Tema 3: **Circuitos Combinacionales**

Contenido

Objetivos

- Estudiar el comportamiento y la estructura interna de los dispositivos lógicos más sencillos
- Diseñar sistemas combinacionales complejos a partir de sistemas básicos
- Aplicar los conocimientos adquiridos para abordar la resolución de problemas reales mediante el diseño de sistemas digitales de control.

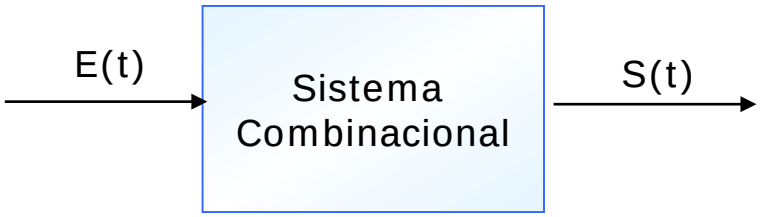
Bibliografía

- Capítulo 6 del libro de “Fundamentos de Sistemas Digitales” (7ª Edición). Thomas L. Floyd. Ed. Prentice Hall

1. Introducción

Definición de sistema combinacional

- Es un conjunto de dispositivos lógicos en el que la(s) salida(s) dependen exclusivamente del valor actual de la(s) entrada(s).



- A todo sistema combinacional le corresponde una función lógica.
- Sistemas combinacionales integrados: en un CI se incluyen uno o varios sistemas combinacionales.
- Criterio de simplificación:
  En un sistema combinacional se busca la menor cantidad de puertas lógicas básicas.

2. Circuitos Aritméticos.

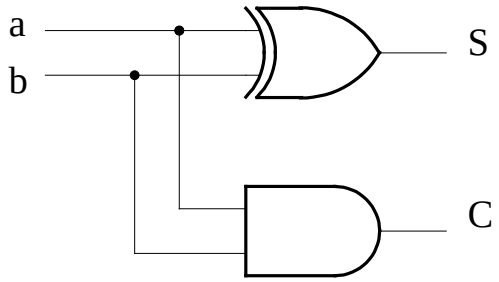
2.1.1 Semisumador

Es un circuito combinacional capaz de sumar dos dígitos binarios, proporcionando como salidas la suma y el posible acarreo

a	b	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

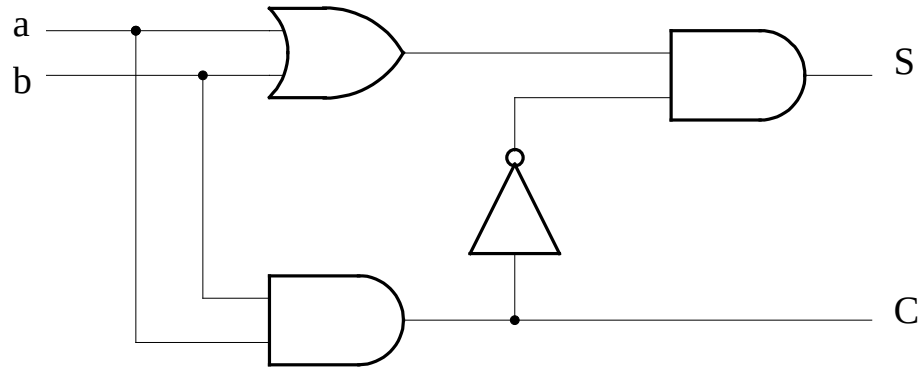
$$S = \bar{a}b + a\bar{b} = a \oplus b$$

$$C = ab$$

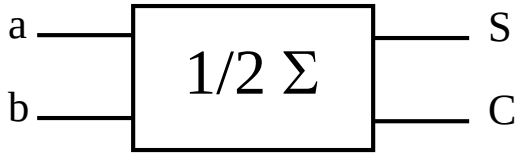


Si operamos esta expresión se transforma en:

$$S = a \oplus b = (a+b)\overline{ab} = (a+b)\bar{C}$$



Expresándolo como un bloque funcional:



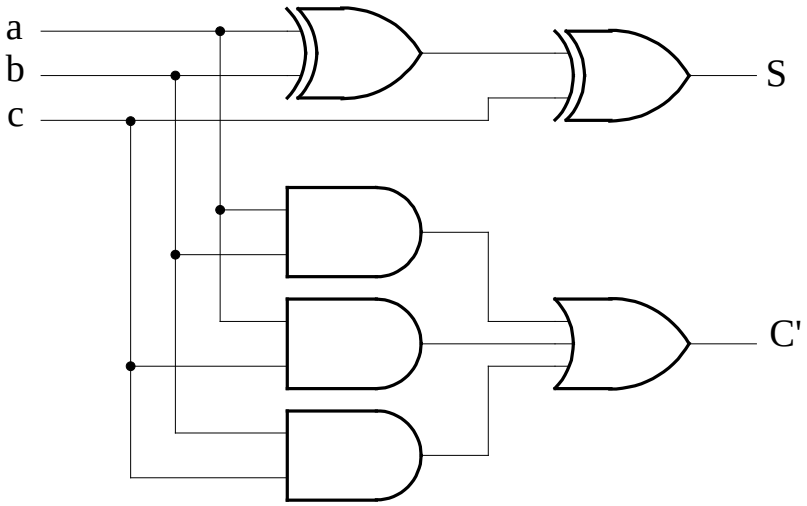
2.1.2 Sumador Completo

Es un circuito combinacional capaz de sumar dos dígitos binarios junto con el posible acarreo procedente de la etapa anterior y proporcionando como salidas la suma y el acarreo producido

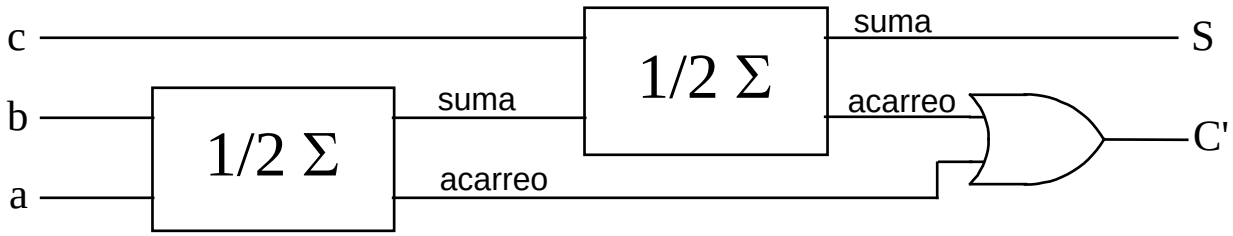
a	b	c	S	C'
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

$$S = \bar{a}\bar{b}c + \bar{a}b\bar{c} + a\bar{b}\bar{c} + abc = a \oplus b \oplus c$$

$$C' = ab + ac + bc$$

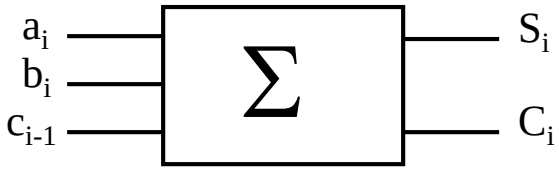


Mediante semisumadores:

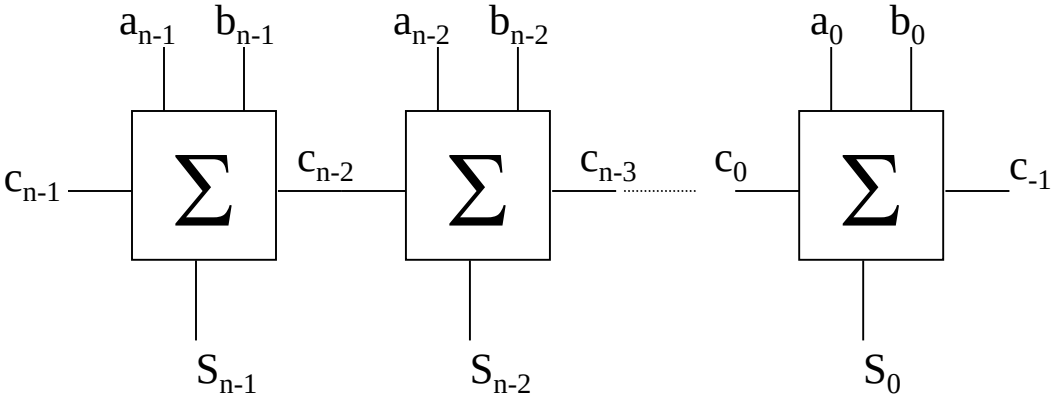


2.1.3 Sumador binario en paralelo

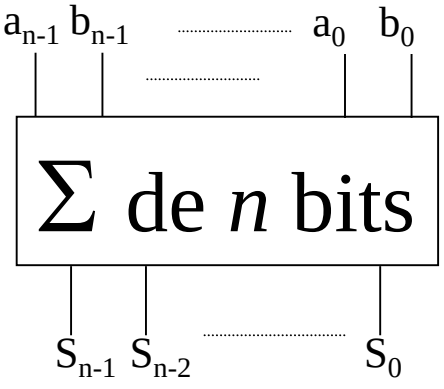
Bloque Funcional del sumador completo de 1 bit:



Un sumador en paralelo de ***n*** bits con bloques funcionales:



Bloque funcional del sumador paralelo de ***n*** bits

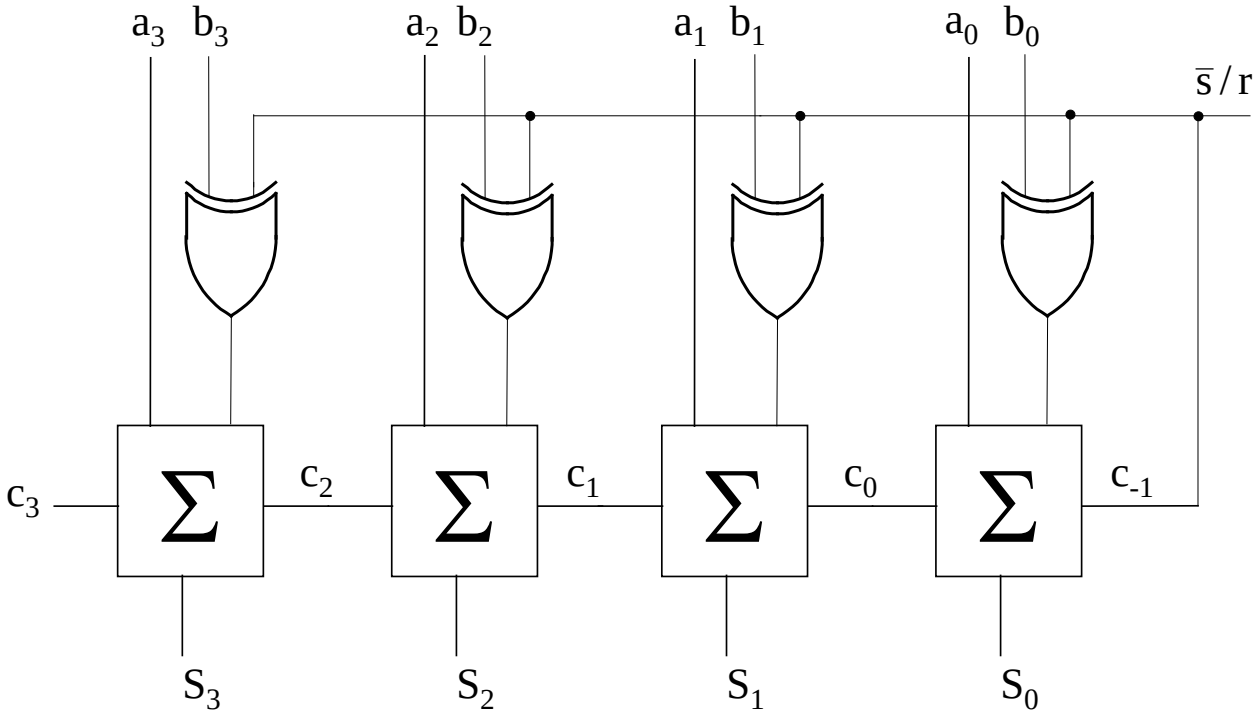


2.3 Sumador / Restador binario

$\bar{s}/r$	b	b’
0	0	0
0	1	1
1	0	1
1	1	0

$$b' = (\bar{s}/r) \oplus b$$

En complemento a 2:

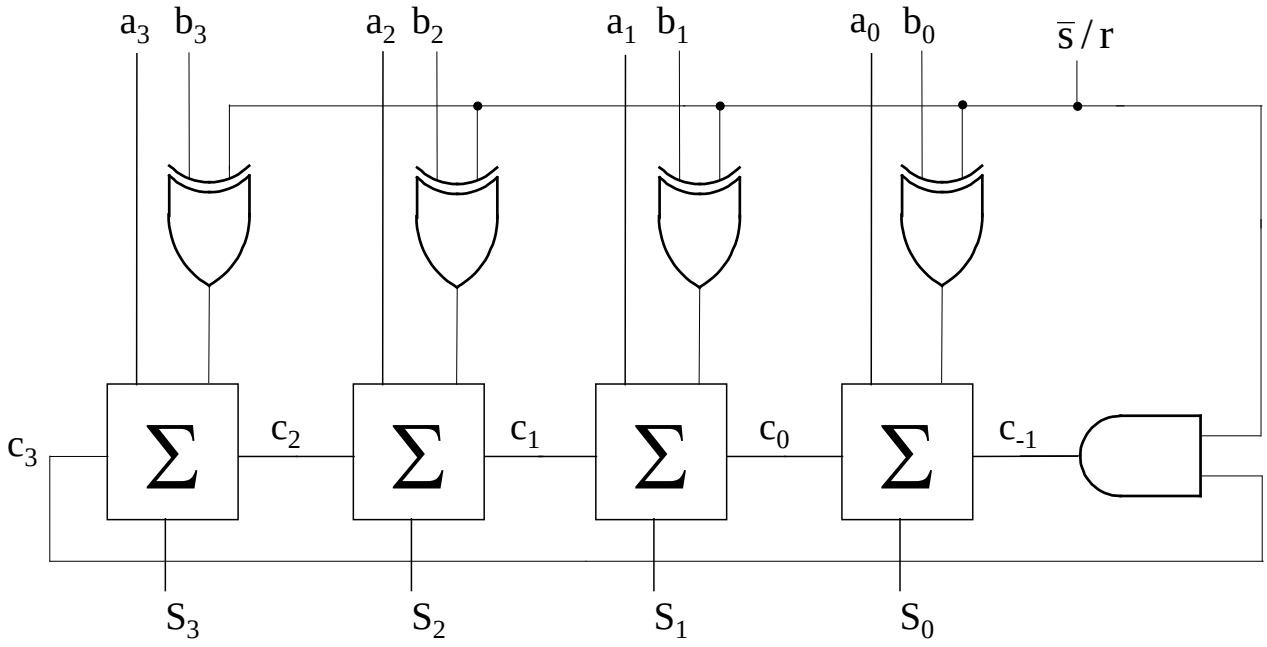


2.3 Sumador / Restador binario

$\bar{s}/r$	b	b’
0	0	0
0	1	1
1	0	1
1	1	0

$$b' = (\bar{s}/r) \oplus b$$

En complemento a 1:



3 Decodificadores

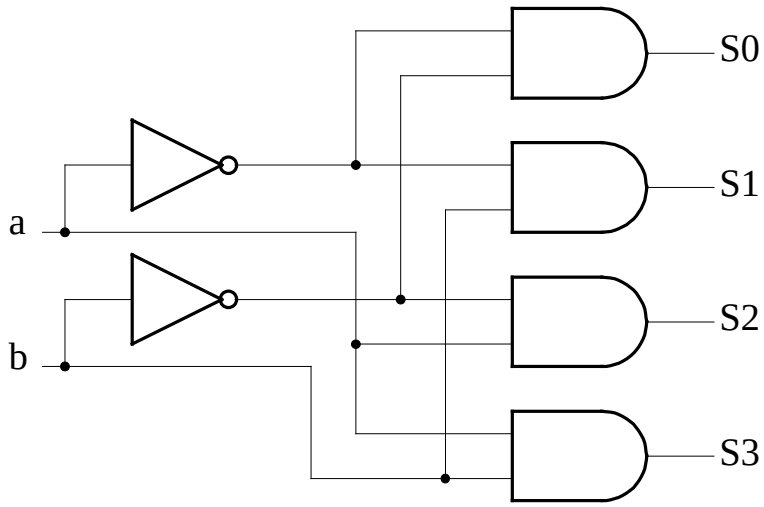
Podemos definir al **Decodificador** como un circuito combinacional que consta de *n* entradas y 2^n salidas como máximo. Este circuito pone a uno la salida cuyo índice coincide con la combinación binaria presente en las entradas.

Un decodificador de 2 a 4 líneas presenta la siguiente tabla de verdad:

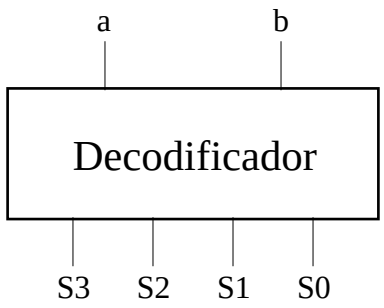
a	b	S_0	S_1	S_2	S_3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

$$S_0 = \bar{a}\bar{b}$$
$$S_1 = \bar{a}b$$
$$S_2 = a\bar{b}$$
$$S_3 = ab$$

Esquema:

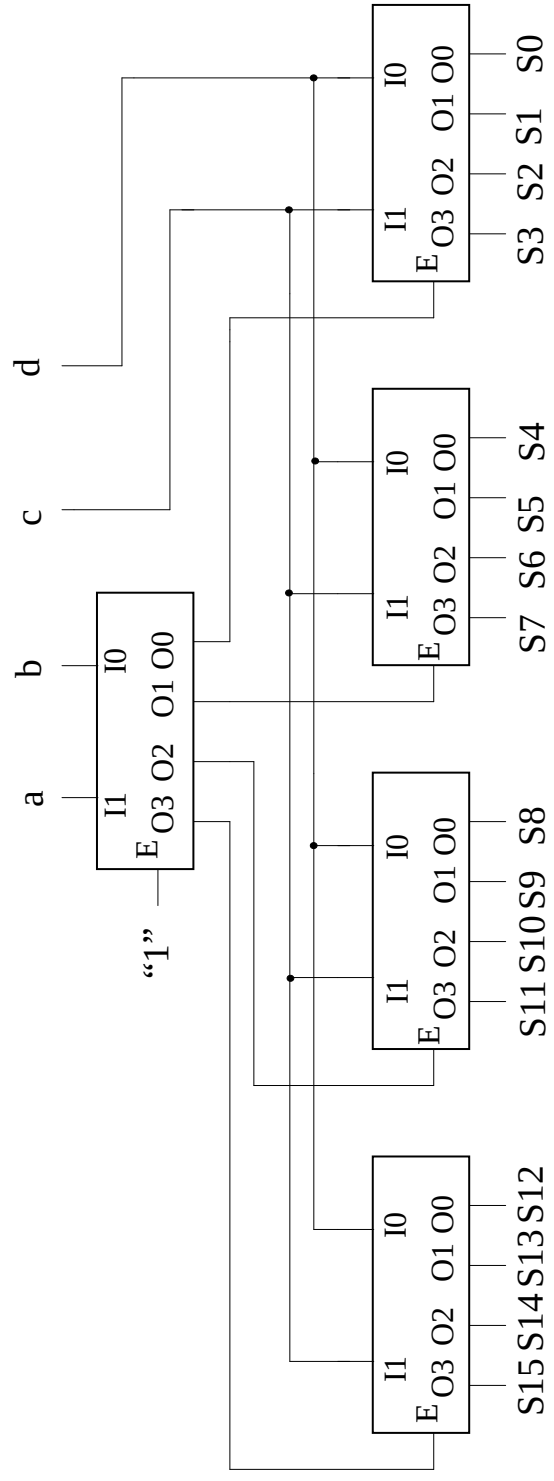


Bloque Funcional:



3 Decodificadores

Agrupación de Decodificadores: Decodificador de 4 a 16 Líneas mediante la entrada de habilitación E



3 Decodificadores

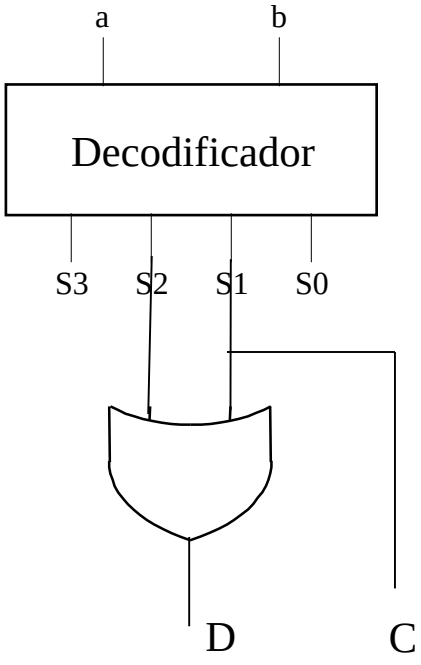
Implementación de funciones lógicas

Puede utilizarse el **decodificador** de n-entradas como circuito para implementar funciones lógicas de n variables lógicas. La idea se basa en agrupar aquellas combinaciones de las entradas que produzcan un valor lógico 1 en la salida.

a	b	D	C
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

$$C = \sum_{2}(1)$$

$$D = \sum_{2}(1,2)$$



4 Codificadores

Podemos definir al **Codificador** como un circuito combinacional que consta de 2^n entradas y *n* salidas. Este circuito coloca en sus salidas la combinación binaria correspondiente al índice de la entrada activada.

TIPOS: Prioritarios/No Prioritarios

EJEMPLO

Un Codificador Decimal-BCD (No prioritario) tiene la siguiente tabla de verdad:

E0	E1	E2	E3	E4	E5	E6	E7	E8	E9	DCBA	NA
1	0	0	0	0	0	0	0	0	0	0000	0
0	1	0	0	0	0	0	0	0	0	0001	0
0	0	1	0	0	0	0	0	0	0	0010	0
0	0	0	1	0	0	0	0	0	0	0011	0
0	0	0	0	1	0	0	0	0	0	0100	0
0	0	0	0	0	1	0	0	0	0	0101	0
0	0	0	0	0	0	1	0	0	0	0110	0
0	0	0	0	0	0	0	1	0	0	0111	0
0	0	0	0	0	0	0	0	1	0	1000	0
0	0	0	0	0	0	0	0	0	1	1001	0
0	0	0	0	0	0	0	0	0	0	0000	1

