

Tema 6: Circuitos Digitales Básicos

Escuela Politécnica Superior
Ingeniería Informática
Universidad Autónoma de Madrid



Circuitos digitales básicos

O B J E T I V O S

⇒ **Comprender** las funciones lógicas elementales

⇒ **Habilidad para diseñar** funciones lógicas más complejas (sumadores, codificadores, decodificadores, etc.).

TEMA 6: CIRCUITOS DIGITALES BÁSICOS

6.1 Realización de funciones lógicas con puertas

6.2 Circuitos combinacionales

Sumadores
Comparadores
Decodificadores
Codificadores
Conversores de código
Multiplexores
Demultiplexores

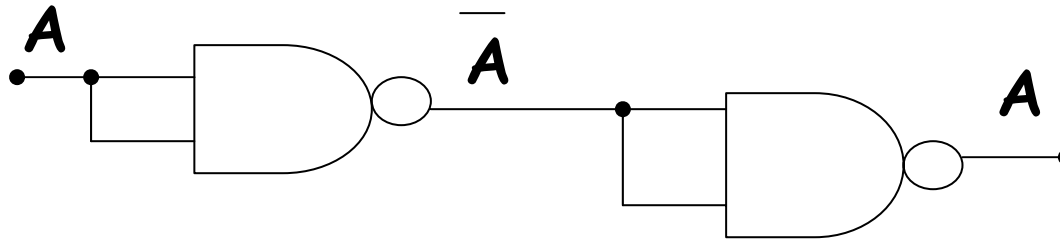
Bibliografía Tema 6:

- *Introduction to Computer Hardware and Data Communications*. P.-A. GOUPILLE. (Prentice Hall, 1993). Cap. 9.
- *Fundamentos de Sistemas Digitales*. T. L. FLOYD. 7ª Ed. (Prentice Hall, 2000). Cap. 4, apartado 4.12. Cap. 6.



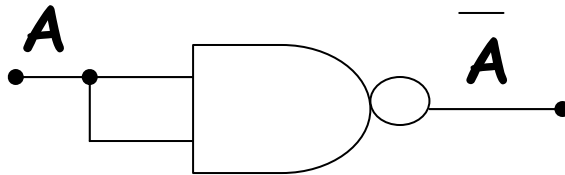
Puerta NAND como elemento lógico universal

- AMPLIFICADOR



Dos puertas NAND utilizadas como amplificador

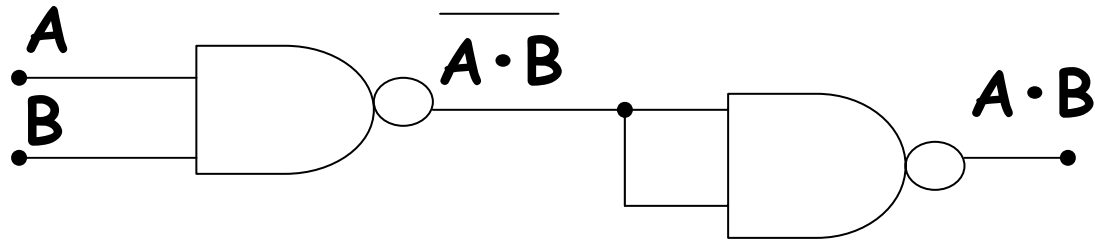
- NOT



Una puerta NAND utilizada como inversor

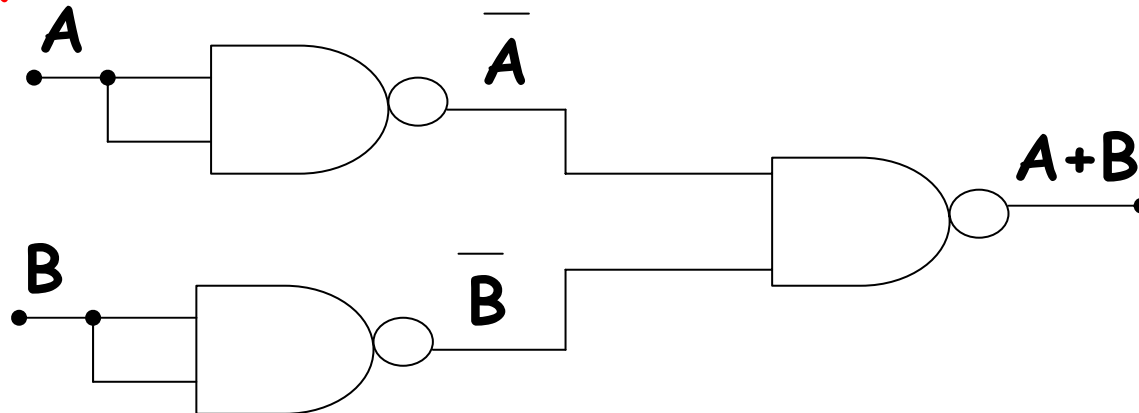
Puerta NAND como elemento lógico universal

- AND



Dos puertas NAND utilizadas como puerta AND

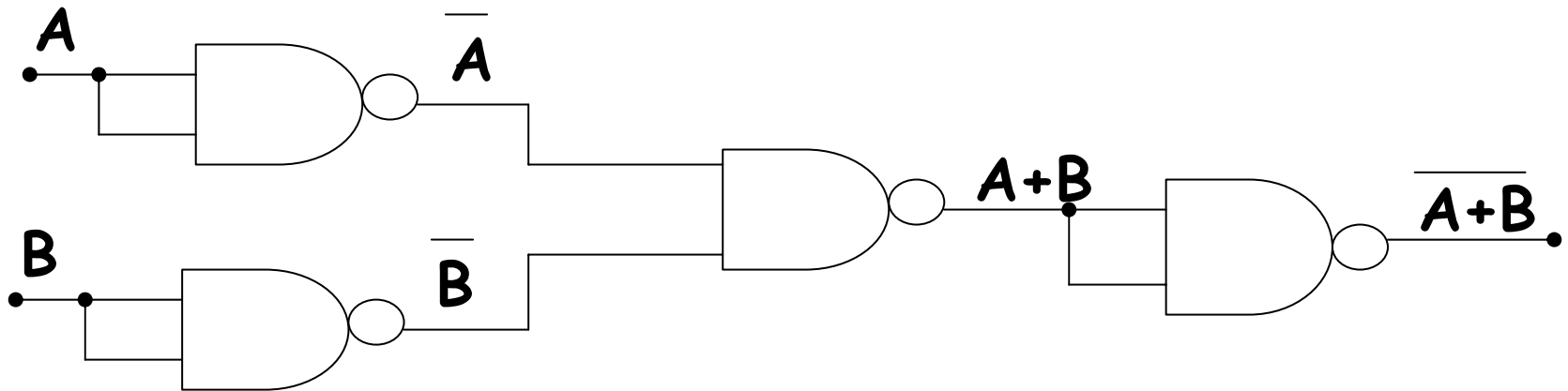
- OR



Tres puertas NAND utilizadas como puerta OR

Puerta NAND como elemento lógico universal

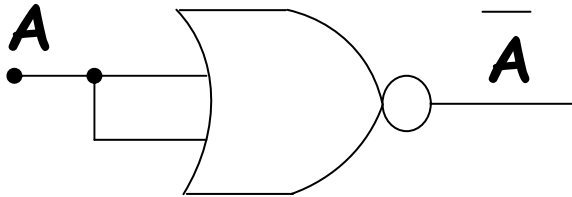
- NOR



Cuatro puertas NAND utilizadas como puerta NOR

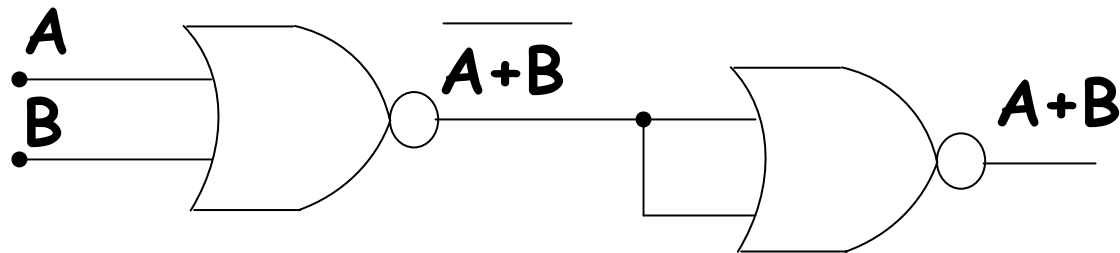
Puerta NOR como elemento lógico universal

