



Tema 3. 2

Sistemas Combinacionales



Índice

- Circuitos combinacionales: concepto, análisis y síntesis.
- Métodos de simplificación de funciones lógicas.
- Estructuras combinacionales básicas
 - Multiplexores
 - Demultiplexores
 - Decodificadores
 - Codificadores
 - Comparadores

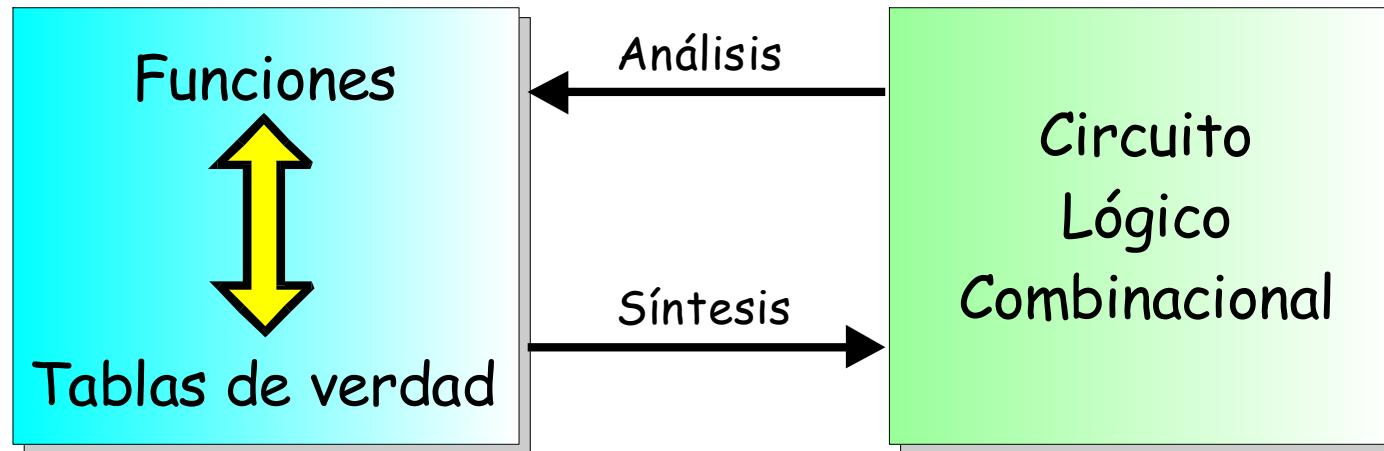
Bibliografía

- Fundamentos de sistemas digitales.
Thomas Floyd.
Prentice-Hall.
- Fundamentos de diseño lógico y computadoras.
M. Morris Mano.
Prentice-Hall

Circuitos combinacionales (I)



- **Concepto:** son aquellos circuitos cuyas salidas, en un determinado instante, son función exclusivamente del valor de las entradas en ese instante.



- **Análisis:** Obtención de la función de salida de un circuito, para cada una de las posibles combinaciones de las “n” variables de entrada.
- **Síntesis:** Obtención del circuito que realiza la función especificada.

Circuitos combinacionales (II)



Los circuitos combinacionales pueden ser:

- Sistemas *unifuncionales*: tienen una sola función de salida
- Sistemas *multifuncionales*: tienen varias funciones de salida

Una función puede ser:

- *Completa*. Su valor está determinado para todas las posibles combinaciones de las variables de entrada.
- *Incompleta*. Existen algunas combinaciones de entrada para las cuales el valor de la función es indeterminado.

Causas:

- Existencia de combinaciones de las variables que nunca se presentan.
- Existencia de combinaciones de las variables para las que el valor que tome la función sea indiferente.



Síntesis de circuitos combinacionales



Proceso a seguir para obtener un circuito combinacional óptimo:

- Establecer la tabla de verdad, desde el enunciado del problema.
- Obtener la función canónica expresada en minterms o en maxterms, a partir de la tabla de verdad.
- Simplificar la función canónica, bien en forma algebraica (aplicando teoremas y postulados del Álgebra de Boole), bien mediante la aplicación de métodos gráficos sencillos (Karnaugh) o con el método tabular numérico de Quine-McCluskey.
- Realizar la función simplificada, mediante las oportunas puertas lógicas.