$$D = E8 + E9$$

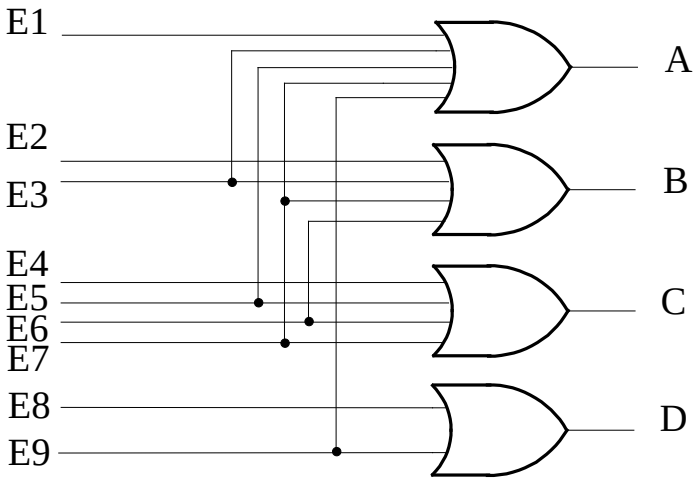
$$C = E4 + E5 + E6 + E6$$

$$B = E2 + E3 + E6 + E7$$

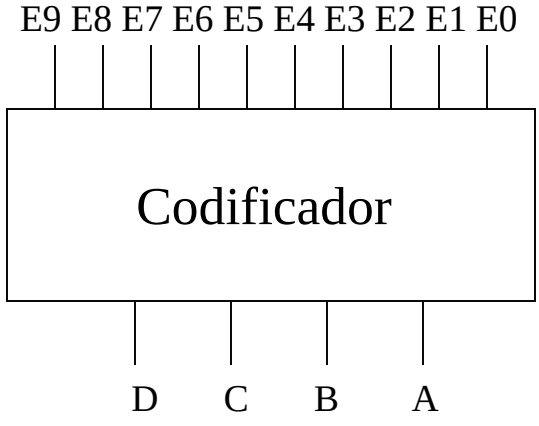
$$A = E1 + E3 + E5 + E7 + E9$$

4 Codificadores

Esquema:



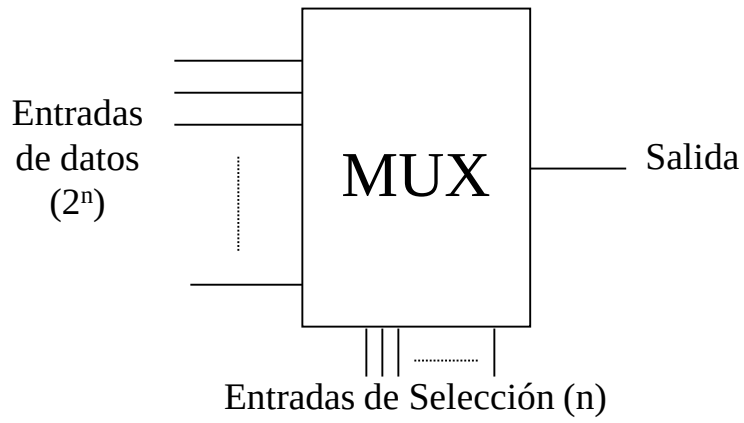
Bloque funcional:



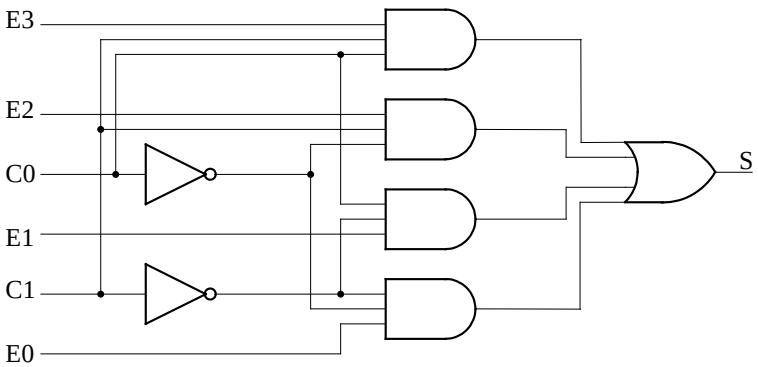
5 Multiplexores (selectores de datos)

Podemos definir al **Multiplexor** como un circuito combinacional que consta de 2^n entradas de datos, *n* entradas de selección y una salida. Este circuito coloca en su salida el valor de la entrada cuyo índice coincide con la combinación binaria presente en las entradas de selección.