- NOT O INVERSOR



Una puerta NOR utilizada como inversor

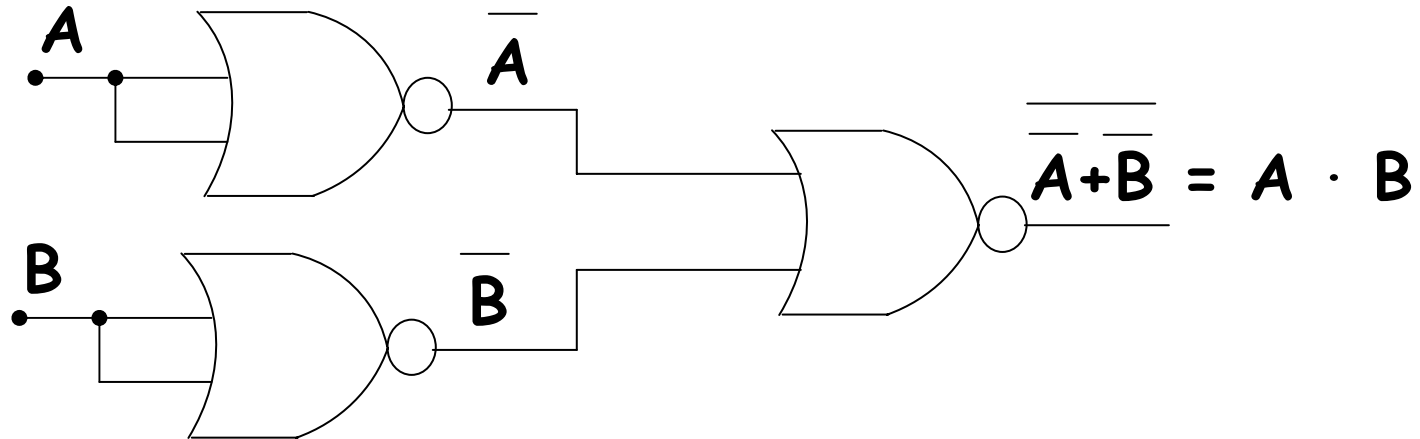
- OR



Dos puertas NOR utilizadas como puerta OR

Puerta NOR como elemento lógico universal

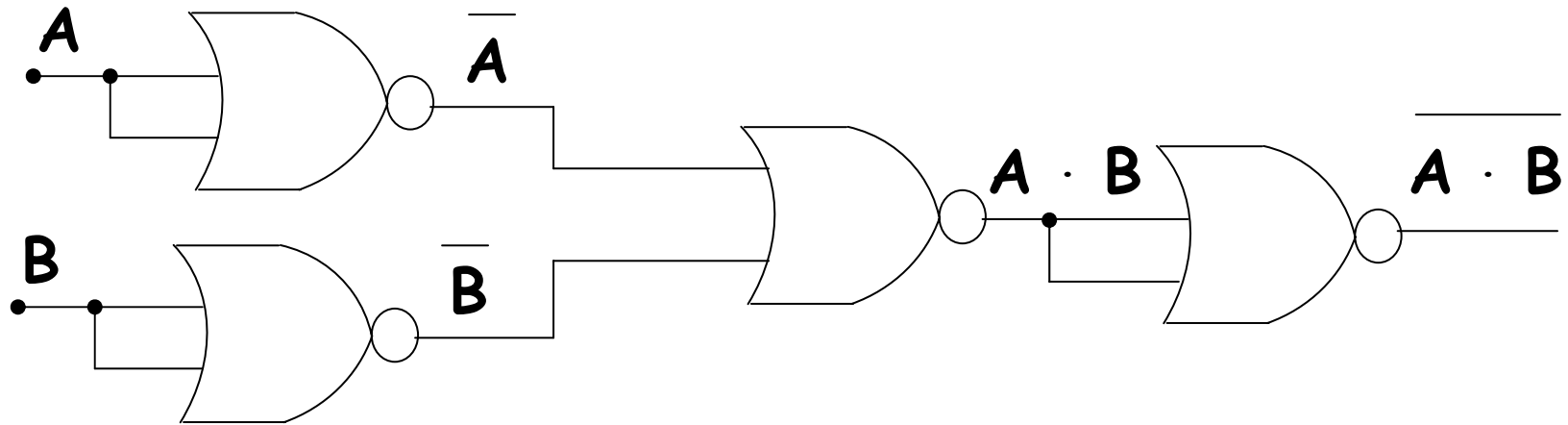
- AND



Tres puertas NOR utilizadas como puerta AND

Puerta NOR como elemento lógico universal

- NAND



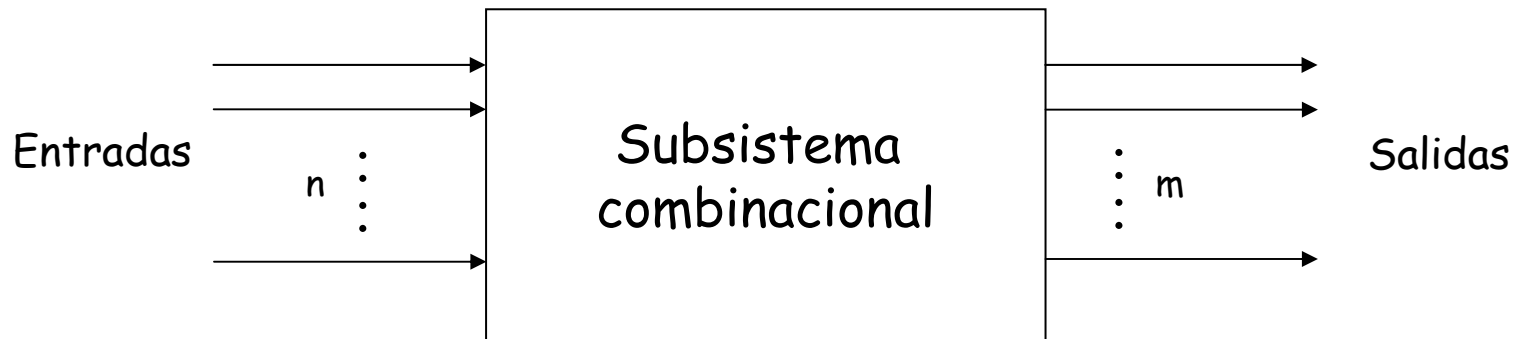
Cuatro puertas NOR utilizadas como puerta NAND

Circuitos combinacionales

- Con la ayuda de las puertas y los circuitos lógicos elementales estudiados en las sesiones anteriores es posible construir los principales elementos de un ordenador
- **Lógica Combinacional** (Goupille, 1993, p. 71): En la lógica combinacional, a diferencia de la lógica secuencial, el estado de la salida depende sólo de la combinación de los valores lógicos presentes actualmente en las entradas
- **Circuito combinacional** (Prieto, 2002, p. 153): un circuito digital, con entradas x_1, \dots, x_n , y salidas z_1, \dots, z_m , es combinacional si las salidas en un instante cualquiera están determinadas exclusivamente por las entradas en ese mismo instante; se dice que es un sistema o subsistema sin memoria

Circuitos combinacionales

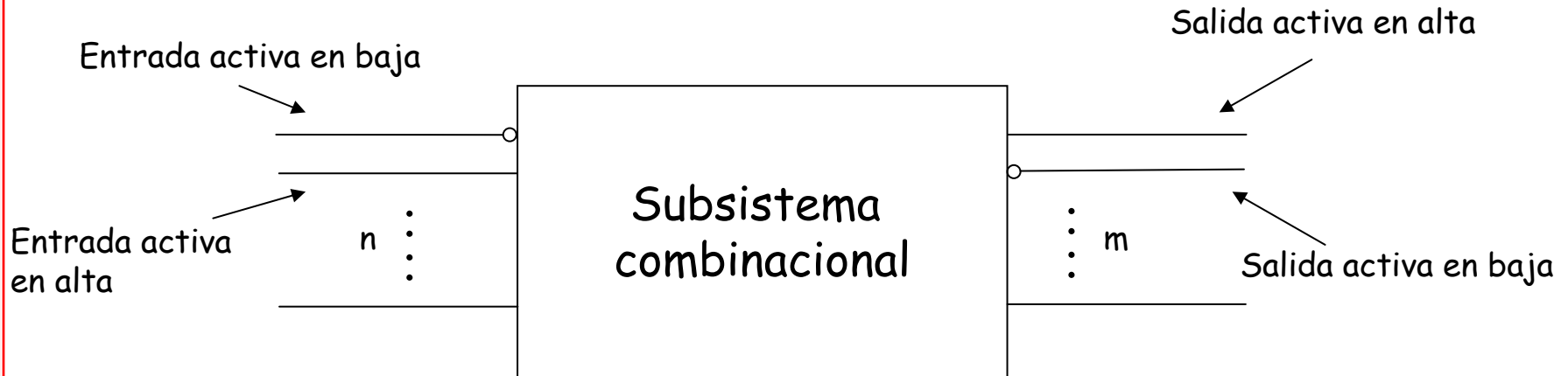
- **Subsistema:** todo circuito integrado cuya complejidad supere al de una simple puerta lógica. Se caracterizan fundamentalmente por su capacidad de **reutilización**
- Un subsistema puede verse como una caja negra con múltiples entradas y salidas:



- Las líneas de entrada y salida pueden clasificarse en:
 - **Líneas de datos:** llevan la información original o procesada por el subsistema
 - **Líneas de control:** indican al subsistema qué operación realizar o permiten al subsistema indicar al usuario el estado resultante de la operación

Circuitos combinacionales

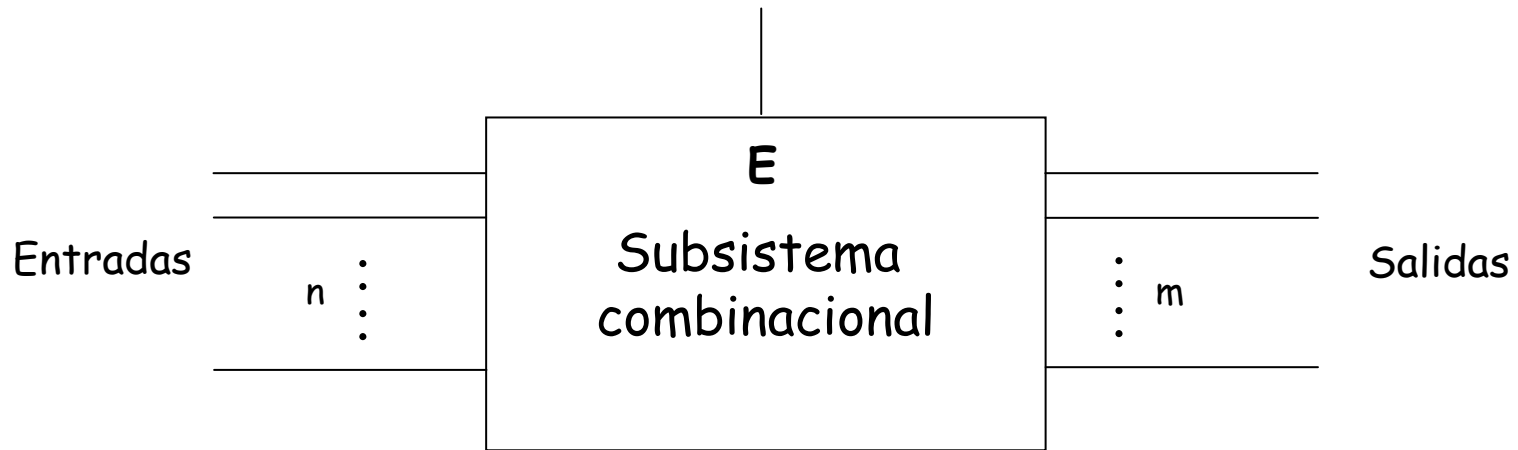
- Las líneas de control se clasifican según su nivel de activación en:
 - Activas a nivel alto o "activas en alta": es decir, con nivel de activación = 1
 - Activas a nivel bajo o "activas en baja": es decir, con nivel de activación = 0



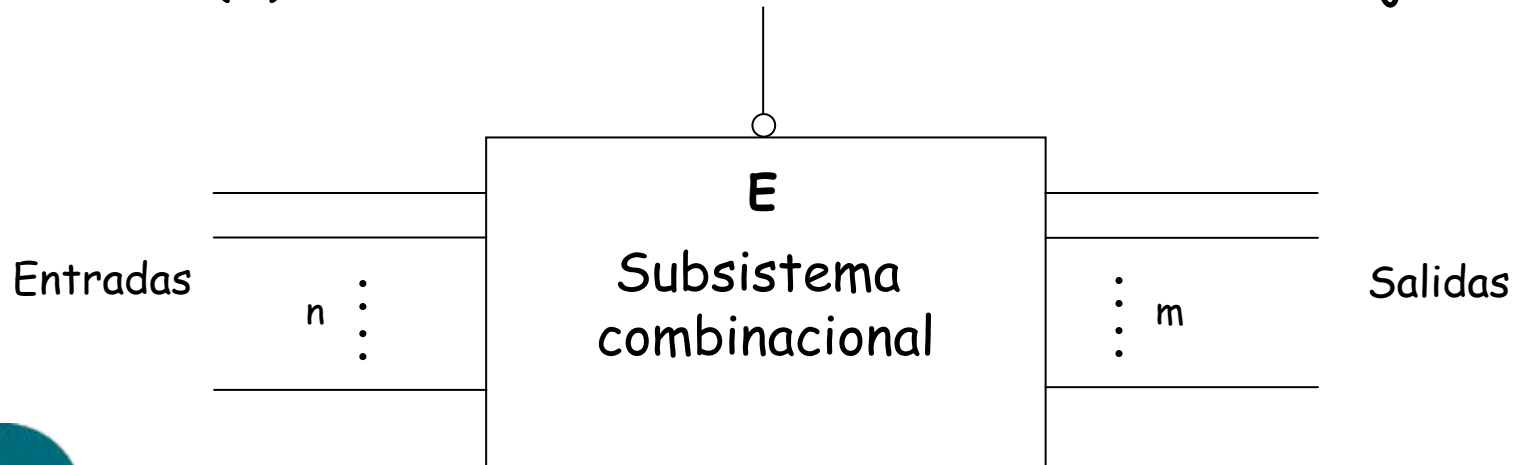
- Una de las entradas de control más comunes es **Enable** (habilitación):
 - **Enable activo**: el subsistema opera normalmente
 - **Enable inactivo**: el subsistema entra en un estado de reposo

Circuitos combinacionales

- Enable (E) o entrada de habilitación activa a nivel alto:



- Enable (E) o entrada de habilitación activa a nivel bajo:



Circuitos combinacionales

- Con estos circuitos se construye, entre otros, los dispositivos que realizan las operaciones estudiadas en el tema de Sistemas de Numeración-Codificación Binaria, que se centró sobre la representación de datos
- En concreto, se considerarán: **Sumadores, Comparadores, Decodificadores, Codificadores, Conversores de Código, Multiplexores y Demultiplexores**