Métodos de simplificación de funciones lógicas (I)



Método algebraico

- Es el método básico de simplificación de funciones y consiste en aplicar directamente la propiedad distributiva a los términos de la función, eliminando variables. Por ejemplo:

$$f_1(d,c,b,a) = d \cdot c \cdot b \cdot a + d \cdot c \cdot b \cdot \bar{a} = d \cdot c \cdot b \cdot (a + \bar{a}) = d \cdot c \cdot b \cdot 1 = d \cdot c \cdot b$$

$$f_2(d,c,b,a) = (d+c+b+a) \cdot (d+c+\bar{b}+a) = (d+c+b \cdot \bar{b}+a) = (d+c+a)$$

- Sin embargo, pocas veces viene expresada la función de forma que sea fácilmente aplicable este método.



Métodos de simplificación de funciones lógicas (II)



Método de Karnaugh

- Método tabular gráfico que se basa en los llamados “mapas de Karnaugh”, consistentes en una tabla de cuadros, cada uno de los cuales representa un término canónico.

Estos cuadros están distribuidos de tal modo que dos cualesquiera de ellos, contiguos físicamente, corresponden a términos canónicos adyacentes.

- ***Términos canónicos adyacentes***: son aquellos para los que sus respectivas configuraciones binarias difieren entre sí en un único bit. Se pueden definir también como aquellos términos a los que se les puede aplicar la propiedad distributiva para simplificar una variable.



Métodos de simplificación de funciones lógicas (III)



- Mapa de Karnaugh para funciones de **dos** variables

		a	
		0	1
b	0	0	1
	1	2	3

		b a			
		00	01	11	10
b	0	1	3	2	

Tres variables

		b a			
		00	01	11	10
c	0	0	1	3	2
	1	4	5	7	6

Cuatro variables

		b a			
		00	01	11	10
d c	00	0	1	3	2
	01	4	5	7	6
	11	12	13	15	14
	10	8	9	11	10

Métodos de simplificación de funciones lógicas (IV)



Procedimiento de simplificación mediante los mapas de Karnaugh

- 1.- Dibujar el mapa adecuado para la función a simplificar (2, 3 ó 4 variables).
- 2.- Marcar los cuadros correspondientes a los minterms o maxterms ("1").
- 3.- Agrupar, mediante una curva cerrada, el máximo número posible de elementos adyacentes (potencia de 2 \Rightarrow 2, 4, 8, 16). A continuación lo mismo con los que queden y así, sucesivamente, hasta que no reste ningún elemento marcado (con adyacente también marcado) sin agrupar.
- 4.- Escribir la función mediante los términos simplificados obtenidos.

Criterio: en cada grupo desaparece la variable o variables cuyo valor es "0" en la mitad de los cuadros del grupo, y "1" en la otra mitad. Las variables que permanecen son tomadas como "no negadas" si su valor es 1 en todo el grupo de cuadros, y como "negadas" si su valor es 0.

Si algún bit no tiene ninguno adyacente seguirá como término canónico

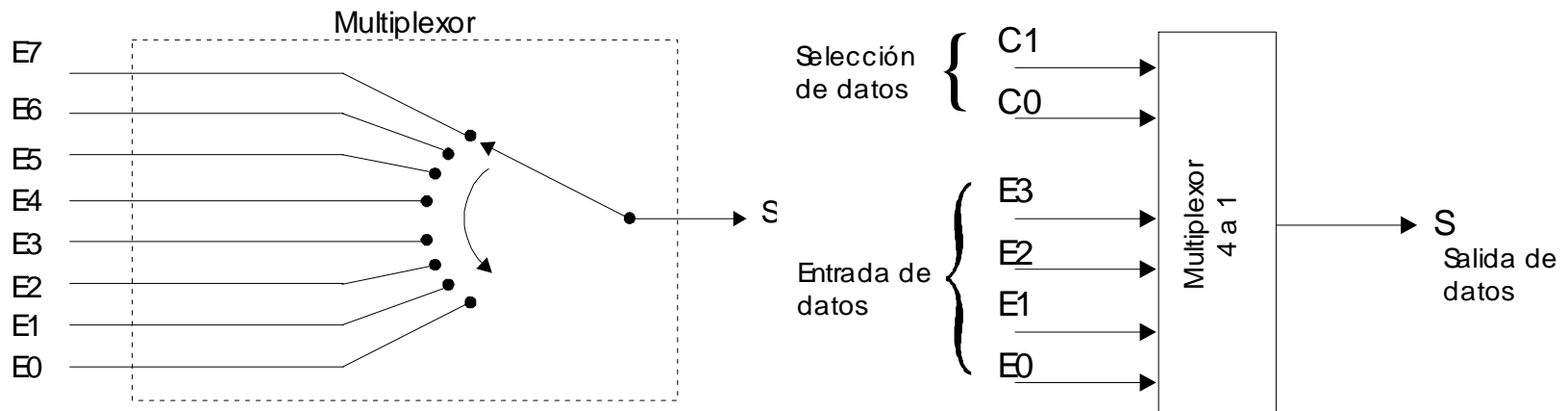


Estructuras Combinacionales básicas (I)

