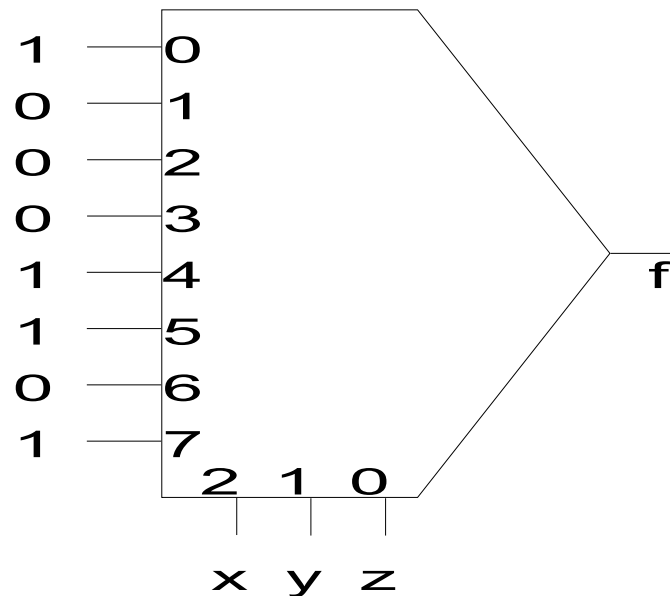
Multiplexores como generadores de funciones

El teorema de expansión de Shannon estudiado en el tema 3 indica que cualquier función de conmutación completa de n variables puede ser expresada como

$$F(x_1, \dots, x_n) = \sum f(i) \cdot m_i(x_1, \dots, x_n)$$

$$Z = \sum d_i \cdot m_i(x_1, \dots, x_n)$$

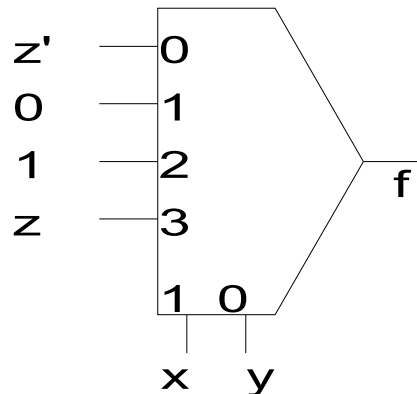
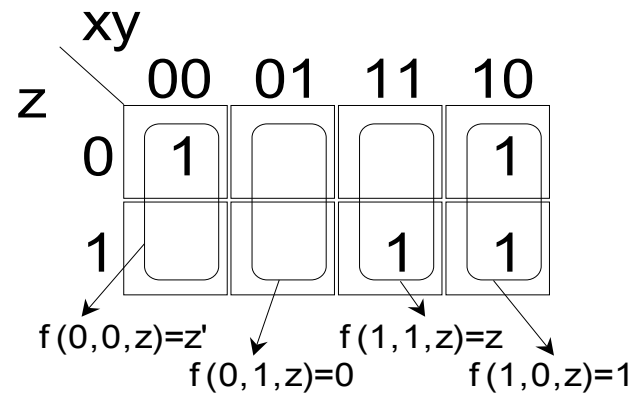
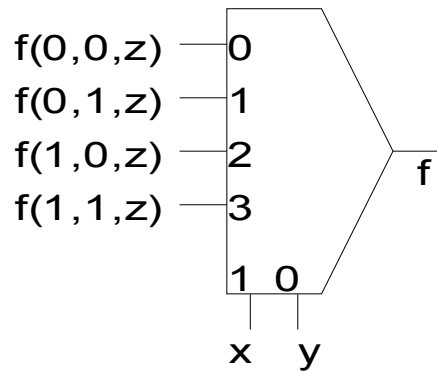
Ejemplo1 : Usando un multiplexor de 8 canales, implementar la función de conmutación $f = \sum(0, 4, 5, 7)$



SUBSISTEMAS COMBINACIONES DE PROPÓSITO GENERAL

Multiplexores como generadores de funciones

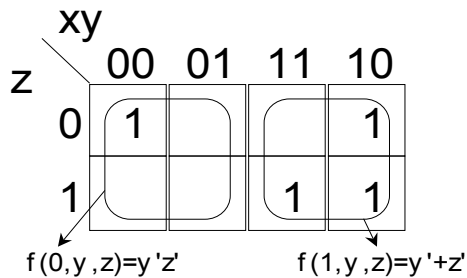
Ejemplo 2: Diseñar la función de conmutación $f = \Sigma(0,4,5,7)$ usando multiplexores de 4 canales.



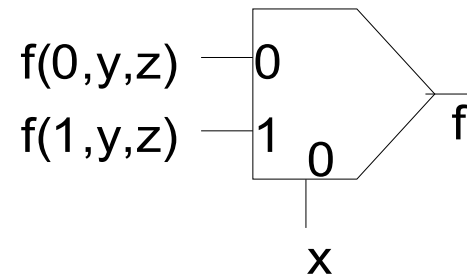
SUBSISTEMAS COMBINACIONES DE PROPÓSITO GENERAL

Multiplexores como generadores de funciones

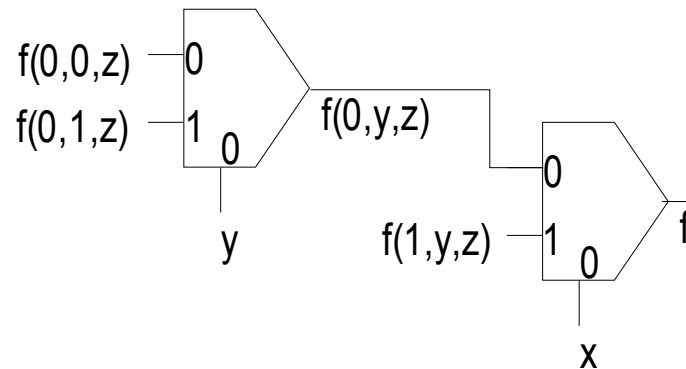
Ejemplo 3. Diseñar la función de conmutación $f = \Sigma(0,4,5,7)$ usando multiplexores de 2 canales.