Sumadores

- Reglas básicas de la suma binaria

$$0 + 0 = 0$$

$$0 + 1 = 1$$

$$1 + 0 = 1$$

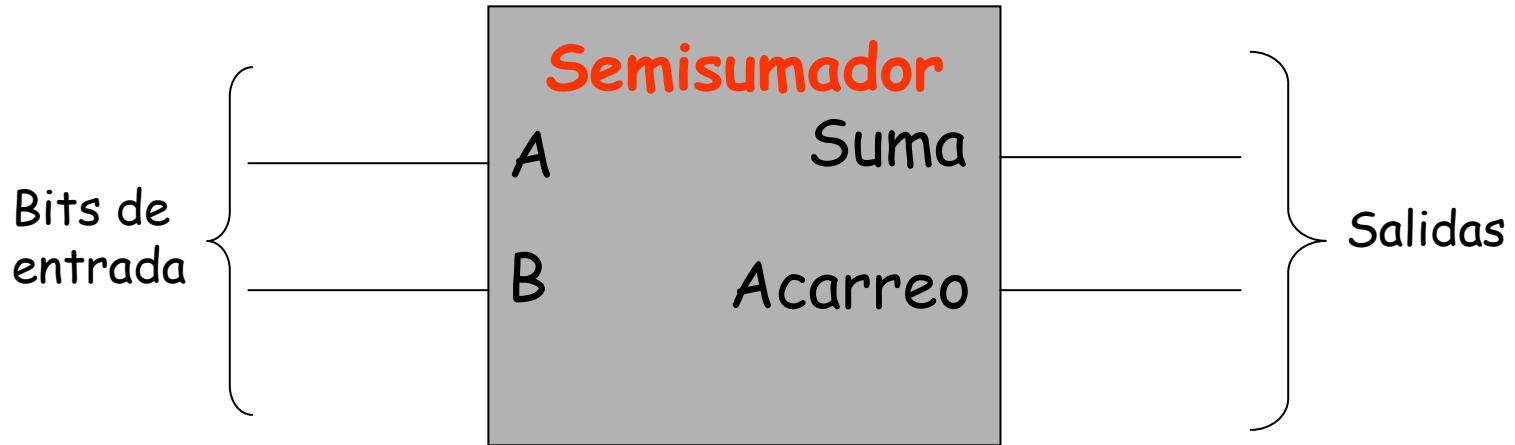
$$1 + 1 = 10$$

- Estas operaciones se realizan mediante un circuito lógico (compuesto de puertas lógicas) denominado semisumador
- Semisumador** - (Floyd, 2000, p. 332-333)
 - Propósito: permite sumar dos bits sin tener en cuenta los acarreos provenientes de la adición de bits anteriores
 - Un semisumador admite dos dígitos binarios en sus entradas y genera dos dígitos binarios en sus salidas: un bit de suma y un bit de acarreo

Sumadores

- **Semisumador** - (Floyd, 2000, p. 332-333)

- Símbolo lógico:



- Tabla de verdad:

A	B	Suma	C _{out}
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Suma = Σ

C_{out} = acarreo de salida

A y B = variables de entrada (operandos)

Sumadores

- **Semisumador** - (Floyd, 2000, p. 332-333)
 - **Ecuaciones:** A partir del funcionamiento lógico de un semisumador mostrado en la tabla de verdad, las expresiones correspondientes a la SUMA y al ACARREO DE SALIDA se pueden obtener como funciones de las entradas. Es decir:
 - * La salida de acarreo C_{out} es 1 sólo cuando A y B son 1. Por tanto, C_{out} puede expresarse como una operación AND de las variables de entrada:

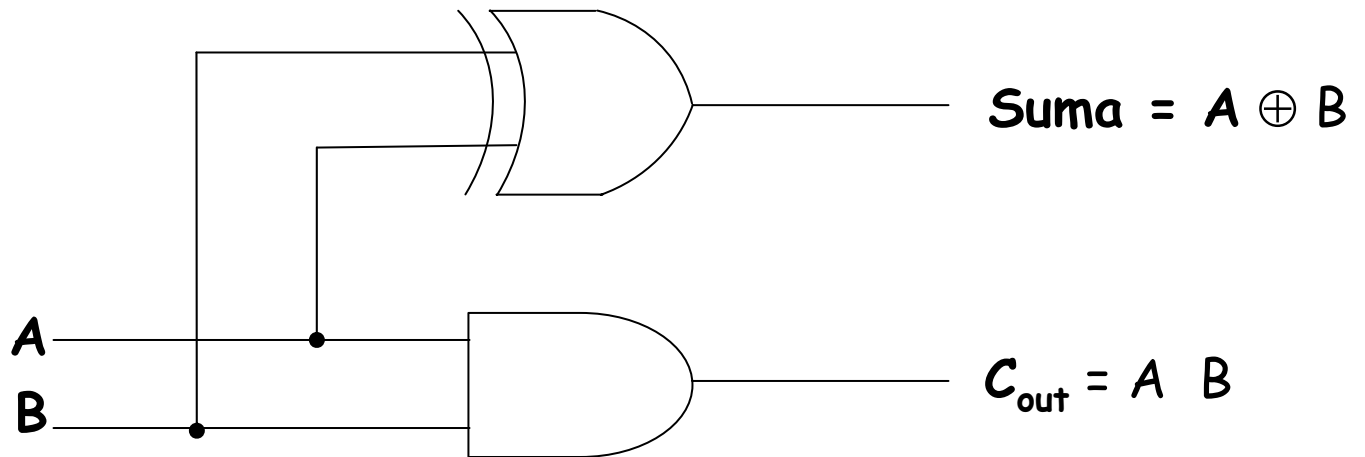
$$C_{out} = A \cdot B$$

- * La salida correspondiente a la **Suma** es 1 sólo si las variables A y B son distintas. Por tanto, la suma puede expresarse como una operación OR-exclusiva (XOR) de las variables de entrada:

$$Suma = A \oplus B$$

Sumadores

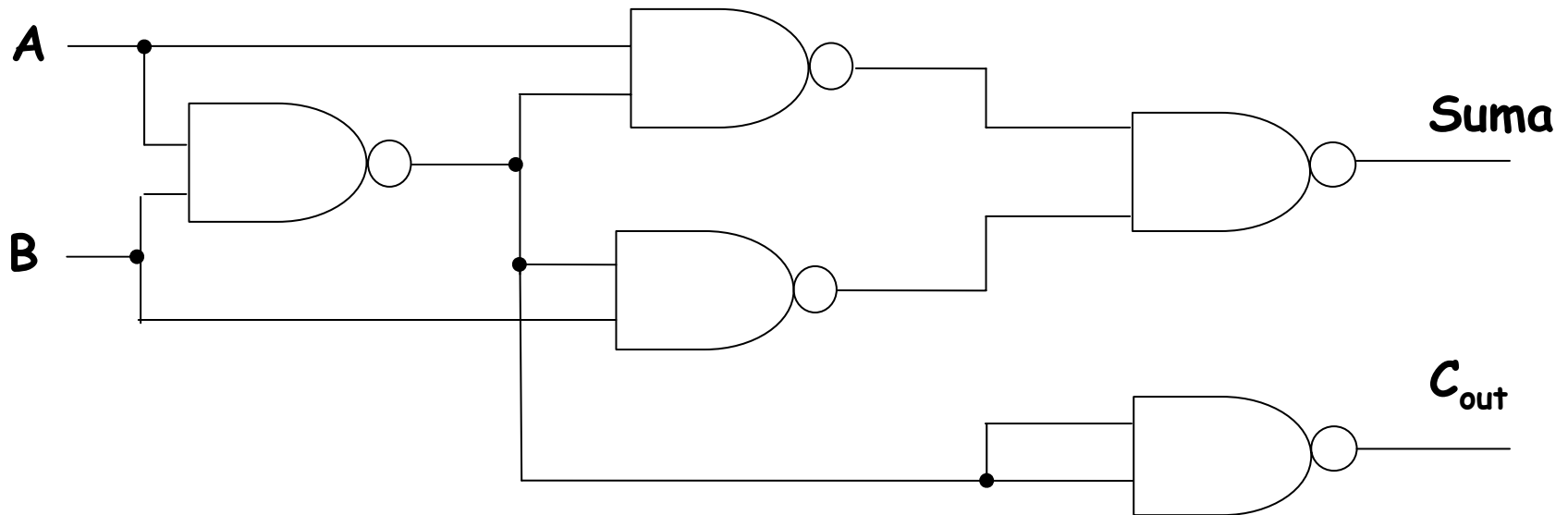
- **Semisumador** - (Floyd, 2000, p. 332-333)
 - Circuito lógico:



Sumadores

- **Semisumador con puertas NAND**

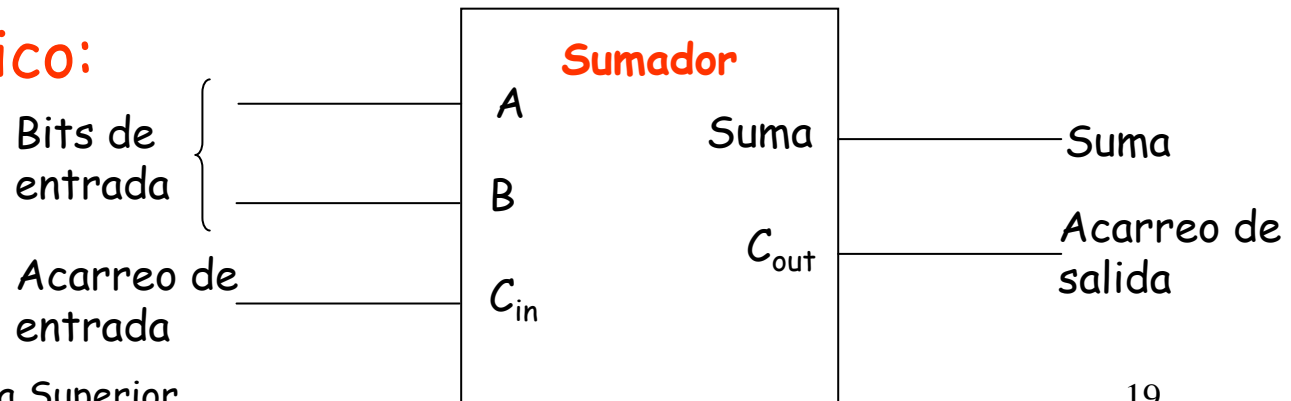
- Todo circuito puede construirse utilizando puertas NAND, siendo estas puertas más económicas



Sumadores

- **Sumador completo** - (Floyd, 2000, p. 333-335)
 - **Propósito:** permite sumar dos bits teniendo en cuenta los acarreos provenientes de la adición de bits anteriores
 - **Un sumador completo acepta dos bits de entrada y un acarreo de entrada, y genera una salida de suma y un acarreo de salida**

- **Símbolo lógico:**



Sumadores

- **Sumador completo** - (Floyd, 2000, p. 333-335)

- Tabla de verdad:

A	B	C_{in}	Suma	C_{out}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

C_{in} = acarreo de entrada, algunas veces se designa por CI

C_{out} = acarreo de salida, algunas veces se designa por CO

Suma = Σ

A y B = variables de entrada (operandos)