Multiplexores



- Un multiplexor es un circuito que tiene 2^n entradas de información (canales), una sola salida y un mecanismo de selección que determina cuál de las entradas es la que se transfiere su información a la única salida. Se comporta como un conmutador de entrada múltiple y salida única, pero cuyo control no es mecánico, sino electrónico.

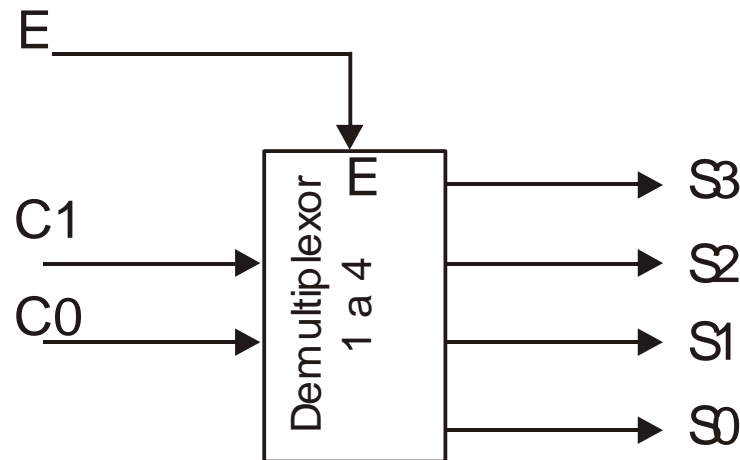


Estructuras Combinacionales básicas (II)

Demultiplexores



- Realizan la función inversa de un multiplexor, es decir, se comportan como conmutadores de entrada única y salida múltiple, existiendo un mecanismo de control que selecciona la salida hacia la que se envía la información de entrada.
- En general un demultiplexor tiene una única entrada de información, 2^n salidas y “n” entradas de control en las que se introduce el número binario correspondiente a la salida seleccionada.