$$F(x,y,z) = d_0 x' + d_1 x$$



$$F(1,y,z) = d_0 y' + d_1 y$$



SUBSISTEMAS COMBINACIONES DE PROPÓSITO GENERAL

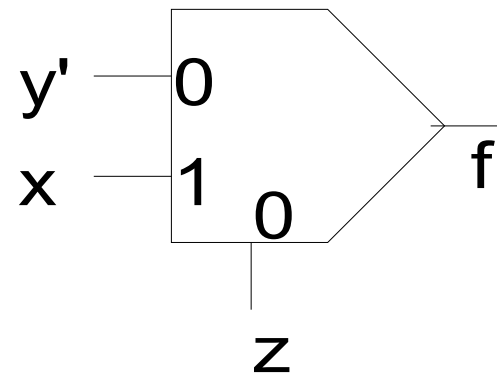
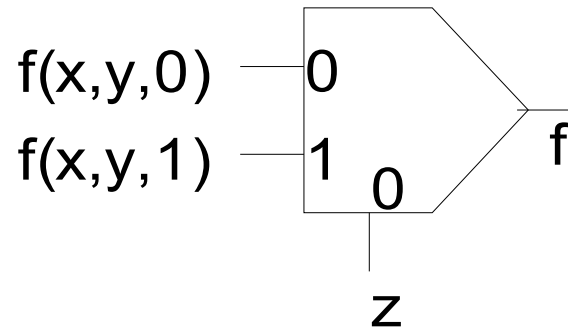
Multiplexores como generadores de funciones

Otra solución:

$$F(x,y,z) = d_0 z' + d_1 z$$

xy		00	01	11	10
z	0	1			1
	1			1	1

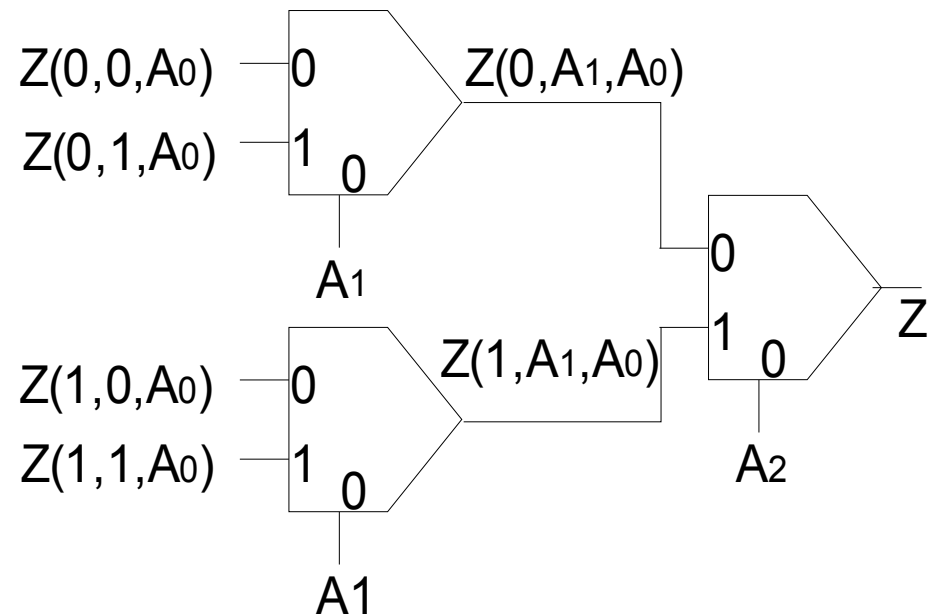
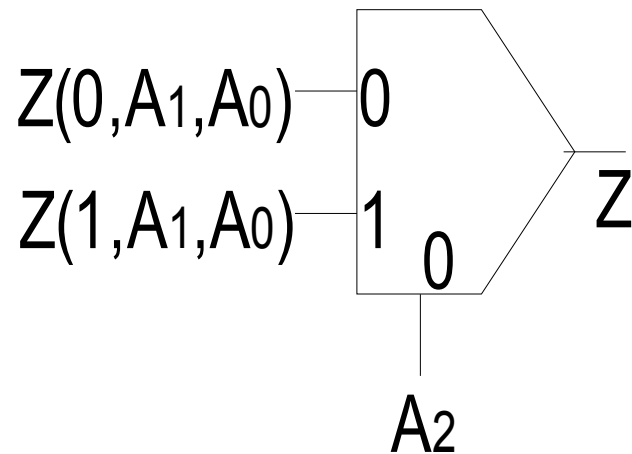
$f(x,y,0) = y'$ $f(x,y,1) = x$