Sumadores

- **Sumador completo** - (Floyd, 2000, p. 333-335)
 - **Ecuaciones:** Un sumador completo suma los dos bits de entrada y el bit de acarreo de entrada
 - * A partir del semisumador, sabemos que la suma de los dos bits de entrada A y B consiste en la operación OR-exclusiva (XOR) entre esas dos variables, $A \oplus B$. Para la SUMA del acarreo de entrada (C_{in}) a los bits de entrada, hay que volver a aplicar la operación OR-exclusiva (XOR), obteniéndose la siguiente ecuación de salida:

$$\text{Suma} = (A \oplus B) \oplus C_{in}$$

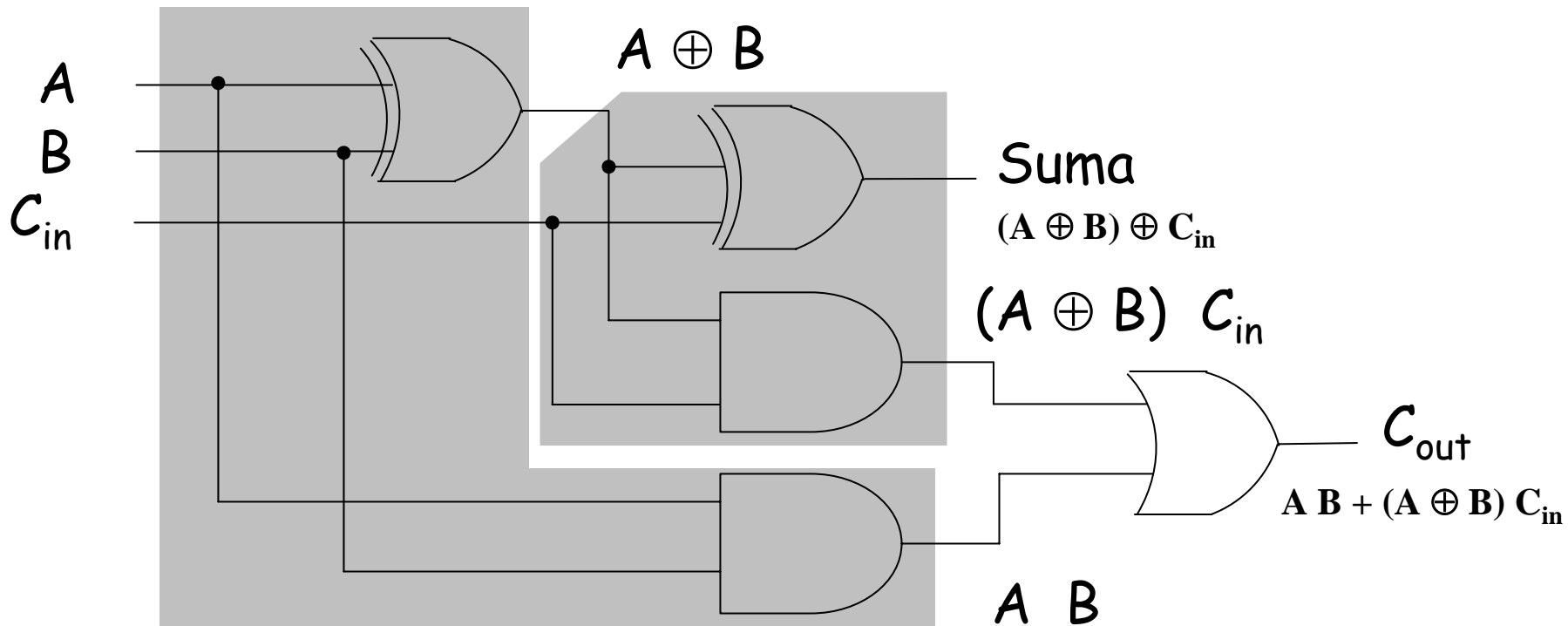
- * El ACARREO DE SALIDA es 1 cuando las dos entradas de la primera puerta XOR son 1, o cuando las dos entradas de la segunda puerta XOR son 1. Por tanto:

$$C_{out} = A B + (A \oplus B) C_{in}$$

Nota: Esta función, una vez implementada, se combina con la de la suma lógica para constituir un circuito sumador completo

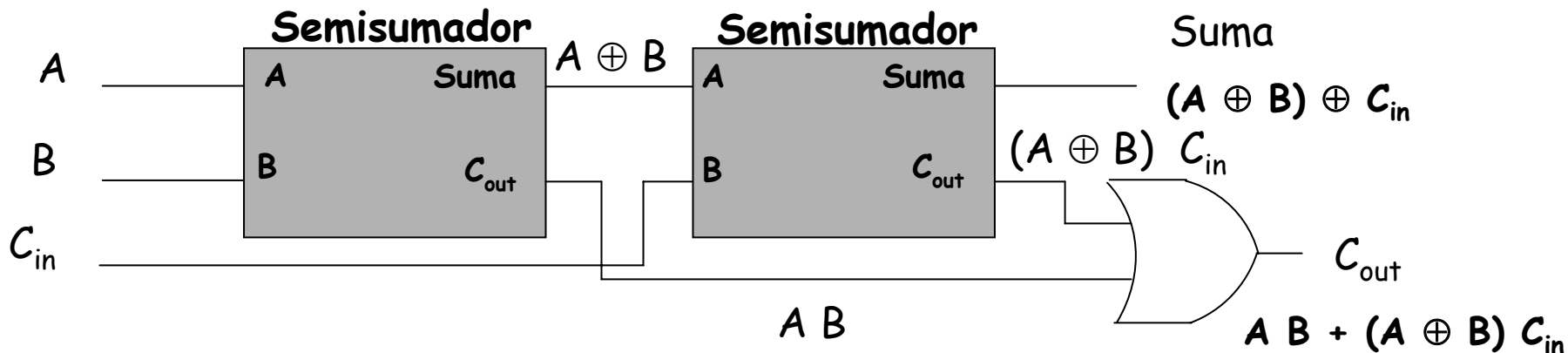
Sumadores

- **Sumador completo** - (Floyd, 2000, p. 333-335)
 - **Circuito lógico** de un sumador completo (cada semisumador se representa por un área sombreada):



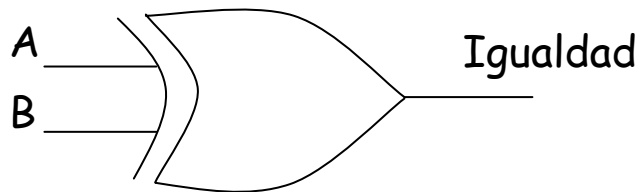
Sumadores

- **Sumador completo** - (Floyd, 2000, p. 333-335)
 - Sumador completo implementado mediante semisumadores:



Comparadores

- La función básica de un Comparador consiste en comparar las magnitudes de dos cantidades binarias para determinar su relación, es decir si $A=B$ (comparador básico), $A>B$, $A<B$ - (Floyd, 2000, p. 345-349)
- Podemos tener un circuito básico comparador que determine si dos números son iguales. La puerta XOR se puede emplear como un comparador básico, ya que su salida es 1 si sus dos bits de entrada son diferentes y 0 si son iguales
- Circuito, ecuación y tabla de verdad:



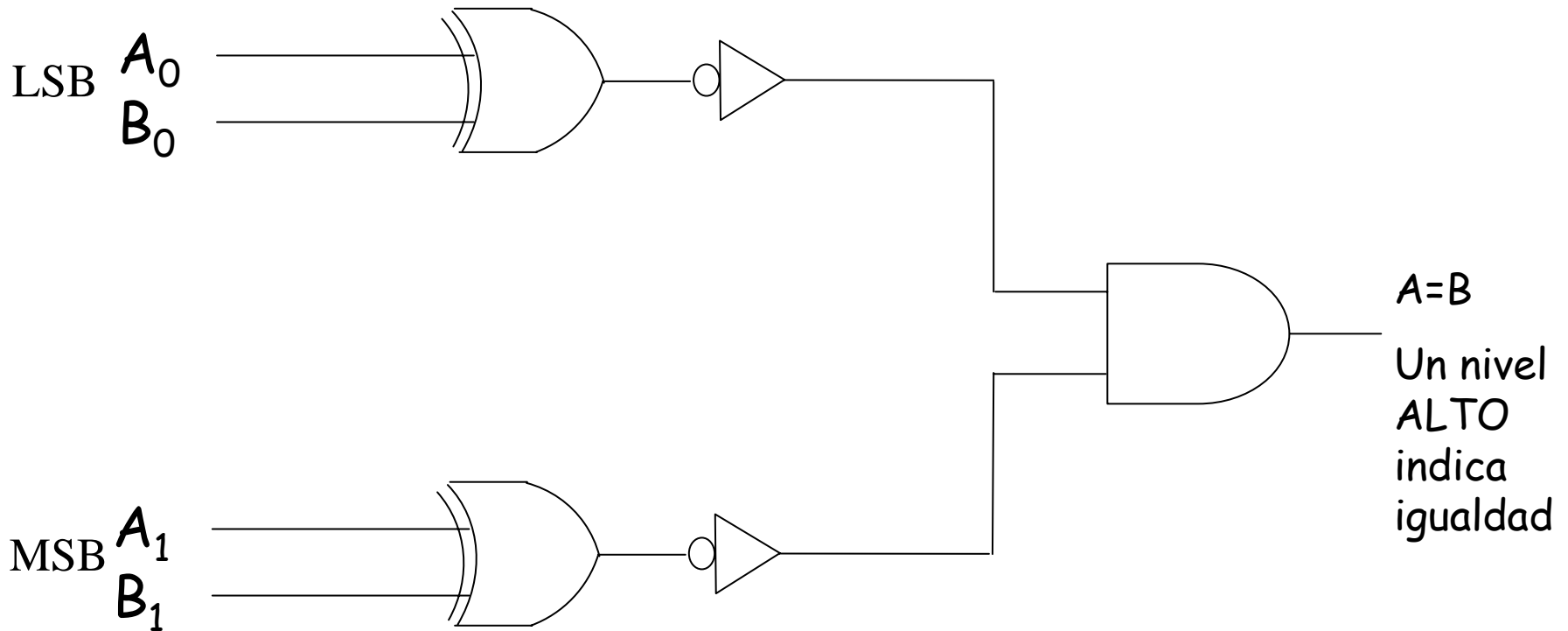
$$\text{Igualdad} = A \oplus B$$

A	B	Igualdad
0	0	0 Los bits de entrada son iguales
0	1	1 Los bits de entrada son distintos
1	0	1 Los bits de entrada son distintos
1	1	0 Los bits de entrada son iguales

Comparadores

- Comparador de igualdad de dos números binarios de 2 bits

- Circuito:



Formato general: Número binario $A=A_1A_0$

Número binario $B=B_1B_0$

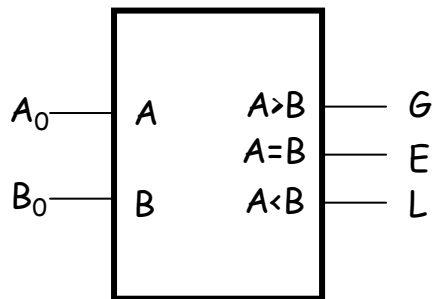
Comparadores

- Estructura:

- Entradas: dos números de n bits cada uno ($A_{n-1,0}$, $B_{n-1,0}$)
- Salidas: resultados de la comparación G ($A > B$), E ($A = B$), L ($A < B$) activas en alta

- Para $n=1$ bit, se tiene un comparador de magnitud de 2 números de 1 bit cada uno con indicación de desigualdad

- Símbolo lógico y tabla de verdad:



	A	B	$A > B$ G	$A = B$ E	$A < B$ L
	0	0	0	1	0
	0	1	0	0	1
	1	0	1	0	0
	1	1	0	1	0

- **Ejercicio:** Determinar las ecuaciones del comparador de magnitud de 2 bits y su circuito lógico correspondiente

Comparadores

- Para $n=4$ bits, se tiene un comparador de 4 bits con indicación de desigualdad

- Algoritmo:

1. Comenzando por el bit de mayor orden y para cada bit dentro del número, examinar (i =peso de bit=3,2,1,0):
 - 1.1. Si $A_i=1$ y $B_i=0$, entonces A es mayor que B
 - 1.2. Si $A_i=0$ y $B_i=1$, entonces A es menor que B
 - 1.3. Si $A_i=B_i$, entonces tenemos que examinar los siguientes bits de orden inmediatamente inferior

Hasta que se encuentre una desigualdad

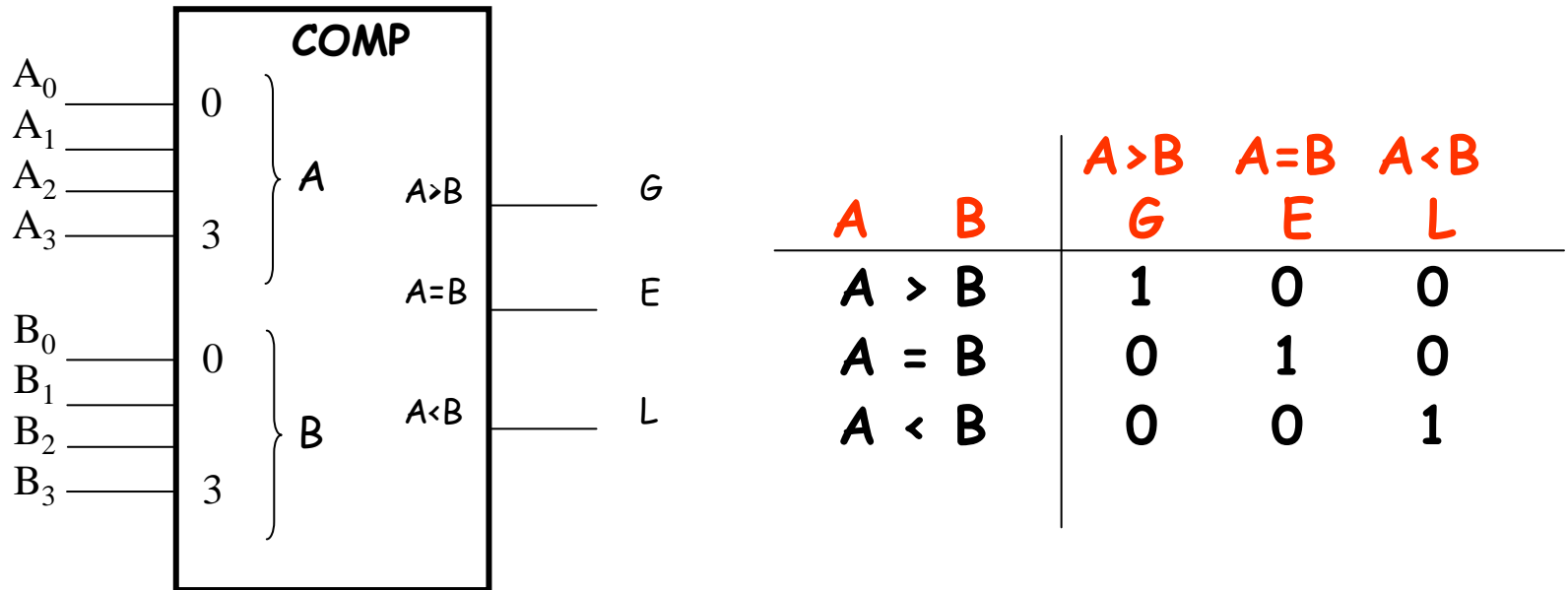
- En el algoritmo, la relación de más alto orden es la que tiene prioridad



Comparadores

- Comparador de dos números de 4 bits con indicación de desigualdad

- Símbolo lógico y tabla de verdad resumida:



- **Ejercicio:** ¿Cuáles serán las salidas del comparador cuando $A_3A_2A_1A_0 = 0110$ y $B_3B_2B_1B_0 = 0011$?

Decodificadores

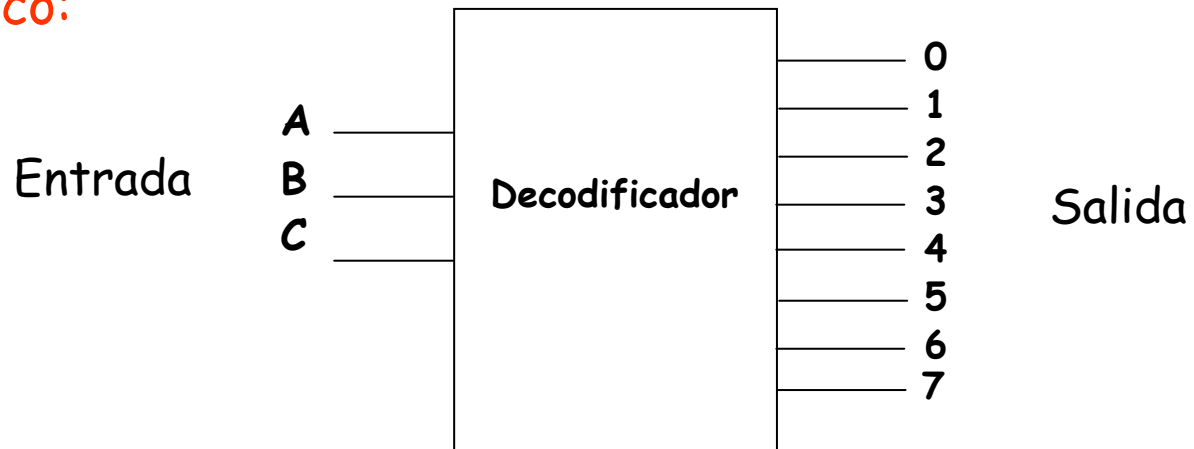
- La función básica de un decodificador es detectar la presencia de una determinada combinación de bits (código) en sus entradas y señalar la presencia de este código mediante un cierto nivel de salida (Floyd, 2000, p. 350-360)
- **Estructura:**
 - **n** entradas de datos
 - **m** salidas de datos, con $m \leq 2^n$, activas en alta o baja
- **Denominación:** $DEC_n a m$
- **Propósito:** generar m minterms (salidas activas en alta) o m maxterms (salidas activas en baja) asociados a las n entradas
- Por tanto, tan sólo puede haber una **única salida activa** al mismo tiempo para cada combinación de las entradas



Decodificadores

- Un decodificador. Noción Básica

- Estructura: 3 entradas de datos y $2^3=8$ salidas de datos
- Ejemplo: Si las entradas son $C = 1$, $B = 1$, $A = 0$, la salida activada será la línea que corresponde al 6 (110_2 corresponde a 6_{10})
- Símbolo lógico:



- Un decodificador es un conjunto de circuitos lógicos que puede convertir, por ejemplo un número binario a un número decimal o bien convertir cada código BCD en uno de los 10 dígitos decimales

Decodificadores

• Decodificador 2 a 4


- Estructura: 2 entradas de datos y $2^2=4$ salidas de datos
- Para salidas activas a nivel alto

* Símbolo lógico, tabla de verdad y ecuaciones de salida:

A_1					A		O							
					A_1	A_0	O_0	O_1	O_2	O_3	$O_0 = \overline{A_1} \overline{A_0}$	$O_1 = \overline{A_1} A_0$	$O_2 = A_1 \overline{A_0}$	$O_3 = A_1 A_0$
A_0	1	0	O_0											
		1	O_1		0	0	1	0	0	0	0			
	0	2	O_2		0	1	0	1	0	0	0			
		3	O_3		1	0	0	0	1	0	0			
					1	1		0	0	0	0			

- Para salidas activas a nivel bajo

* Símbolo lógico, tabla de verdad y ecuaciones de salida:



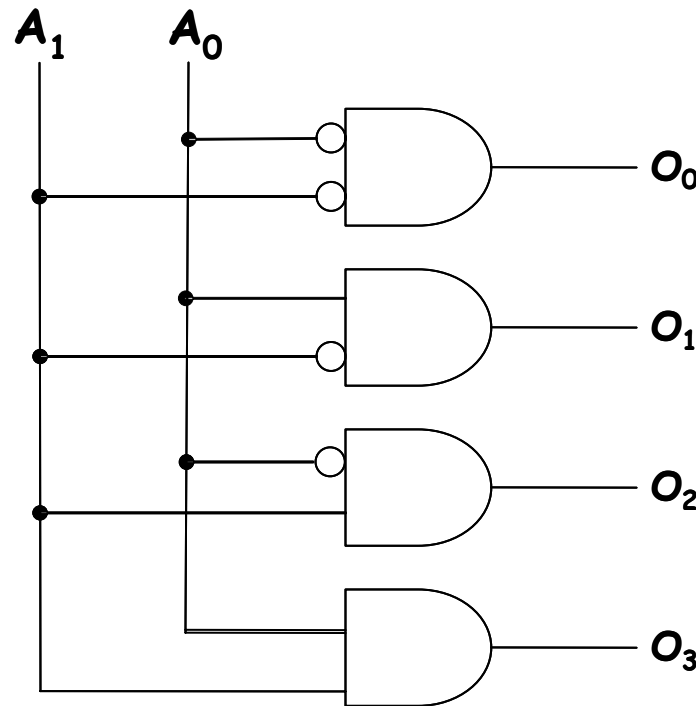
A_1	1	0	O_0	A_1	A_0	O_0	O_1	O_2	O_3	$O_0 = A_1 + A_0$ $O_1 = A_1 + \overline{A_0}$ $O_2 = \overline{A_1} + A_0$ $O_3 = \overline{A_1} + \overline{A_0}$		
		1	O_1	0	0	0	1	1	1			
A_0	0	2	O_2	0	1	1	0	1	1			
		3	O_3	1	0	1	1	0	1			
				1	1	1	1	1	0			

31



Decodificadores

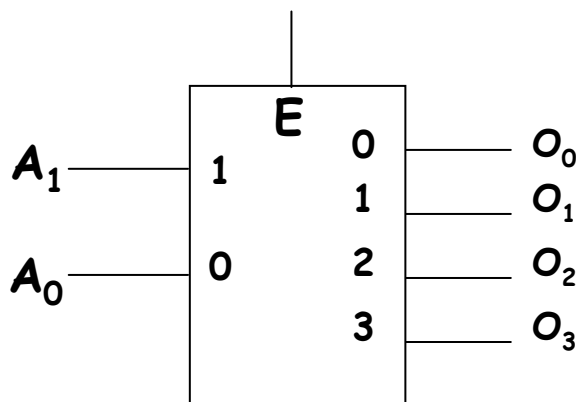
- Decodificador 2 a 4
 - Circuito lógico con salidas activas a nivel alto:



- **Ejercicio:** Dibujar el circuito del decodificador 2 a 4 con salidas activas a nivel bajo

Decodificadores

- Decodificador 2 a 4 con entrada Enable activa a nivel alto y salidas activas a nivel alto
 - Símbolo lógico, tabla de verdad y ecuaciones de salida:



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

$$O_0 = E \overline{A_1} \overline{A_0} \quad O_1 = E \overline{A_1} A_0 \quad O_2 = E A_1 \overline{A_0} \quad O_3 = E A_1 A_0$$

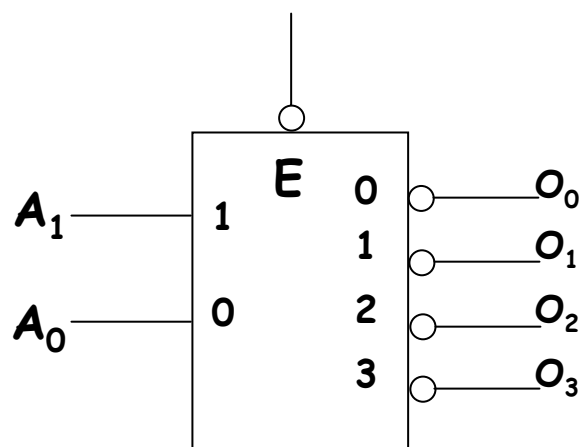
Nota 1: X indica que el nivel (0 o 1) no es importante. No confundir estas X con funciones incompletamente especificadas