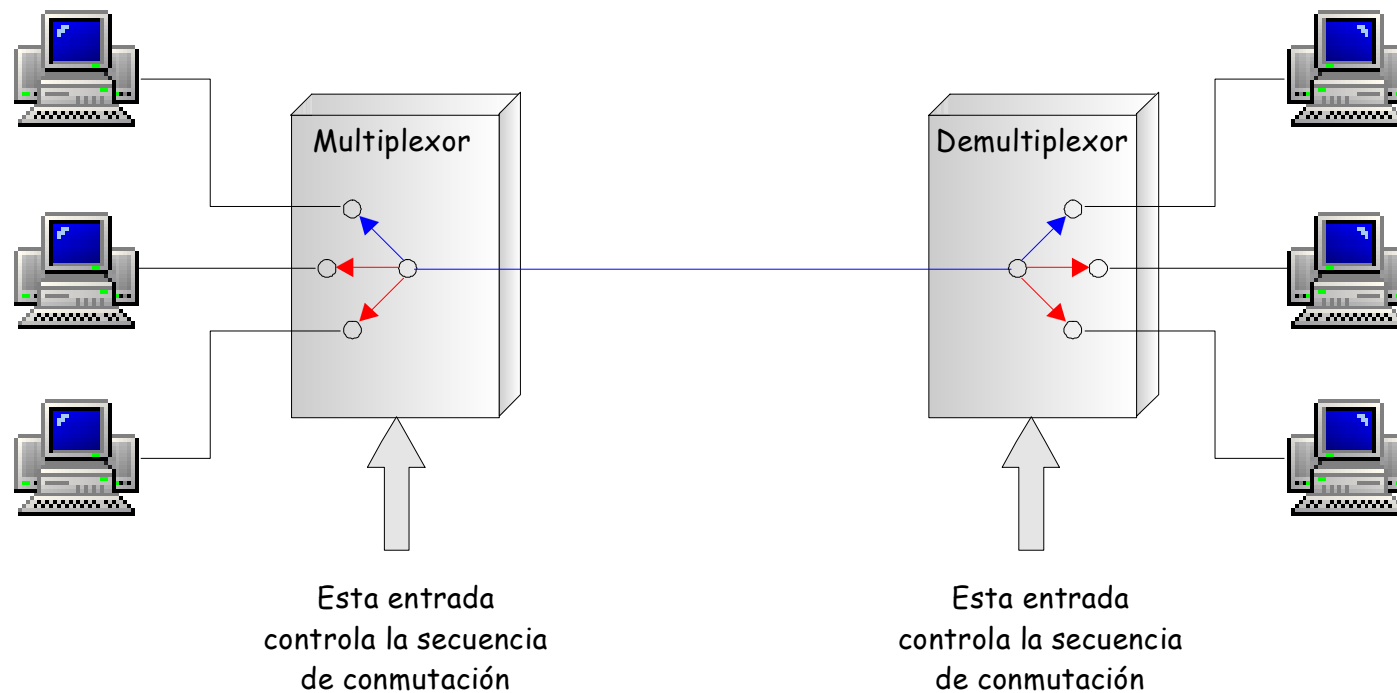


Estructuras Combinacionales básicas (III)

Multiplexores /Demultiplexores



Aplicaciones básicas



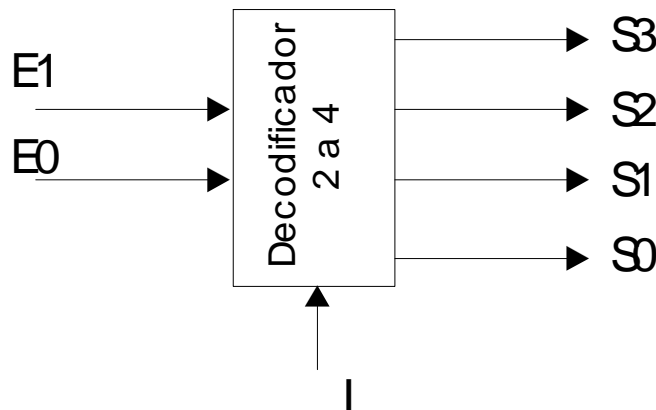
Estructuras Combinacionales básicas (IV)

Decodificadores (1)



- Son circuitos digitales que tienen como entrada (n bits) la información codificada en binario, y tantas salidas como posibles configuraciones binarias distintas de entrada (2^n), activándose en cada momento una sola de ellas, la correspondiente a la combinación binaria aplicada a la entrada.

Decodificador 2 a 4



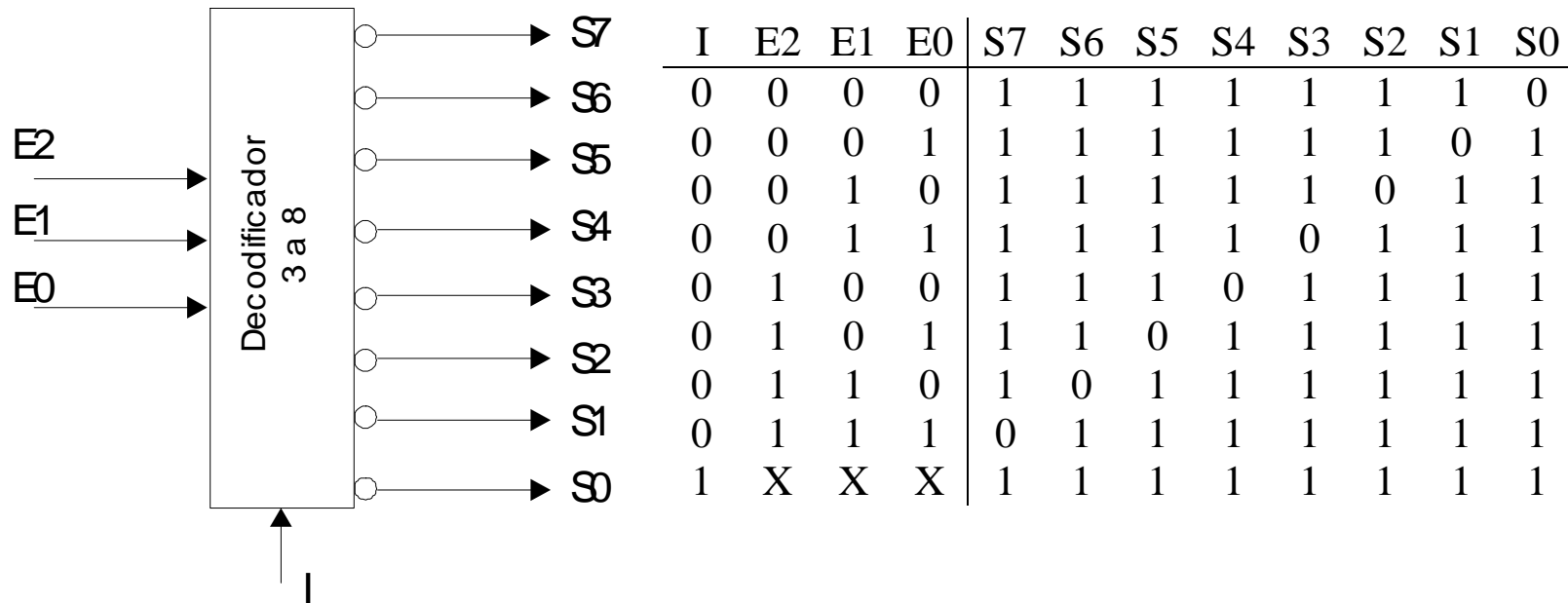
I	E1	E0	S3	S2	S1	S0
0	0	0	0	0	0	1
0	0	1	0	0	1	0
0	1	0	0	1	0	0
0	1	1	1	0	0	0
1	X	X	0	0	0	0