Esquema:

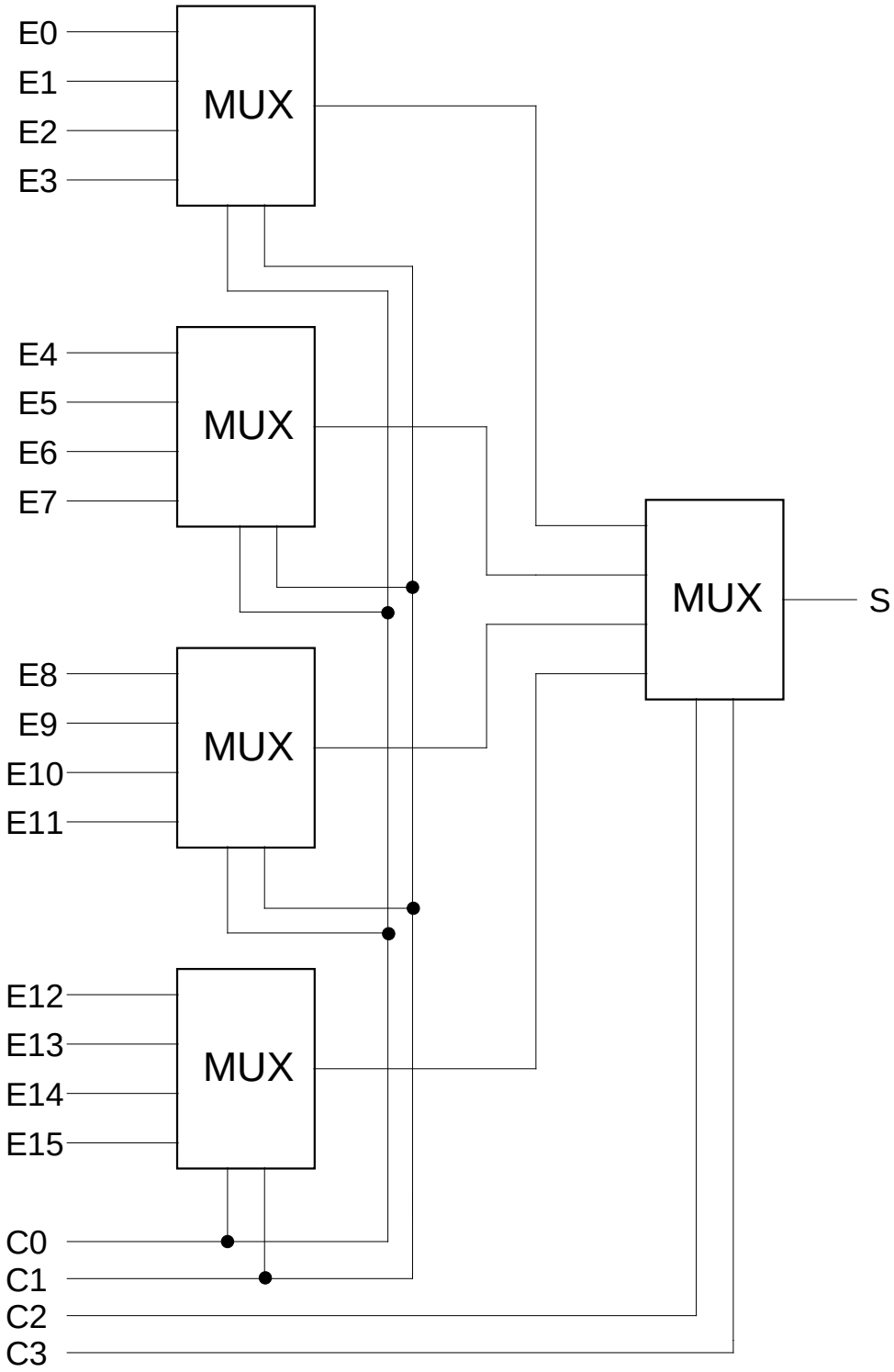


C1	C0	E3	E2	E1	E0	S
0	0	X	X	X	0	0
0	0	X	X	X	1	1
0	1	X	X	0	X	0
0	1	X	X	1	X	1
1	0	X	0	X	X	0
1	0	X	1	X	X	1
1	1	0	X	X	X	0
1	1	1	X	X	X	1



$$S = \overline{C0}\,\overline{C1}E0 + \overline{C1}C0E1 + C1\overline{C0}E2 + C1C0E3$$

5 Multiplexores (asociación)

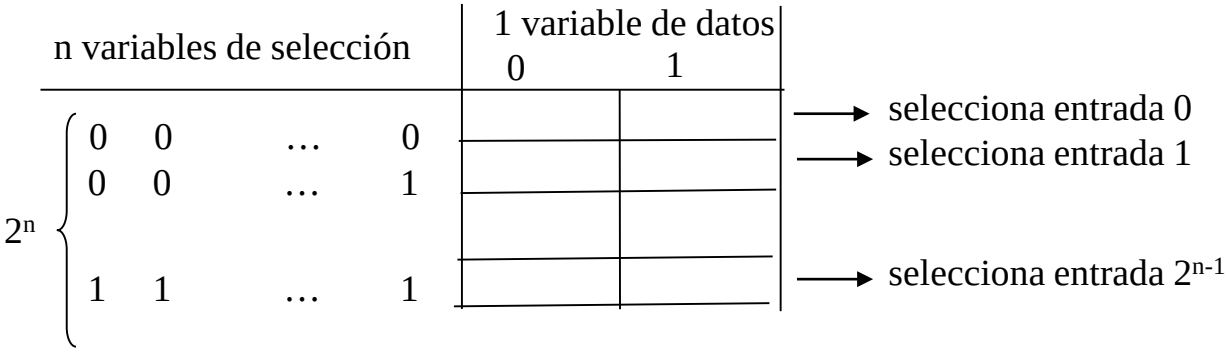


5 Multiplexores (selectores de datos)

Implementación de funciones lógicas

Puede utilizarse el **multiplexor** de n-entradas de selección como circuito para implementar funciones lógicas de n+1 variables lógicas. La idea consiste en asociar a las entradas de selección las variables lógicas exceptuando una de ellas que se utiliza para configurar la entrada de datos.

Se construye una tabla de verdad organizada de la siguiente manera:



Casos posibles para cada fila en la variable de datos:

Combinación en la entrada i	0	0	→ Siempre 0 en E_i
Combinación en la entrada i	0	1	→ variable_datos en E_i
Combinación en la entrada i	1	0	→ $\overline{\text{variable\_datos}}$ en E_i
Combinación en la entrada i	1	1	→ Siempre 1 en E_i

5 Multiplexores (selectores de datos)

Implementación de funciones lógicas: Ejemplo

a	b	D	C
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

$$C = \sum_{2}(1)$$

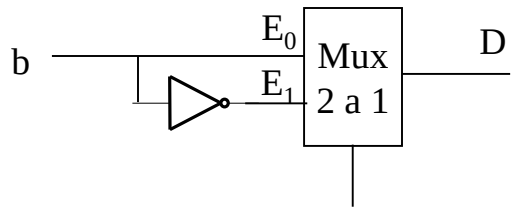
$$D = \sum_{2}(1,2)$$

2 variables como: 1 variable de selección (n-1) + 1 variable de datos
a = variable de selección ; b = variable de datos

Implementación de D

a \ b	0	1
0	0	1
1	1	0

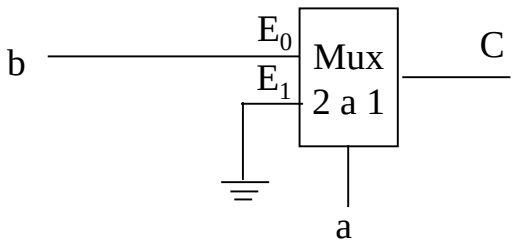
→ selecciona entrada 0 → $E_0=b$
→ selecciona entrada 1 → $E_1=\overline{b}$



Implementación de C

a \ b	0	1
0	0	1
1	0	0

→ selecciona entrada 0 → $E_0=b$
→ selecciona entrada 1 → $E_1=0$

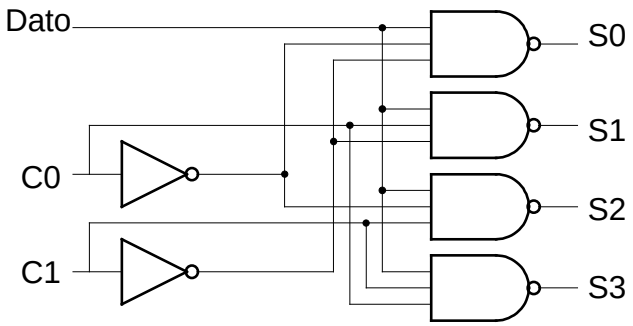


6 Demultiplexores (distribuidores)

Podemos definir al **Demultiplexor** como un circuito combinacional que consta de 1 entrada de datos, *n* entradas de selección y 2^n salidas. Este circuito coloca en la salida cuyo índice coincide con la combinación binaria presente en las entradas de selección el valor de la entrada de datos.

Ejemplo: Demultiplexor de 1 a 4

E	C1	C0	S3	S2	S1	S0
0	0	0	0	0	0	0
1	0	0	0	0	0	1
0	0	1	0	0	0	0
1	0	1	0	0	1	0
0	1	0	0	0	0	0
1	1	0	0	1	0	0
0	1	1	0	0	0	0
1	1	1	1	0	0	0



$$S0 = \overline{C0}\,\overline{C1}E0$$

$$S1 = \overline{C1}C0E1$$

$$S2 = C1\overline{C0}E2$$

$$S3 = C1C0E3$$

Mediante un decodificador