Nota 2: Nótese que E (Enable) multiplica a cada ecuación de un decodificador 2 a 4

Decodificadores

- Decodificador 2 a 4 con entrada Enable activa a nivel bajo y salidas activas a nivel bajo

- Símbolo lógico, tabla de verdad y ecuaciones de salida:

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

$$O_0 = E + A_1 + A_0 \quad O_1 = E + A_1 + \overline{A_0} \quad O_2 = E + \overline{A_1} + A_0 \quad O_3 = E + \overline{A_1} + \overline{A_0}$$

- **Ejercicio:** Dibujar los circuitos lógicos correspondientes a ambos tipos de decodificadores con entrada Enable

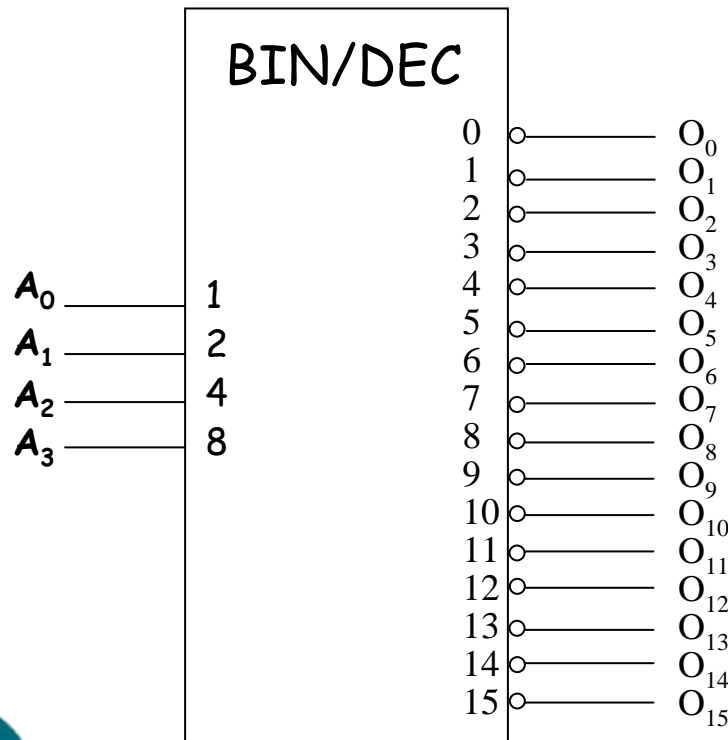
Decodificadores

- **Decodificador 4 a 16 con salidas activas a nivel bajo**
 - Estructura: 4 entradas de datos y $2^4=16$ salidas de datos
 - En una de las 2^4 líneas de salida se indica la presencia de una combinación de 4 bits. Se denomina también decodificador 1 de 16, ya que para cualquier código dado en las entradas, sólo se activa 1 de las 16 posibles salidas



Decodificadores

- Decodificador 4 a 16 con salidas activas a nivel bajo
 - Símbolo lógico (la etiqueta BIN/DEC indica que una entrada binaria produce su correspondiente salida decimal) y ecuaciones:



$$O_0 = A_3 + A_2 + A_1 + A_0$$

$$O_1 = A_3 + A_2 + A_1 + \overline{A_0}$$

$$O_2 = A_3 + A_2 + \overline{A_1} + A_0$$

$$O_3 = A_3 + A_2 + \overline{A_1} + \overline{A_0}$$

•
•
•

$$O_{15} = \overline{A_3} + \overline{A_2} + \overline{A_1} + \overline{A_0}$$

Decodificadores

- Decodificador binario

- El 74LS139 es un decodificador doble de tipo 1 a 4, con salidas activas a nivel BAJO o activas en BAJA. Tiene una entrada de habilitación G :

Si $G = 0$, el decodificador está activado

Si $G = 1$, el decodificador está inhibido (ninguna salida puede ser activada)

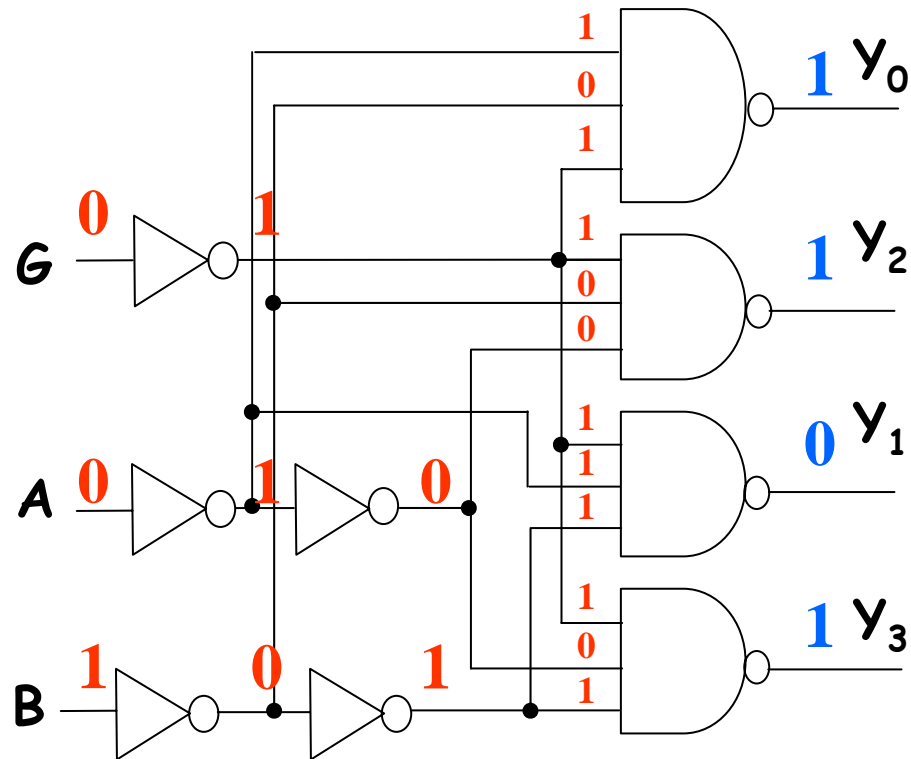
- Tabla de verdad:

G	A	B	Y_0	Y_1	Y_2	Y_3
1	x	x	1	1	1	1
0	0	0	0	1	1	1
0	0	1	1	0	1	1
0	1	0	1	1	0	1
0	1	1	1	1	1	0

Nota: X indica que el nivel (0 o 1) no es importante. No confundir estas X con funciones incompletamente especificadas

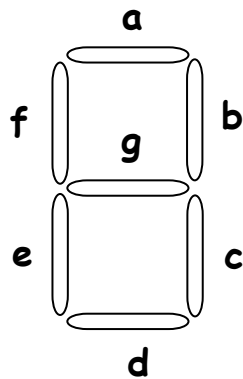
Decodificadores

- Decodificador binario
 - Circuito del 74LS139:

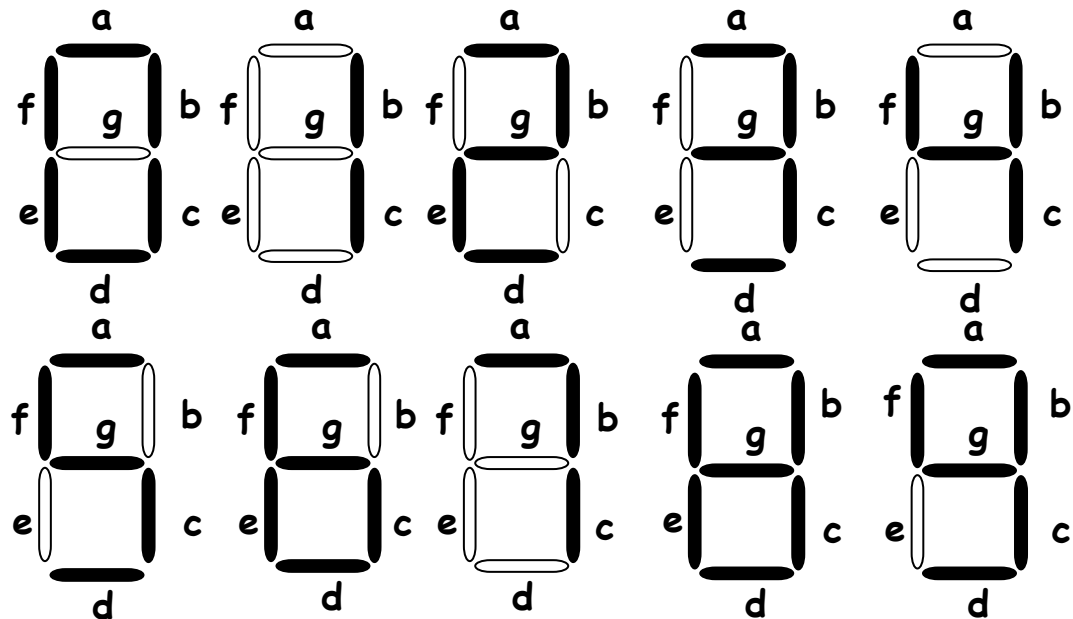


Decodificadores

- **Decodificador BCD a 7 segmentos** - (Floyd, 2000, p. 253-259)
 - Cada segmento se utiliza para representar varios dígitos decimales, pero ninguno de ellos se emplea para representar los diez dígitos; por tanto, cada segmento tiene que activarse mediante su propio circuito de decodificación que detecta la aparición de cualquier número en el que haya que usar ese segmento
 - **Display de 7 segmentos:**



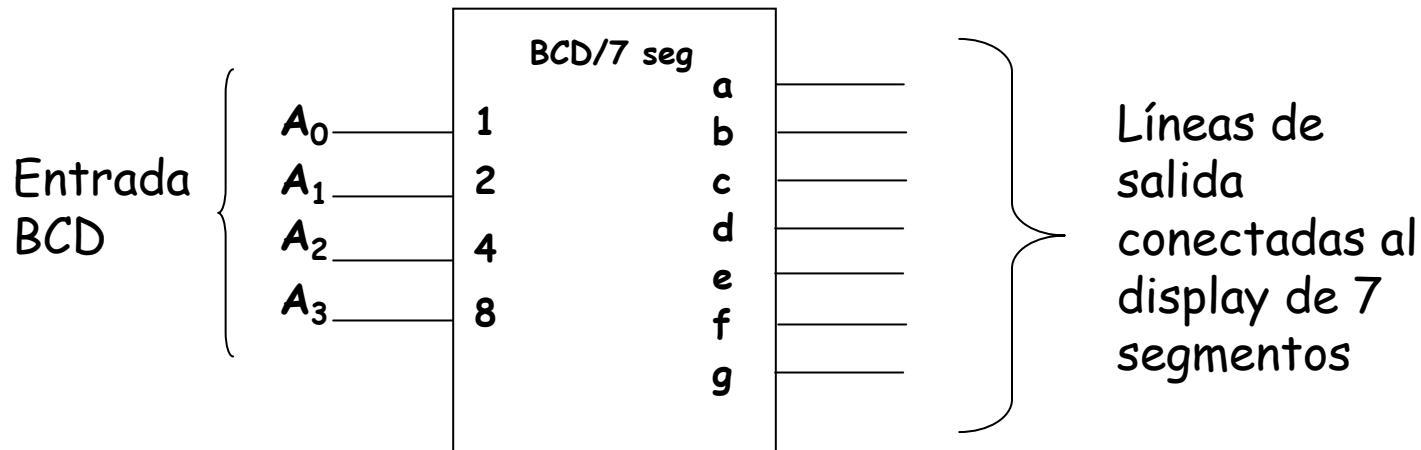
Disposición de los segmentos en un display de 7 segmentos