Estructuras Combinacionales básicas (V)

Decodificadores (2)



Decodificador 3 a 8

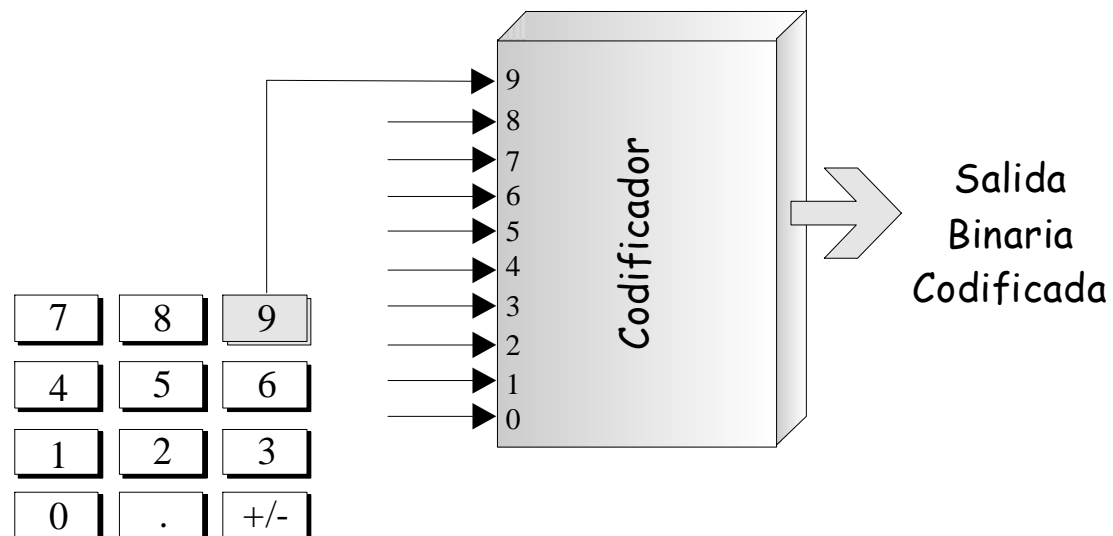


Estructuras Combinacionales básicas (VI)

Codificadores (1)



- Realizan la función inversa de los decodificadores, es decir, poseen en general N entradas y “ n ” salidas de código en las que aparece codificado en binario el valor de la entrada que ha sido activada ($N \leq 2^n$).
 - Codificadores sin prioridad.
 - Codificadores con prioridad.

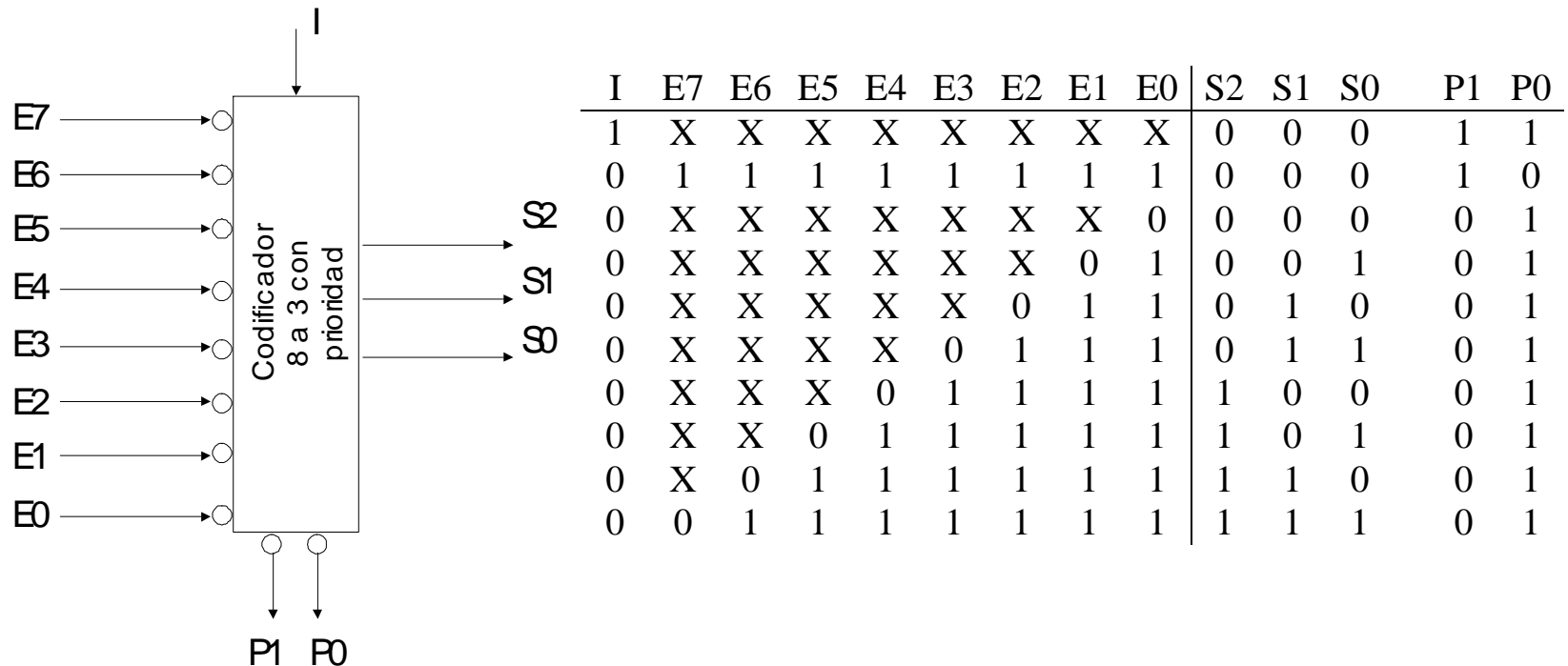


Estructuras Combinacionales básicas (VII)

Codificadores (2)



- Codificador 8 a 3 con prioridad

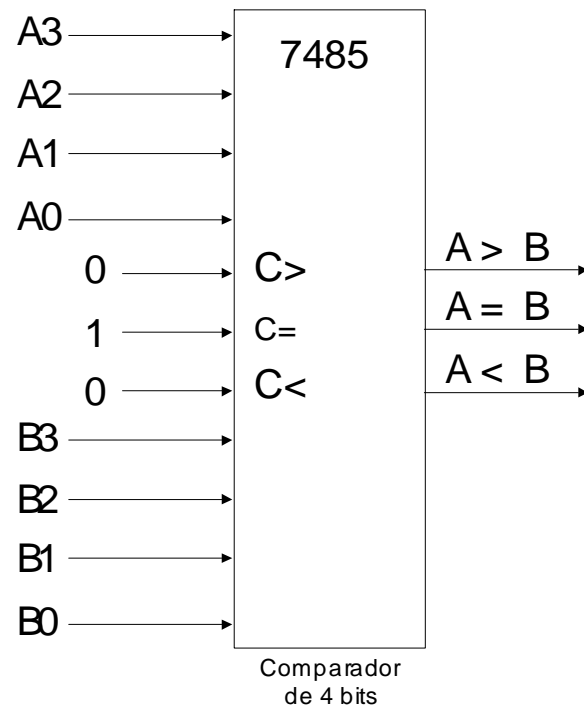


Estructuras Combinacionales básicas (VIII)

Comparadores



- Un comparador de “n” bits es un circuito combinacional que tiene por entradas dos números binarios de “n” bits cada uno (A,B), determinando a su salida si uno es mayor, menor o igual que el otro. Normalmente dispone de tres salidas: $A > B$, $A = B$ y $A < B$