SUBSISTEMAS COMBINACIONES DE PROPÓSITO GENERAL

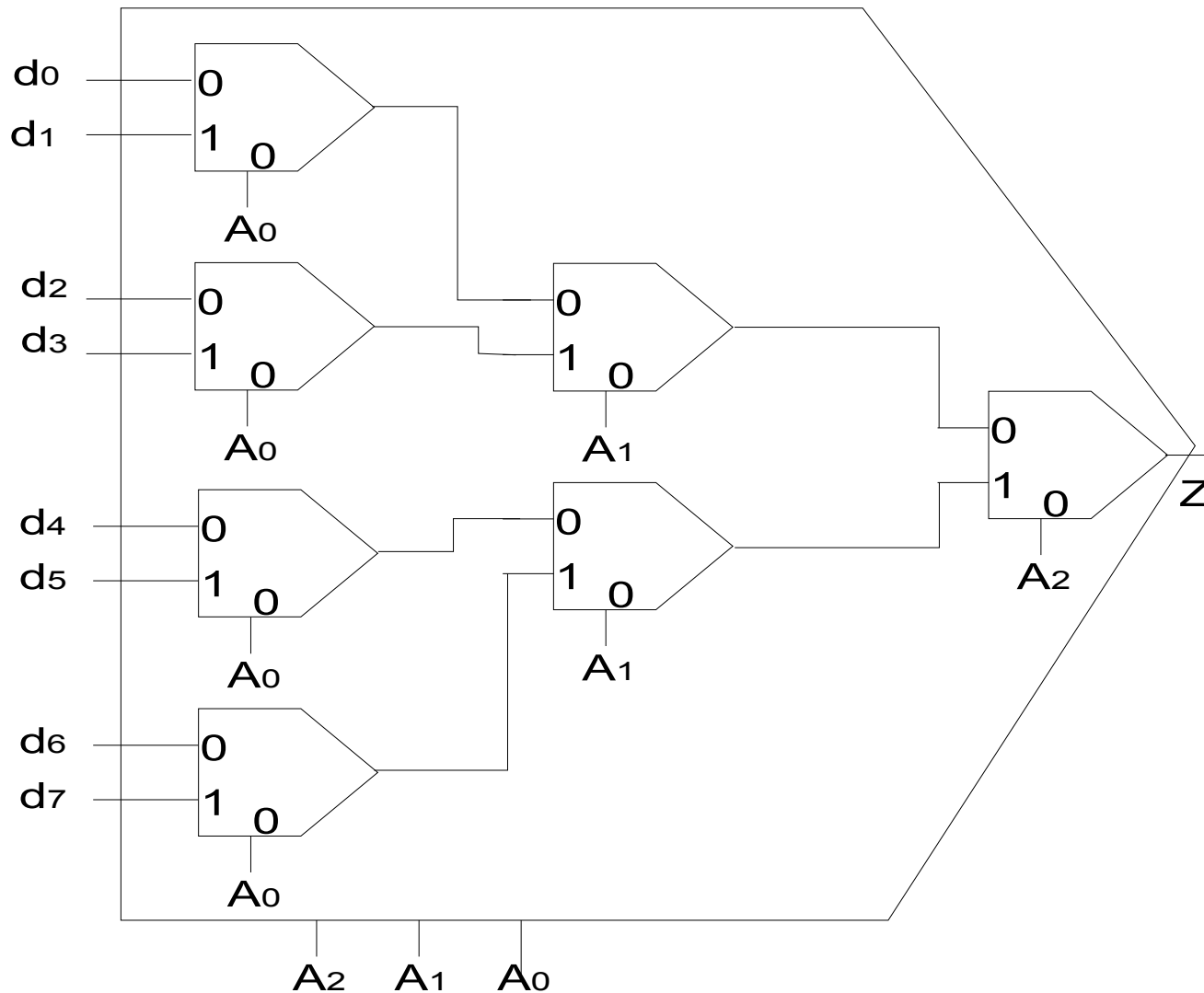
Asociación de multiplexores

$$Z(A_2A_1A_0) = d_0 A_2' A_1' A_0' + d_1 A_2' A_1' A_0 + d_2 A_2' A_1 A_0' + d_3 A_2' A_1 A_0 + d_4 A_2 A_1' A_0' + d_5 A_2 A_1' A_0 + d_6 A_2 A_1 A_0' + d_7 A_2 A_1 A_0$$



SUBSISTEMAS COMBINACIONES DE PROPÓSITO GENERAL

Asociación de multiplexores



SUBSISTEMAS COMBINACIONES DE PROPÓSITO GENERAL

Dispositivos lógicos programables

Plano AND fijo plano OR programable: ROM

Plano AND programable y OR fijo: PAL

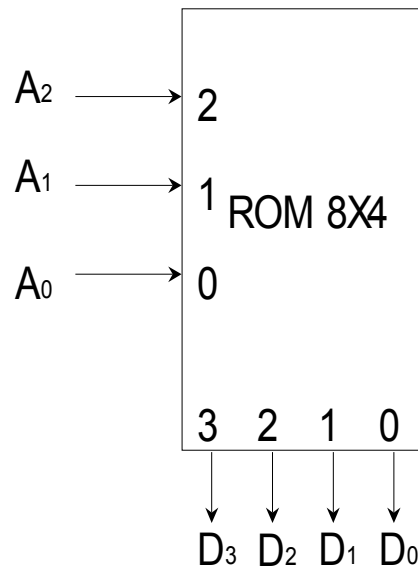
Planos AND y OR programables: PLA

SUBSISTEMAS COMBINACIONES DE PROPÓSITO GENERAL

Dispositivos lógicos programables

ROM

$$\text{Capacidad(bits)} = 2^n \times m$$



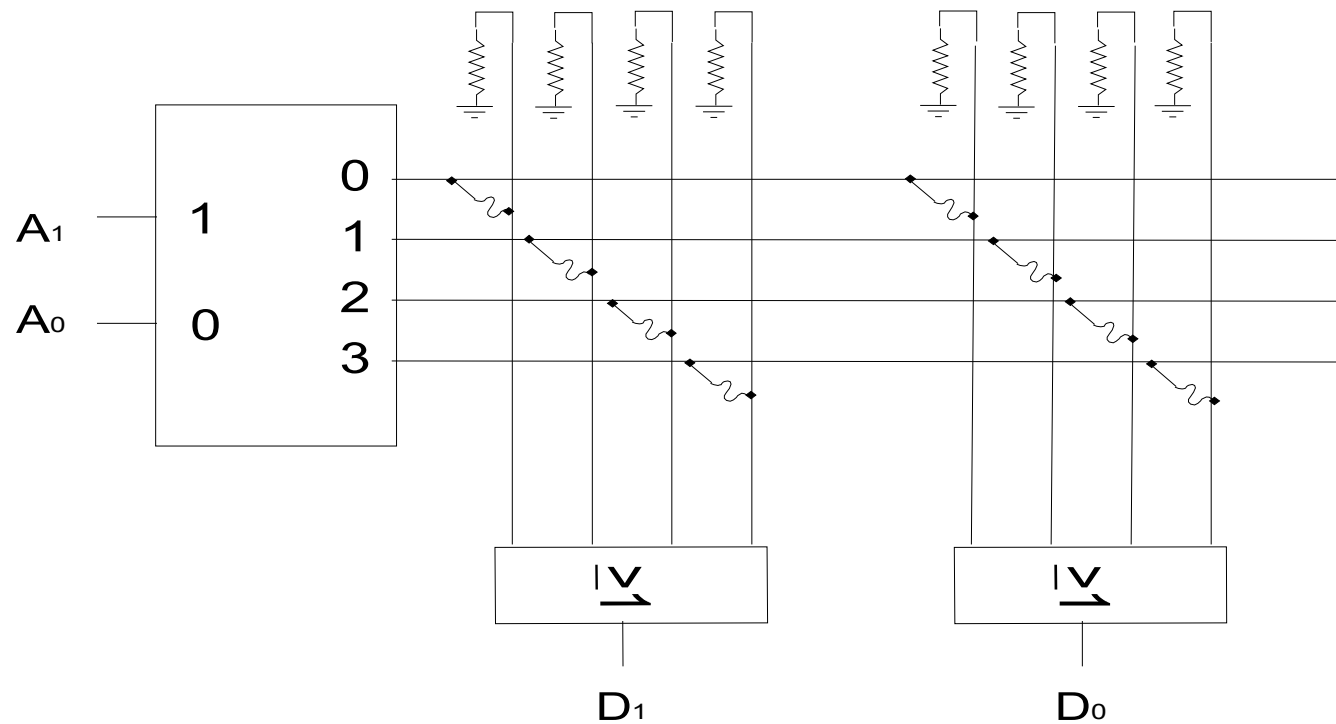
A_2	000	1	1	0	0
	001	0	0	0	1
	010	1	0	1	0
A_1	011	1	1	1	0
	100	0	0	1	1
A_0	101	0	0	0	0
	110	0	0	0	1
	111	1	0	0	0
		D_3	D_2	D_1	D_0