Segmentos activados para cada dígito decimal

Decodificadores

- **Decodificador BCD a 7 segmentos** - (Floyd, 2000, p. 357-360)
 - Este tipo de decodificador acepta código BCD en sus entradas y proporciona salidas capaces de excitar un display de 7 segmentos para indicar un dígito decimal
 - **Símbolo lógico de un decodificador/controlador BCD a 7 segmentos con salidas activas a nivel ALTO:**



Decodificadores

- Decodificador BCD a 7 segmentos - (Floyd, 2000, p. 253-259)

- Tabla de verdad para la lógica de 7 segmentos:

Dígito Decimal	Entradas				Salidas de Segmentos						
	D	C	B	A	a	b	c	d	e	f	g
0	0	0	0	0	1	1	1	1	1	1	0
1	0	0	0	1	0	1	1	0	0	0	0
2	0	0	1	0	1	1	0	1	1	0	1
3	0	0	1	1	1	1	1	1	0	0	1
4	0	1	0	0	0	1	1	0	0	1	1
5	0	1	0	1	1	0	1	1	0	1	1
6	0	1	1	0	1	0	1	1	1	1	1
7	0	1	1	1	1	1	1	0	0	0	0
8	1	0	0	0	1	1	1	1	1	1	1
9	1	0	0	1	1	1	1	1	0	1	1
10	1	0	1	0	x	x	x	x	x	x	x
11	1	0	1	1	x	x	x	x	x	x	x
12	1	1	0	0	x	x	x	x	x	x	x
13	1	1	0	1	x	x	x	x	x	x	x
14	1	1	1	0	x	x	x	x	x	x	x
15	1	1	1	1	x	x	x	x	x	x	x

Salida=1 significa que el segmento está activado; Salida=0 significa que el segmento está desactivado; Salida=X significa indiferente

Decodificadores

- **Decodificador BCD a 7 segmentos** - (Floyd, 2000, p. 253-259)
 - Minimización mediante el mapa de Karnaugh de la lógica del segmento a:

		BA			
		00	01	11	10
DC	00	1		1	1
	01		1	1	1
	11	X	X	X	X
	10	1	1	X	X

- * **Suma de productos mínima de la lógica del segmento a:**

$$a = D + B + CA + \overline{C}\overline{A}$$

- * Hoja de Ejercicios 4. Ejercicio 7. Determinar las expresiones lógicas mínimas de los segmentos b, c, d, e, f y g y sus circuitos correspondientes incluido el circuito para el segmento a

Codificadores

- Circuitos lógicos combinacionales que, esencialmente, realizan la función "inversa" del decodificador - (Floyd, 2000, p. 360-365)
- Permiten que se introduzca en una de sus entradas un nivel activo que representa un dígito tales como un dígito decimal o un dígito octal, y lo convierte en una salida codificada, como BCD o binario
- **Estructura:**
 - **m** entradas de datos, activas en alta o baja
 - **n** salidas de datos, activas en alta o baja
- **Propósito:** proporcionan un código de salida (binario, gray, etc.) asociado a la línea de entrada activa (que sólo puede ser una al mismo tiempo)
- Si $m = 2^n$, el codificador se dice que es **completo**



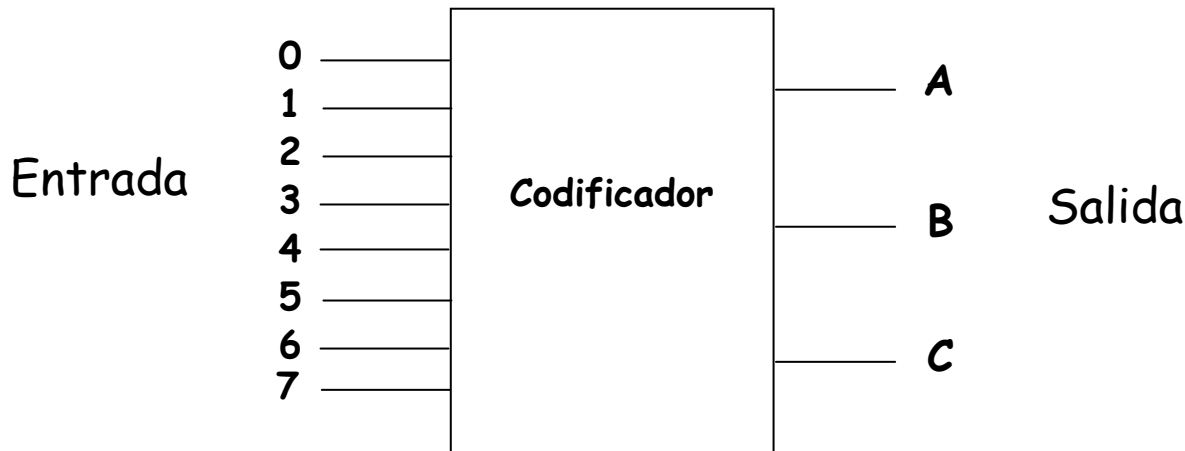
Codificadores

- Un codificador. Noción Básica

- Ejemplo: Si la entrada 4 está activada (es decir, si es la única entrada a nivel lógico 1 mientras que las otras entradas están a nivel 0), las salidas tendrán los siguientes valores:

$$C = 1, B = 0, A = 0 \text{ (} 4_{10} = 100_2 \text{)}$$

- Símbolo lógico:



Codificadores

• Codificador Binario

- Símbolo, tabla de verdad y ecuaciones de salida:

I_0	0	1	O_1	I_0	I_1	I_2	I_3	O_1	O_0
I_1	1			1	0	0	0	0	0
I_2	2	0	O_0	0	1	0	0	0	1
I_3	3			0	0	1	0	1	0
				0	0	0	1	1	1

$$O_0 = I_1 + I_3$$

$$O_1 = I_2 + I_3$$

• Codificador Gray

- Símbolo, tabla de verdad y ecuaciones de salida:

I_0	0	1	O_1	I_0	I_1	I_2	I_3	O_1	O_0
I_1	1			1	0	0	0	0	0
I_2	2	0	O_0	0	1	0	0	0	1
I_3	3			0	0	1	0	1	1
				0	0	0	1	1	0

$$O_0 = I_1 + I_2$$

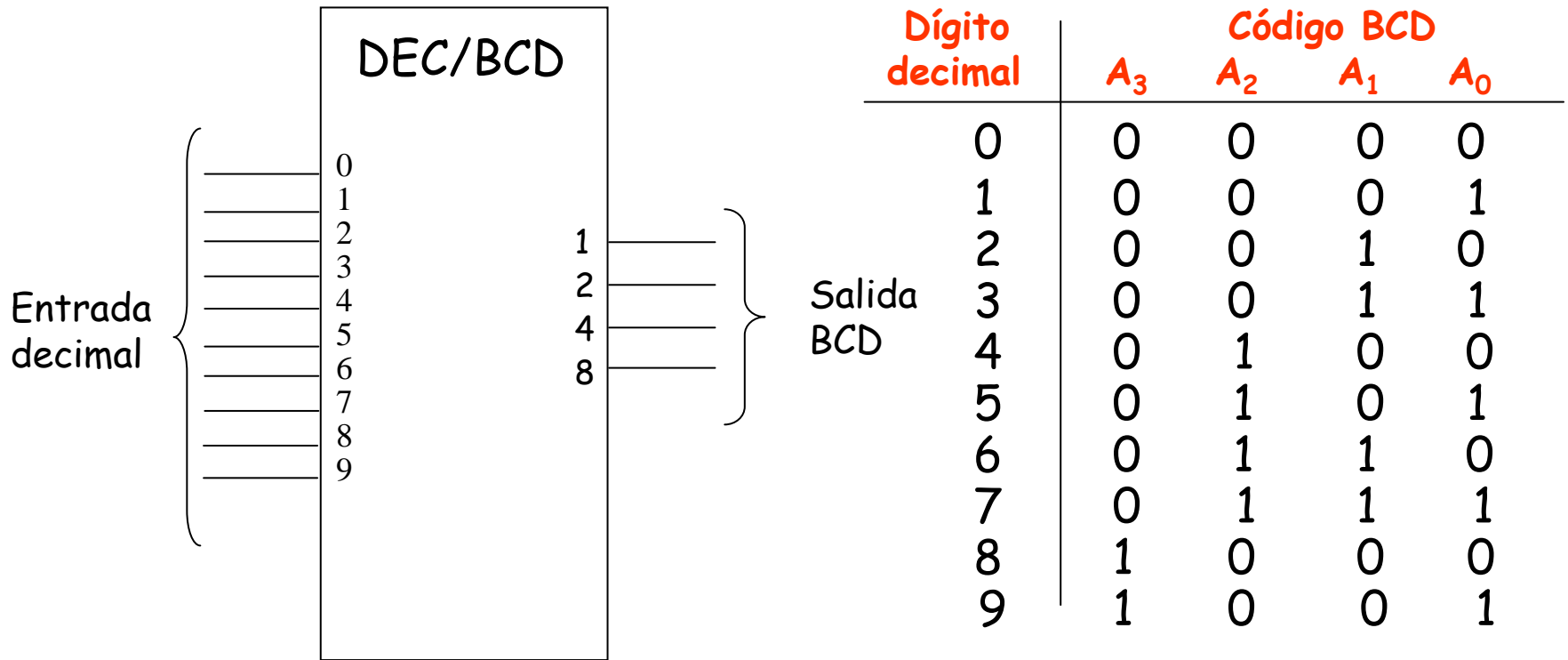
$$O_1 = I_2 + I_3$$



Codificadores

- Codificador decimal-BCD

- Símbolo lógico, tabla de verdad y ecuaciones de salida:



$$A_0 = 1 + 3 + 5 + 7 + 9 \quad A_1 = 2 + 3 + 6 + 7 \quad A_2 = 4 + 5 + 6 + 7 \quad A_3 = 8 + 9$$

Codificadores

- Codificador decimal-BCD

- Circuito:

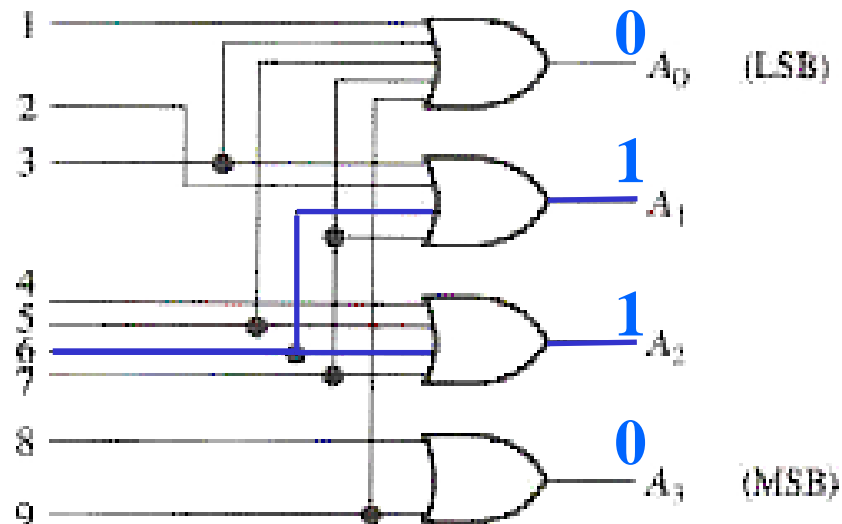


Diagrama lógico básico de un codificador decimal-BCD. No se necesita una entrada para el dígito 0, ya que las salidas BCD están todas a nivel BAJO cuando no hay entradas a nivel ALTO.