Ay B	C>	C=	C<	A>B	A=B	A<B
$A > B$	X	X	X	1	0	0
$A < B$	X	X	X	0	0	1
$A = B$	0	0	1	0	0	1
$A = B$	0	1	0	0	1	0
$A = B$	1	0	0	1	0	0

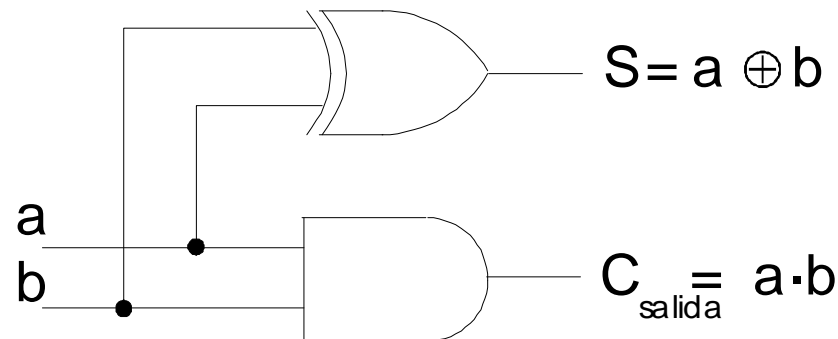
Circuitos sumadores binarios (I)



Semisumador

Suma de dos únicos bits. Proporciona como salidas la suma y el acarreo.

Entradas		Salidas	
Sumando a	Sumando b	Suma S	Acarreo C_{salida}
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1



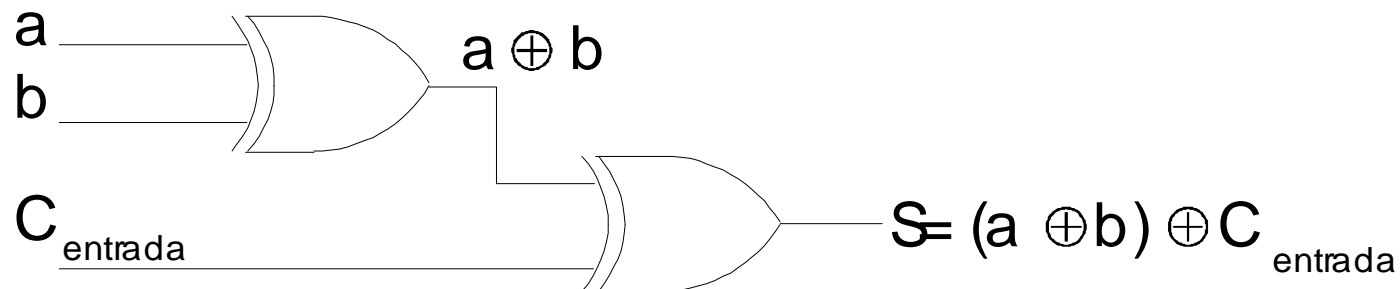
Circuitos sumadores binarios (II)



Sumador completo

Suma de tres bits: 2 bits i-ésimos más el acarreo

Entradas			Salidas	
Sumando a	Sumando b	Acarreo C _{entrada}	Suma S	Acarreo C _{salida}
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

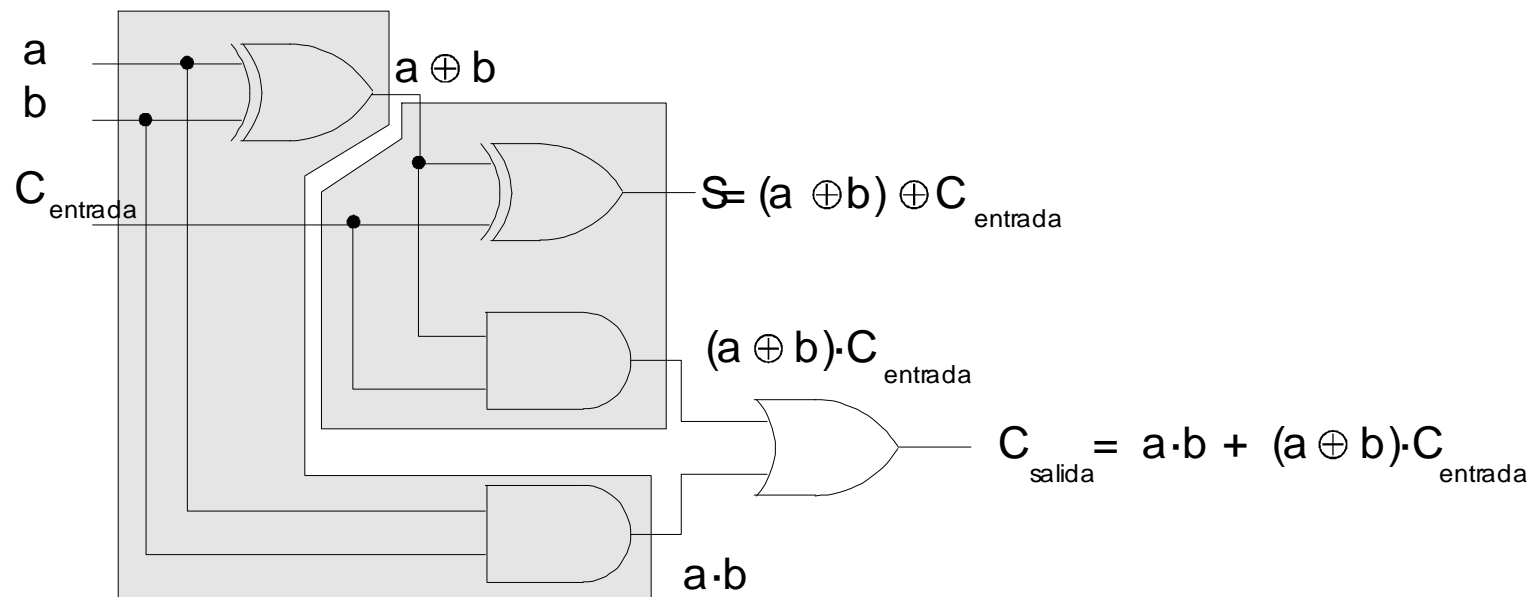


Circuitos sumadores binarios (III)



Sumador completo

Circuito en función de semisumadores

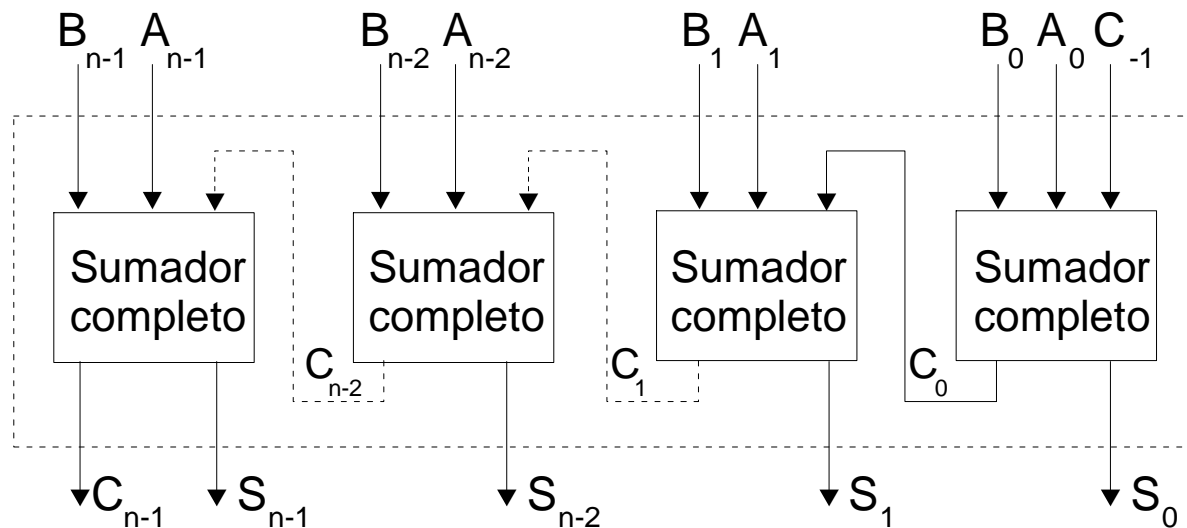


Circuitos sumadores binarios (IV)



El acarreo en los circuitos sumadores

- Sumador paralelo con acarreo serie



- Sumador paralelo con acarreo paralelo (acarreo anticipado)

$$C_i = A_i B_i + (A_i \oplus B_i) C_{i-1} = G_i + P_i C_{i-1}$$

- Generación de acarreo de la propia etapa $G_i = A_i B_i$
- Propagación del acarreo de la etapa anterior $P_i = A_i \oplus B_i$
- Sumador en paralelo con acarreo mixto

Circuitos sumadores/restadores



Operaciones con signo (Circuito Sumador/Restador)

- En complemento a uno
- En complemento a dos