SUBSISTEMAS COMBINACIONES DE PROPÓSITO GENERAL

Dispositivos lógicos programables

ROM

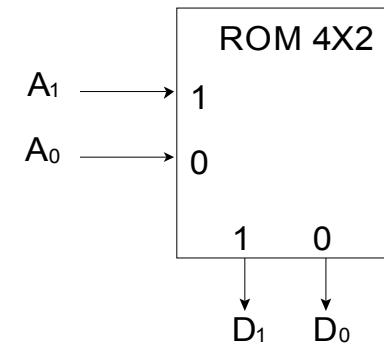
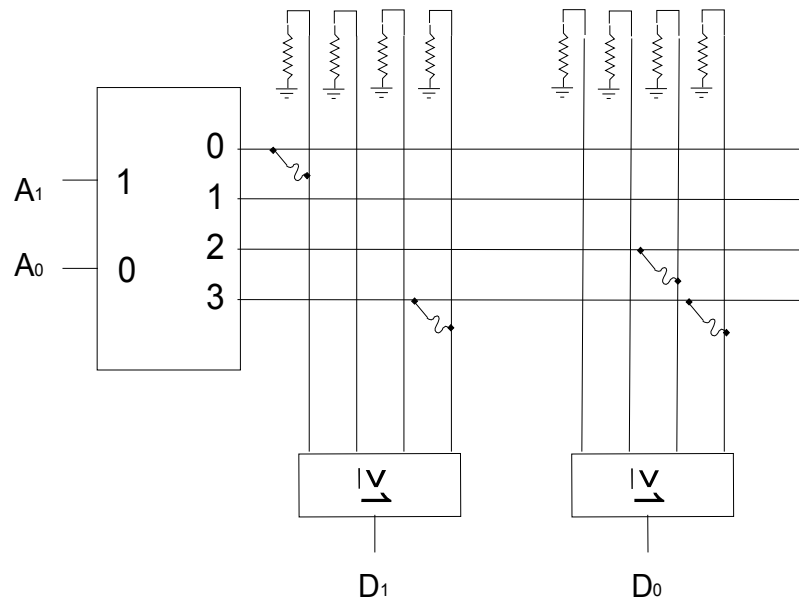
Una ROM de $2^n \times m$, está formada por un decodificador de m líneas de entrada, $2^n \times m$ fusibles o interconexiones y m puertas OR. La siguiente figura esquematiza una ROM de 4×2 .



SUBSISTEMAS COMBINACIONES DE PROPÓSITO GENERAL

Dispositivos lógicos programables

ROM



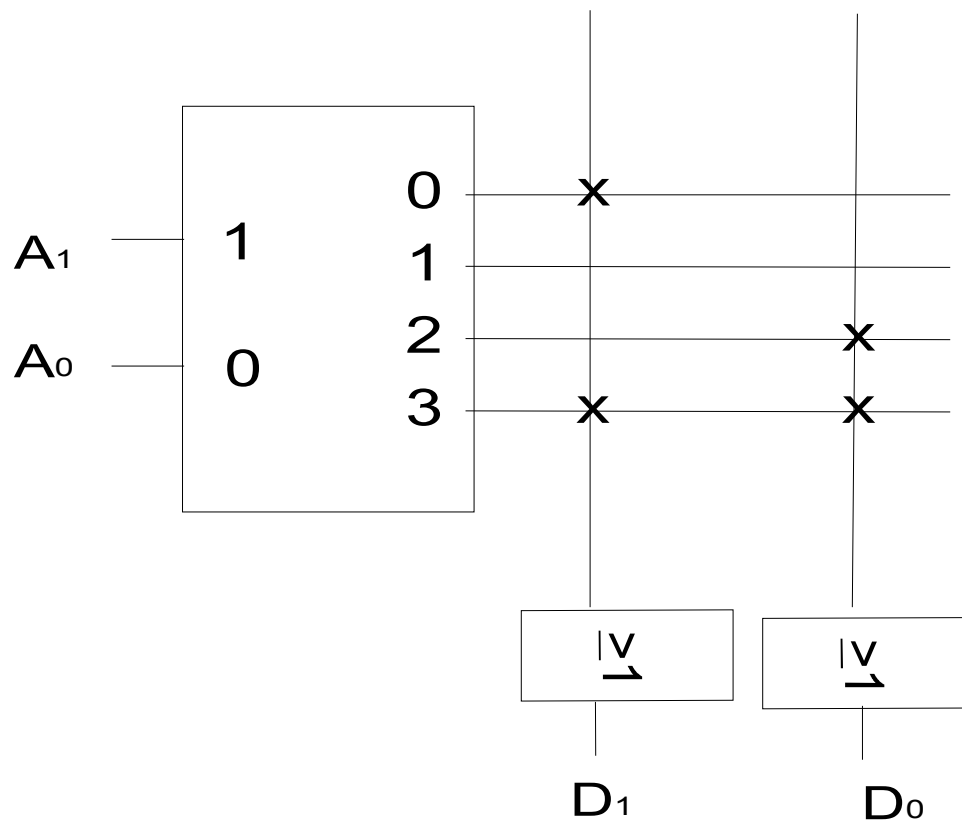
A ₁	00	1	0
A ₁	01	0	0
A ₀	10	0	1
A ₀	11	1	1
		D ₁	D ₀

SUBSISTEMAS COMBINACIONES DE PROPÓSITO GENERAL

Dispositivos lógicos programables

ROM

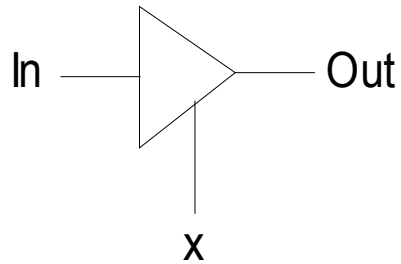
Representación simplificada



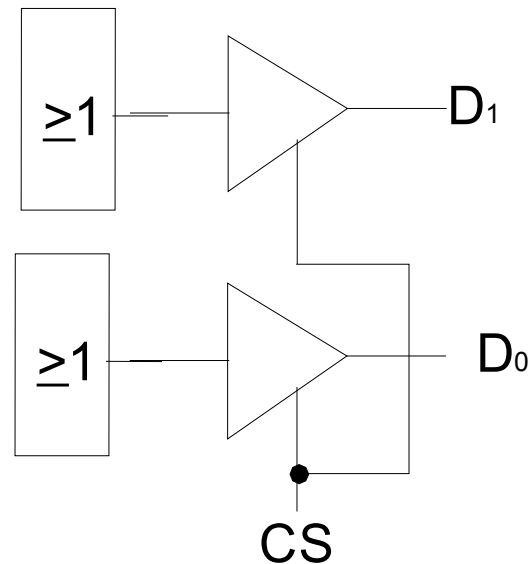
SUBSISTEMAS COMBINACIONES DE PROPÓSITO GENERAL

Dispositivos lógicos programables

ROM



X	Out
	=
0	HI
1	In

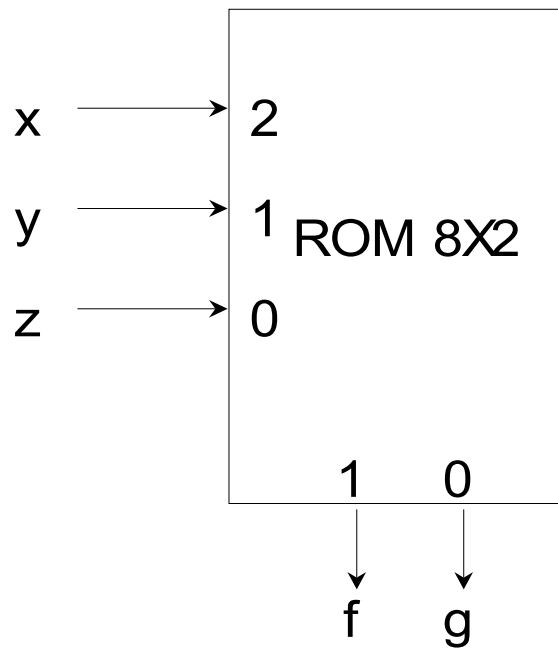


SUBSISTEMAS COMBINACIONES DE PROPÓSITO GENERAL

Dispositivos lógicos programables

La ROM como generador de funciones

Ejemplo 1: Implementar las funciones $f = \Sigma(1,3,5,7)$ y $g = \Sigma(0,2,4,5)$ haciendo uso de una ROM.



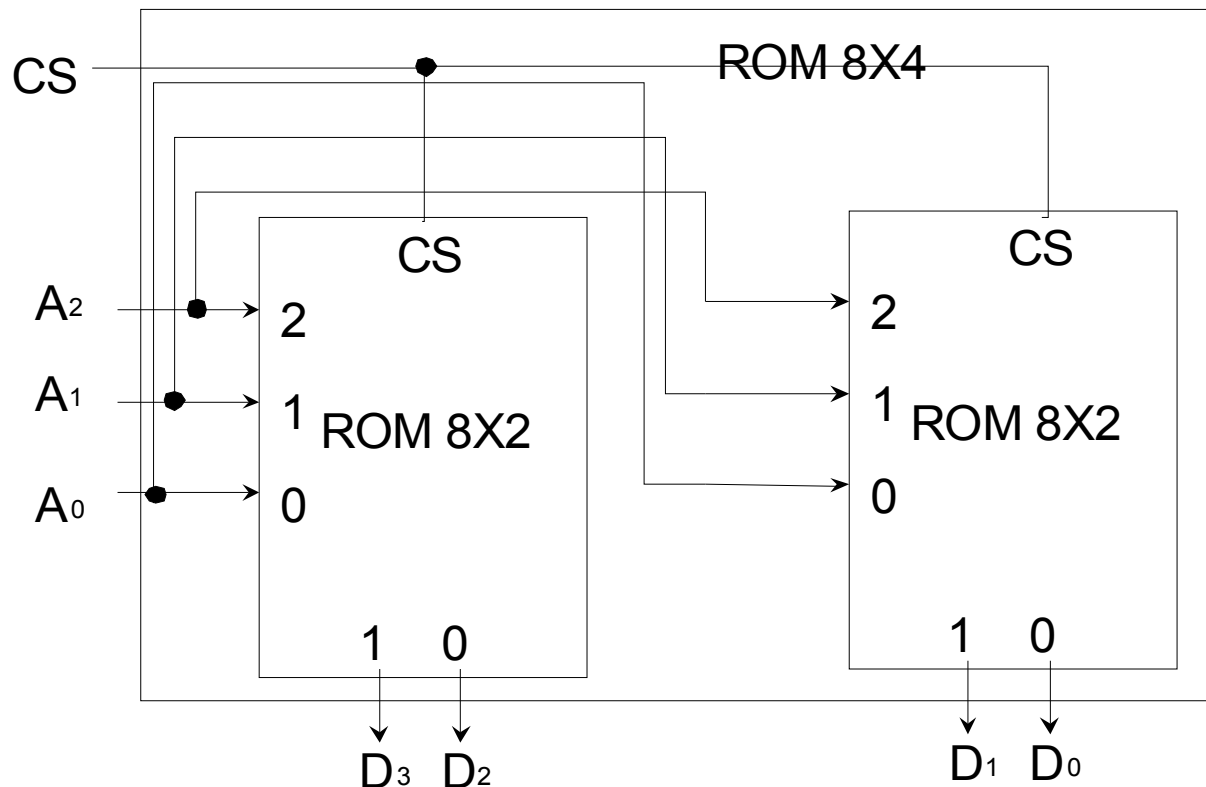
A_2	A_1	A_0	D_1	D_0
0	0	0	0	1
0	0	1	1	0
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	1
1	1	0	0	0
1	1	1	1	0

SUBSISTEMAS COMBINACIONES DE PROPÓSITO GENERAL

Dispositivos lógicos programables

Asociación de ROM

Ejemplo 1: Se dispone de ROM de 3X2 y se desea construir una ROM de 3x4.



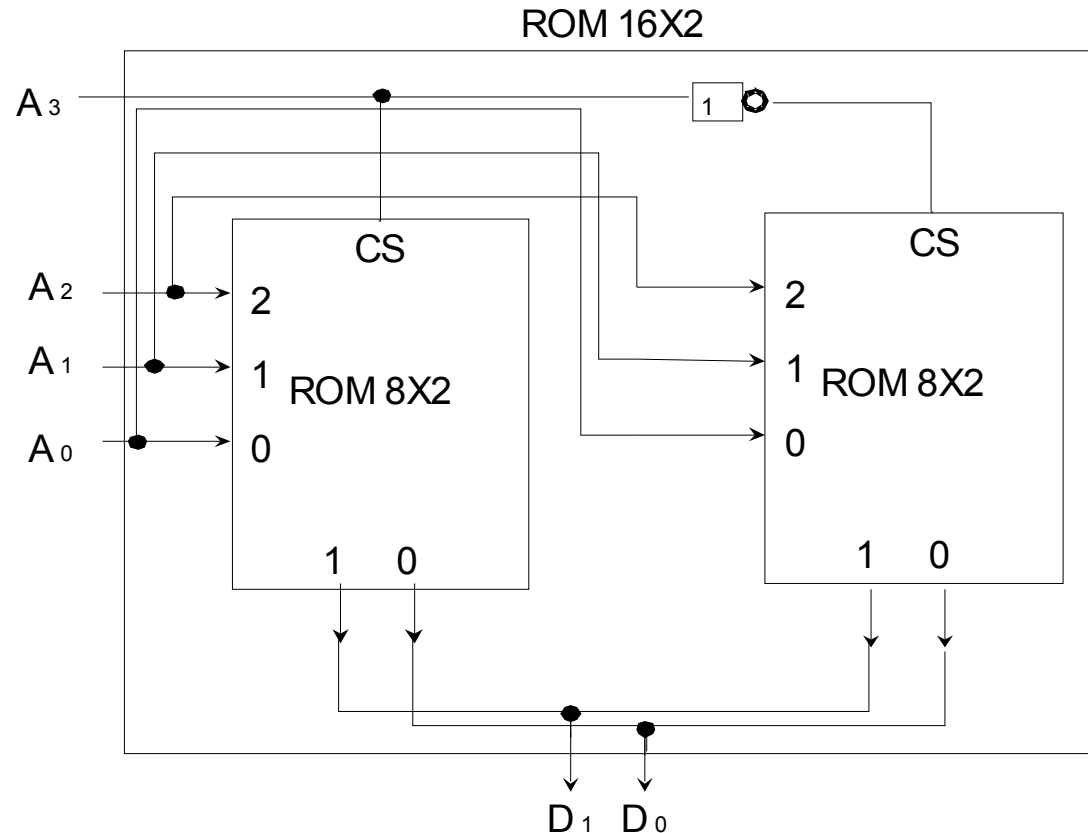
SUBSISTEMAS COMBINACIONES DE PROPÓSITO GENERAL

Dispositivos lógicos programables

Asociación de ROM

Ejemplo 2

Se dispone de ROMs de 8x2, y se desea construir una ROM de 16x2.

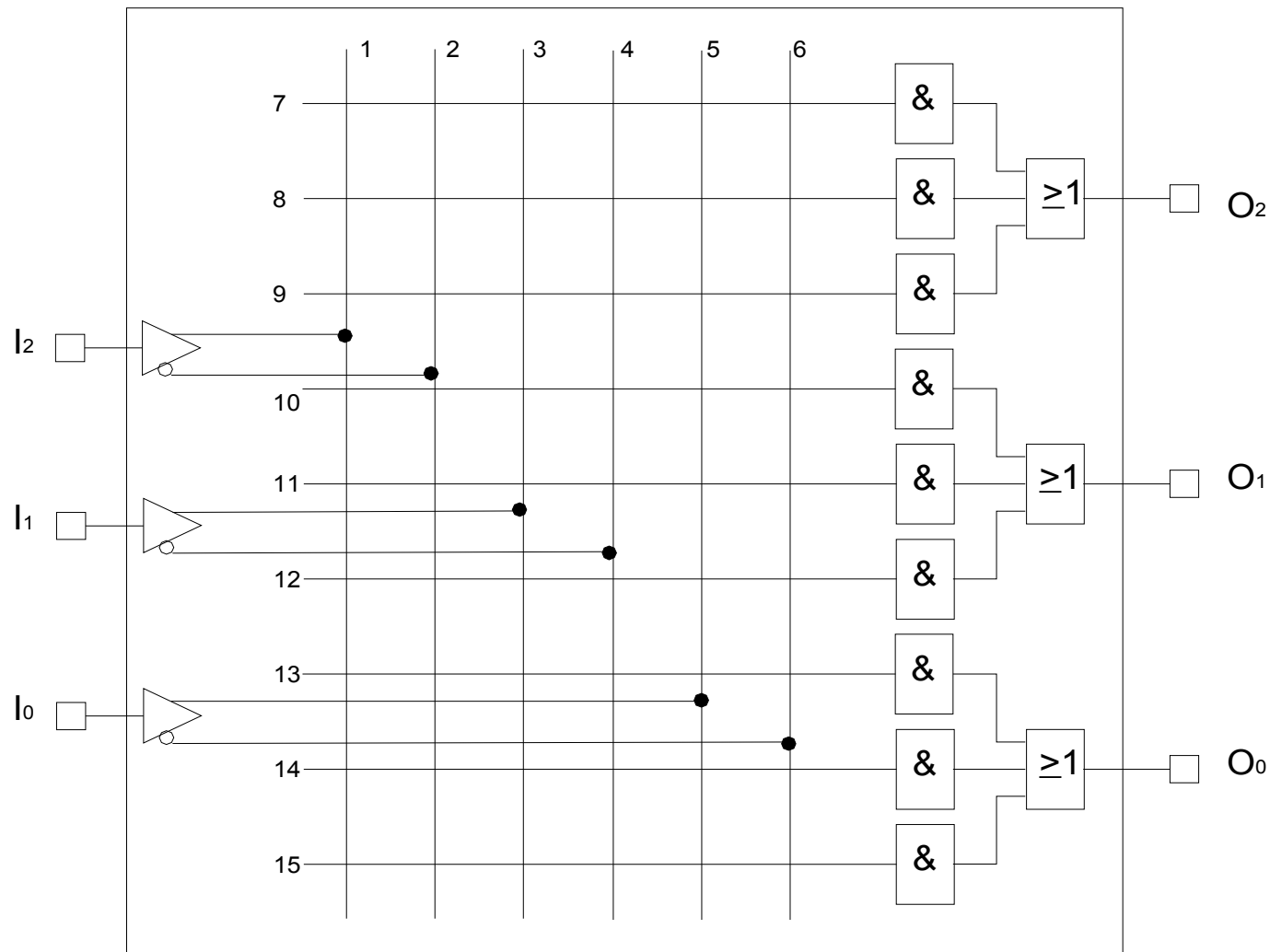


SUBSISTEMAS COMBINACIONES DE PROPÓSITO GENERAL

Dispositivos lógicos programables

PAL

Plano AND programable y plano OR fijo

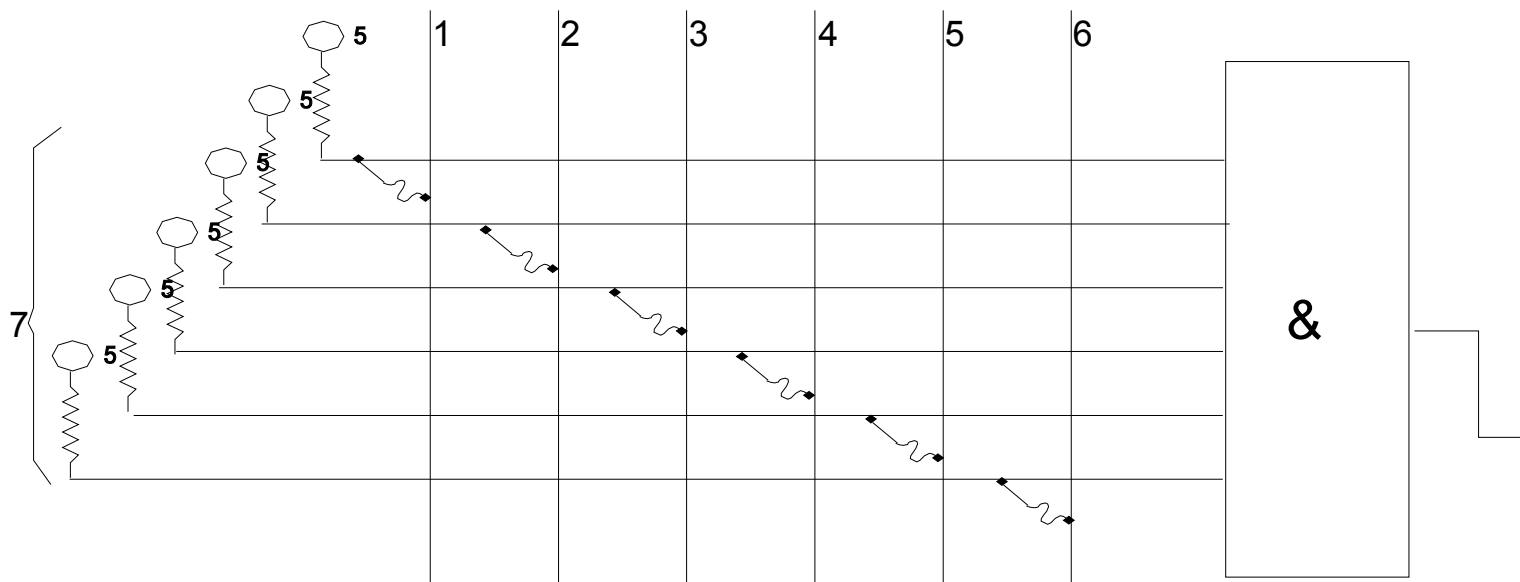


SUBSISTEMAS COMBINACIONES DE PROPÓSITO GENERAL

Dispositivos lógicos programables

PAL

Detalle de la generación de un término producto

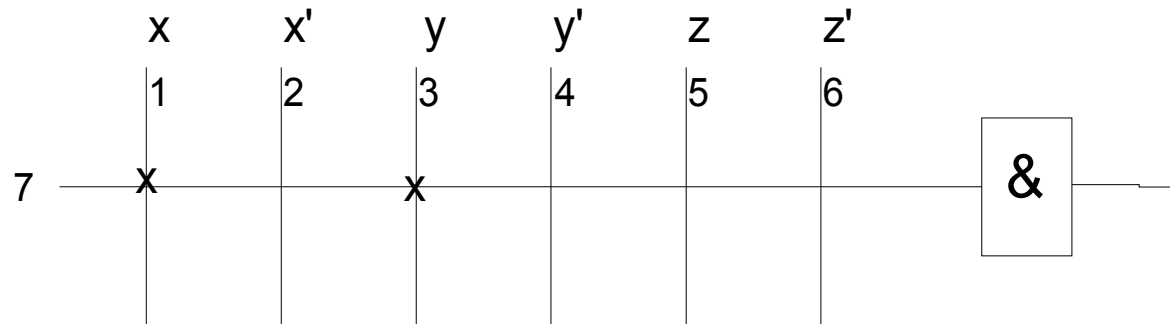


SUBSISTEMAS COMBINACIONES DE PROPÓSITO GENERAL

Dispositivos lógicos programables

PAL

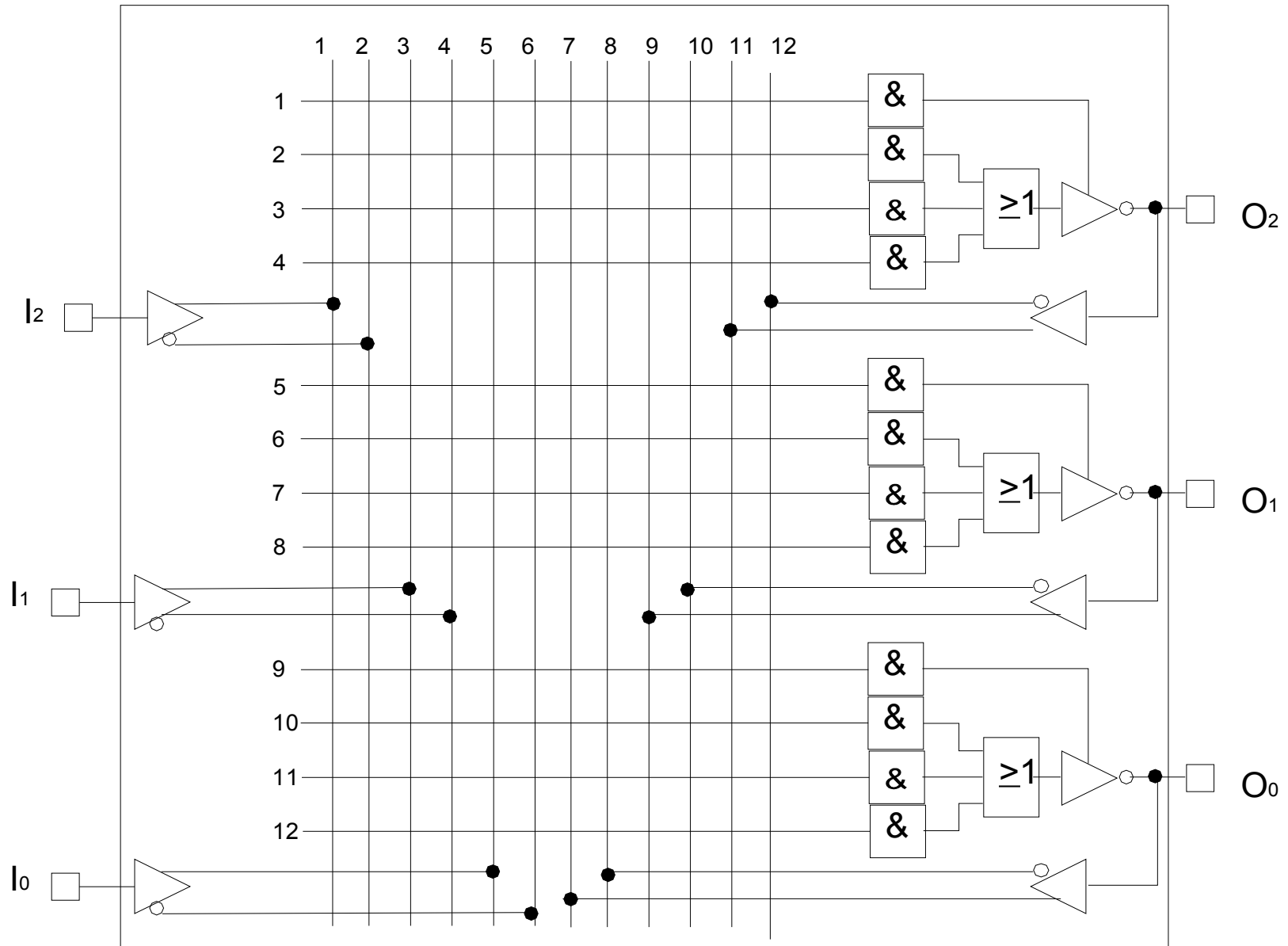
Representación simplificada del término producto



SUBSISTEMAS COMBINACIONES DE PROPÓSITO GENERAL

Dispositivos lógicos programables

PAL



SUBSISTEMAS COMBINACIONES DE PROPÓSITO GENERAL

Dispositivos lógicos programables

PAL

Ejemplo: Usando la primera estructura PAL, implementar las funciones de conmutación siguientes $f = \Sigma(0, 2, 4, 7)$, $g = \Sigma(0, 1, 2, 6, 7)$, $h = \Sigma(6, 7)$.

$$H(x, y, z) = xyz' + xyz$$

		xy			
		00	01	11	10
z	0	1	1		1
	1			1	

f

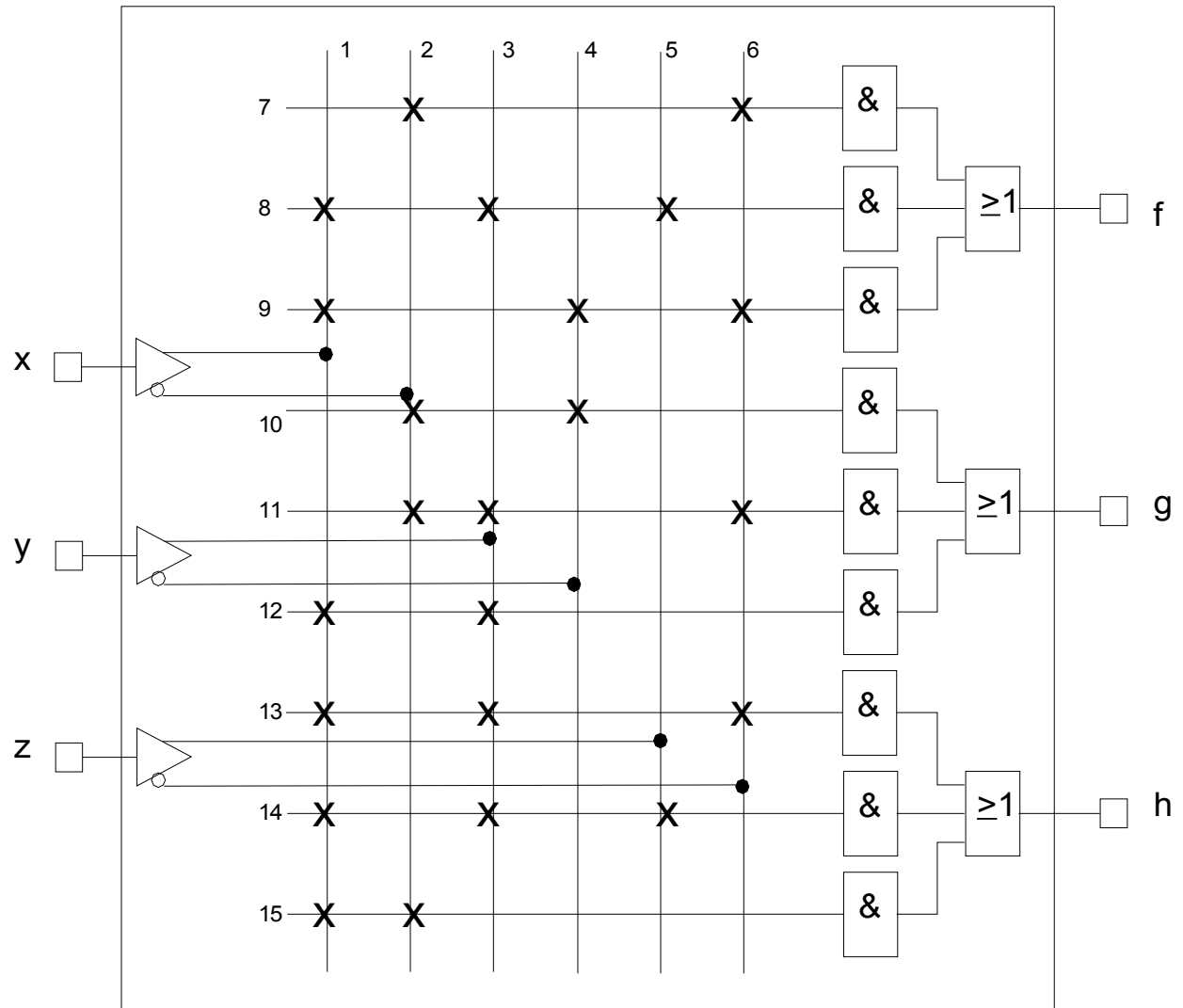
		xy			
		00	01	11	10
z	0	1	1	1	
	1	1		1	

g

Las expresiones para f y g son:

$$F(x, y, z) = x'z' + xyz + xy'z'$$

$$G(x, y, z) = x'y' + x'yz' + xy$$



SUBSISTEMAS COMBINACIONES DE PROPÓSITO GENERAL

Dispositivos lógicos programables

PLA

Este dispositivo tiene tanto el plano AND como el OR totalmente programables. La siguiente figura muestra la estructura de una PLA de 3 entradas y 3 salidas.



SUBSISTEMAS COMBINACIONES DE PROPÓSITO GENERAL

Dispositivos lógicos programables

PLA

Ejemplo. Implementar las siguientes funciones en la PLA anterior

$$F = x' + y$$

$$G = x'y + xz'$$

$$H = xz' + x' + y$$