Codificadores

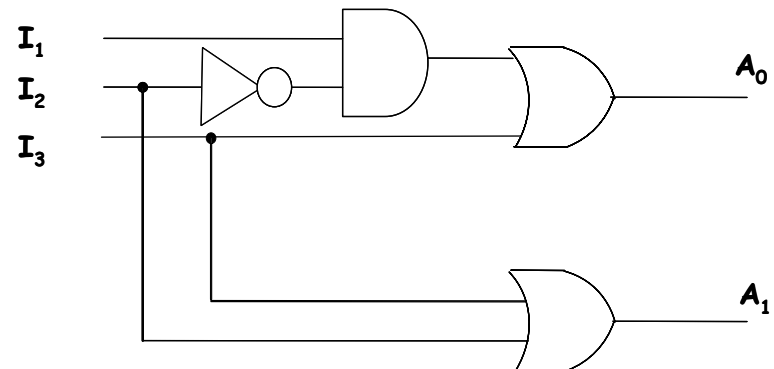
• Codificador de prioridad 4-2

- Permite que haya más de una línea de entrada activa simultáneamente
- Para decidir el valor que se generará en la salida se asignará un peso o prioridad a cada línea de entrada y la de mayor peso que se encuentre activa será la que determine la salida
- **Tabla de verdad, ecuaciones de salida y circuito:**

I_3	I_2	I_1	I_0	A_1	A_0
0	0	0	0	0	0
1	x	x	x	1	1
0	1	x	x	1	0
0	0	1	x	0	1
0	0	0	1	0	0

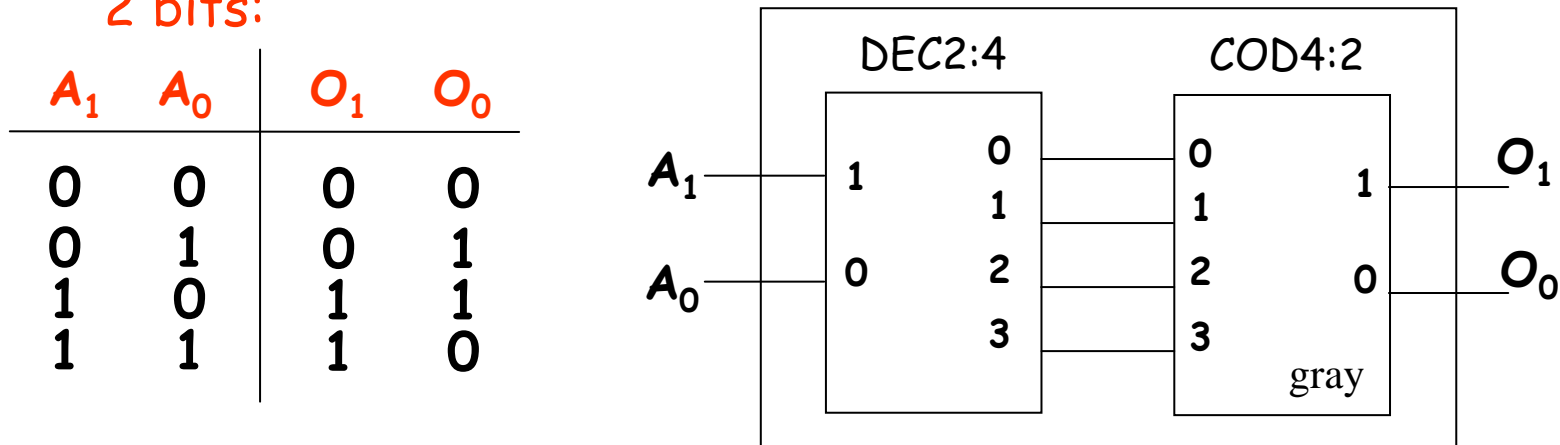
$$A_0 = (I_3 + \overline{I_2} I_1)$$

$$A_1 = (I_3 + I_2)$$



Conversores de código

- **Propósito:** transforman, traducen o convierten una palabra de m bits a la entrada de un código a una palabra de n bits a la salida de otro código
- Por ejemplo: a) de código binario a gray o viceversa; b) de código binario (BCD) a 7-segmentos
- Tabla de verdad y circuito para un convertidor binario a gray de 2 bits:



Nota: un convertidor de código puede implementarse conectando en serie un decodificador a un codificador

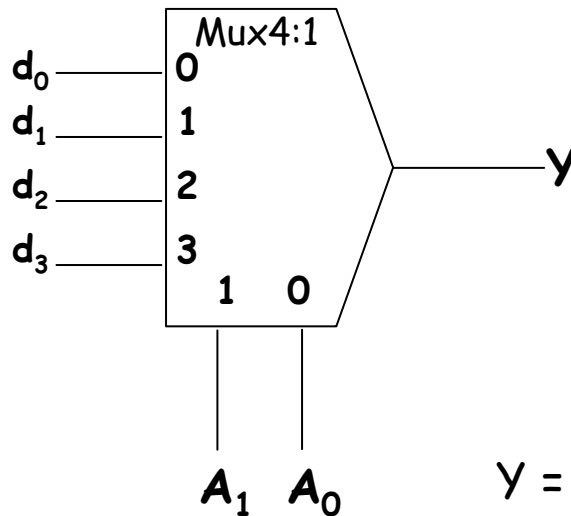
Multiplexores

- Permite dirigir la información digital procedente de diversas fuentes a una única línea para ser transmitida a través de dicha línea a un destino común - (Floyd, 2000, p. 370-379)
- El multiplexor básico posee varias líneas de entrada de datos y una única línea de salida. También posee entradas de selección de datos, que permite realizar la transmisión de datos desde una entrada seleccionable hacia una salida única. También se denomina Selector de Datos
- Estructura:
 - Entradas: $m=2^n$ líneas de datos ("canales", d_{m-1}, \dots, d_0)
 - Entradas: n líneas de selección de canal o líneas de control o líneas de dirección (A_{n-1}, \dots, A_0)
 - Salida: 1 línea de datos (Y)



Multiplexores

- Denominación: $MUX_{m:1}$, con m = número de canales (2^n)
- Propósito: transfiere el valor presente en el canal indicado por las líneas de selección a la salida Y (función inversa a la del demultiplexor)
- Símbolo lógico, tabla de verdad resumida y ecuación de salida:



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

$$Y = \overline{A_1} \overline{A_0} d_0 + \overline{A_1} A_0 d_1 + A_1 \overline{A_0} d_2 + A_1 A_0 d_3$$

Multiplexor/selector de datos de 1
salida y 4 entradas

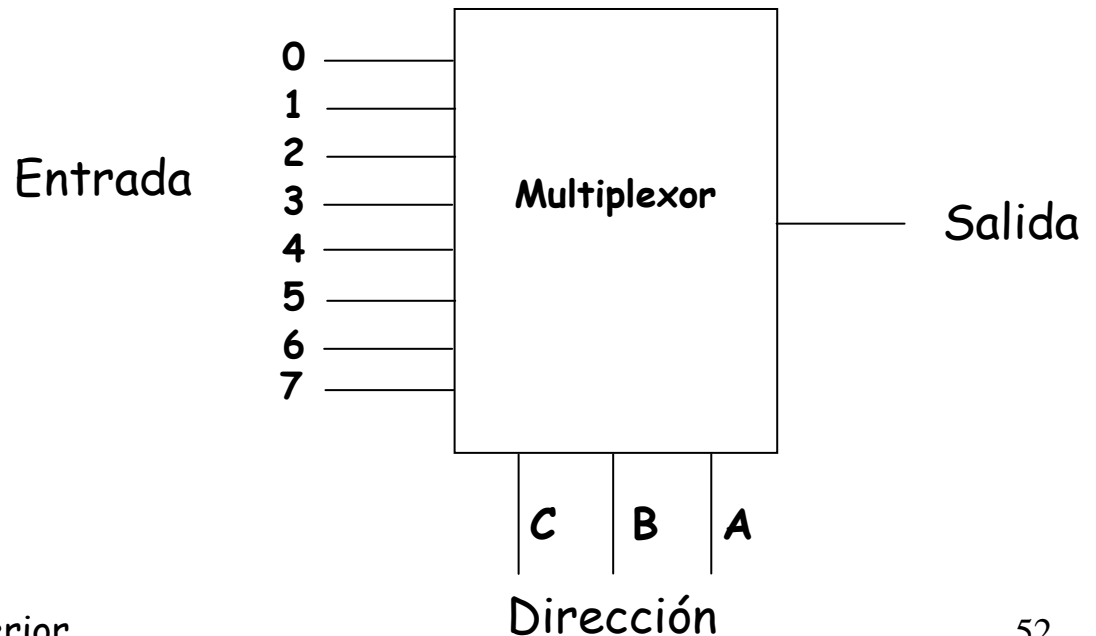
- * Ejercicio: Construir el circuito $MUX_{4:1}$ (Ver Figura 6.46, p. 371 de Floyd).

Multiplexores

- Un multiplexor

- Ejemplo: Si tenemos los valores $C = 1$, $B = 0$ y $A = 1$, en las entradas de dirección, encontraremos en la salida el mismo nivel lógico que el que esté presente en la línea de entrada número 5 ($101_2 = 5_{10}$)

- Símbolo lógico:

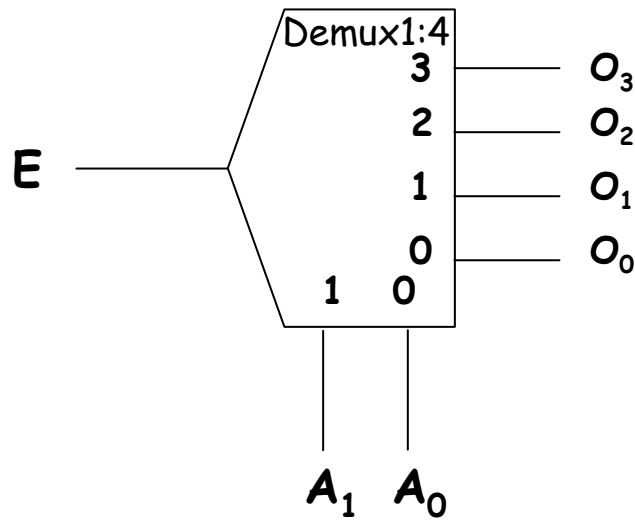


Demultiplexores

- Un demultiplexor toma datos de 1 línea y los distribuye a un determinado número de líneas de salida. Por ello, se denomina también distribuidor de datos - (Floyd, 2000, p. 379-381)
- Estructura:
 - Entradas: n líneas de selección de canal (A_{n-1}, \dots, A_0)
 - Entrada: 1 línea de datos (E)
 - Salidas: $m=2^n$ líneas de datos ("canales", O_{m-1}, \dots, O_0)
- Denominación: DEMUX1 : m , con m =número de canales (2^n)
- Propósito: transfiere el valor presente en la entrada E al canal de salida indicado por las líneas de selección (realiza la función inversa a la del multiplexor)

Demultiplexores

- Símbolo lógico y tabla de verdad:



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

Demultiplexor/distribuidor de
datos de 1 línea a 4 líneas

- * Ejercicio: Construir el circuito DEMUX1 : 4 (Ver Figura 6.54, p. 379 de Floyd).

Demultiplexores

- Un demultiplexor

- Ejemplo: Si tenemos los valores $C = 0$, $B = 1$ y $A = 0$, el nivel lógico en la línea de salida número 2 será el mismo que el de la línea de entrada
- Símbolo lógico:

