

TEMA 3: Circuitos Combinacionales

Índice:

1. Introducción
2. Circuitos Aritméticos
3. Decodificadores
4. Codificadores
5. Multiplexores
6. Demultiplexores

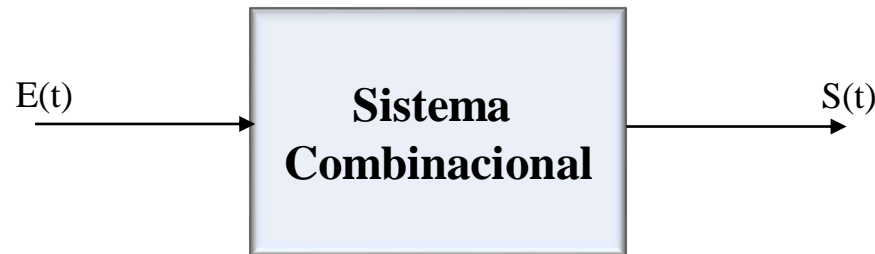
TEMA 3: Circuitos Combinacionales

Bibliografía:

- ❑ T.L.Floyd. Fundamentos de Sistemas Digitales.
 - Cap. 6: Funciones de la Lógica Combinacional
- ❑ C.Blanco. Fundamentos de Electrónica Digital.
 - Cap. 4: Circuitos Combinacionales.
- ❑ J.M^a Angulo y J. García. Sistemas Digitales y Tecnología de Computadores.
 - Cap. 4: Análisis y Diseño de Sistemas Combinacionales.
 - Cap. 6: Elementos Aritméticos.
- ❑ R.J.Tocci. Sistemas Digitales: Principios y Aplicaciones.
 - Cap. 6: Aritmética Digital.
 - Cap 9: Circuitos Lógicos MSI.

1. Introducción. Definición de Sistema Combinacional.

Es un conjunto de dispositivos lógicos en el que las salidas dependen exclusivamente del valor existente en cada momento en las entradas.



- Todo sistema combinacional se puede definir mediante funciones lógicas.
- Habitualmente los sistemas combinacionales se encuentran integrados en bloques que realizan funciones específicas.
- A la hora de implementar un sistema combinacional utilizaremos un criterio de simplificación, buscando la menor cantidad de puertas lógicas básicas.

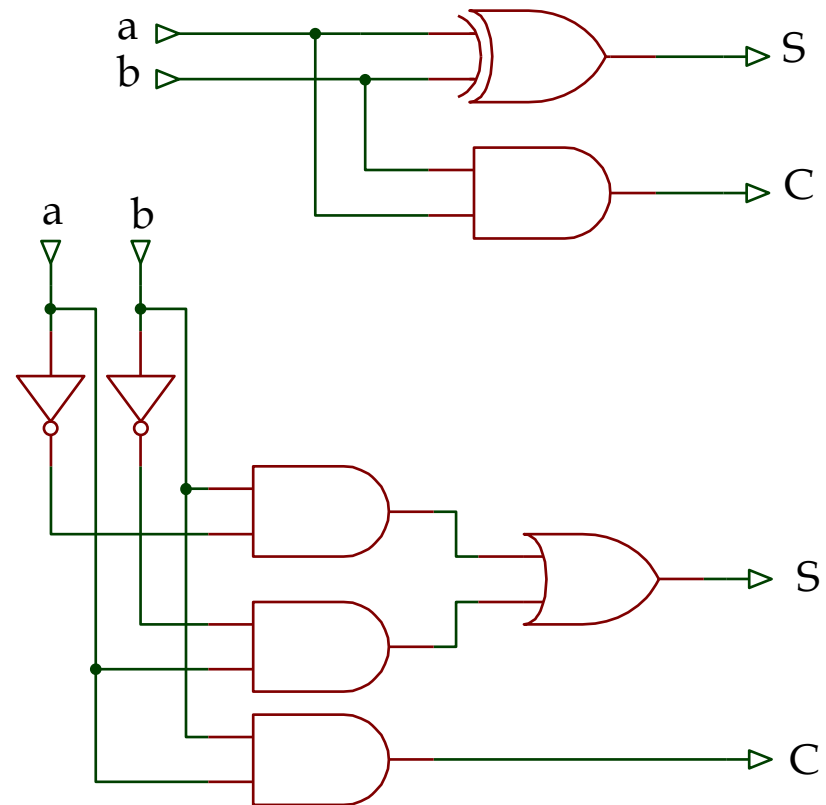
2. Circuitos Aritméticos. Semisumador

Es un circuito combinacional capaz de sumar dos dígitos binarios (a y b) proporcionando como salidas la suma (S) y el posible acarreo (C). Su tabla de verdad será:

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

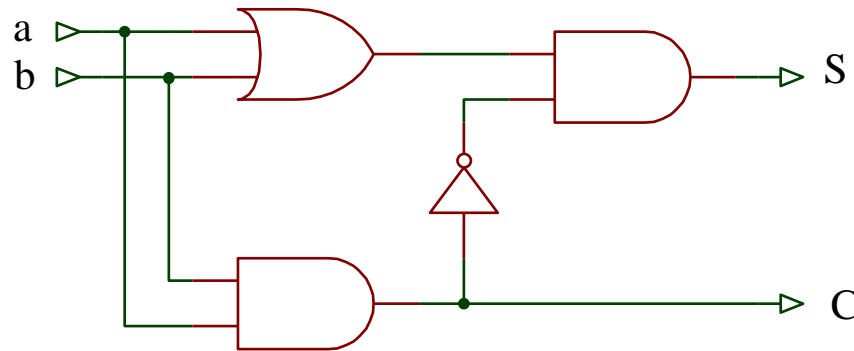
$$S = \bar{a}b + a\bar{b} = a \oplus b$$

$$C = ab$$

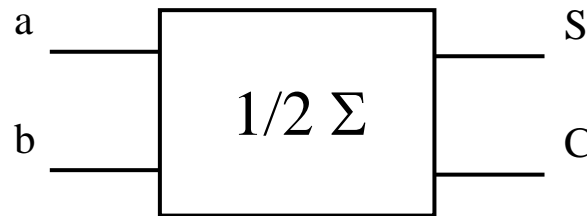


Si operamos la expresión de la suma:

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



Bloque Funcional:



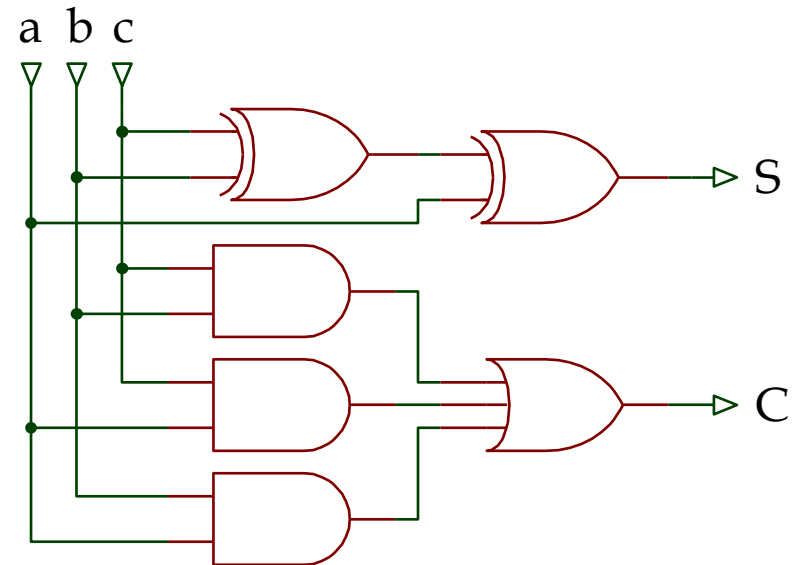
2.2 Sumador Completo

Es un circuito combinatorial 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

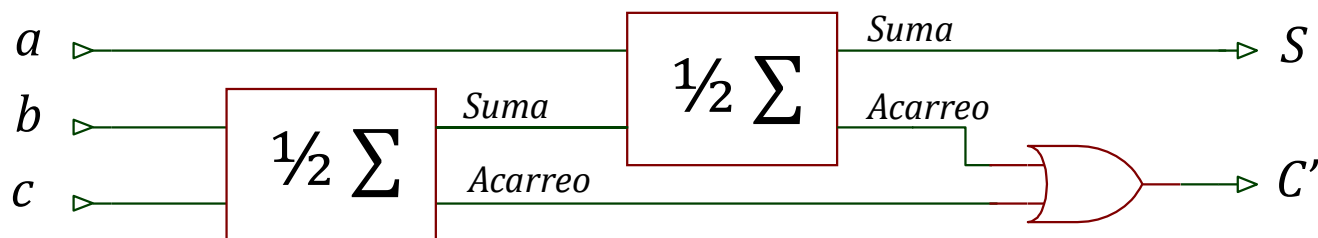
a	b	c	S	C
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

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

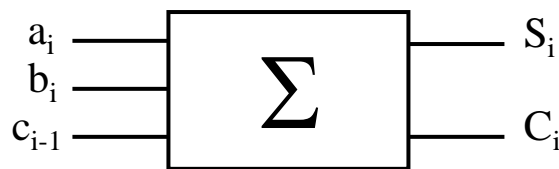
$$C = ab + ac + bc$$



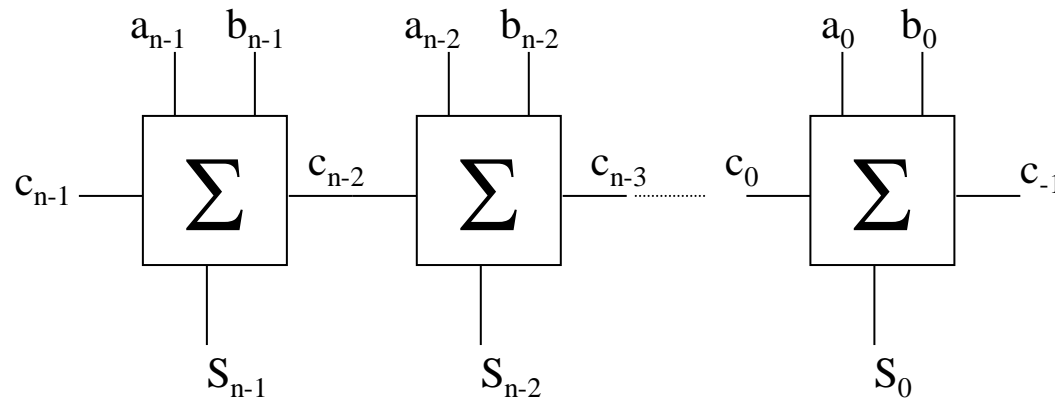
Construcción mediante Semisumadores



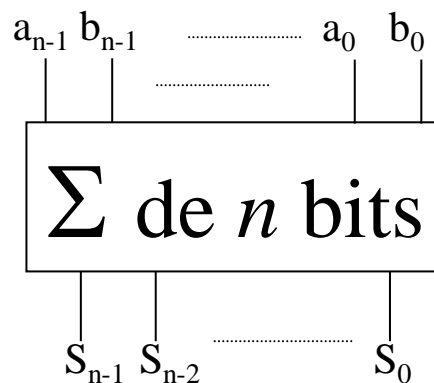
Bloque Funcional de un Sumador Completo de 1 Bit.



Podemos construir sumadores del número de bits que necesitamos simplemente encadenando sumadores completos de 1 bit



Bloque funcional del sumador paralelo de n bits

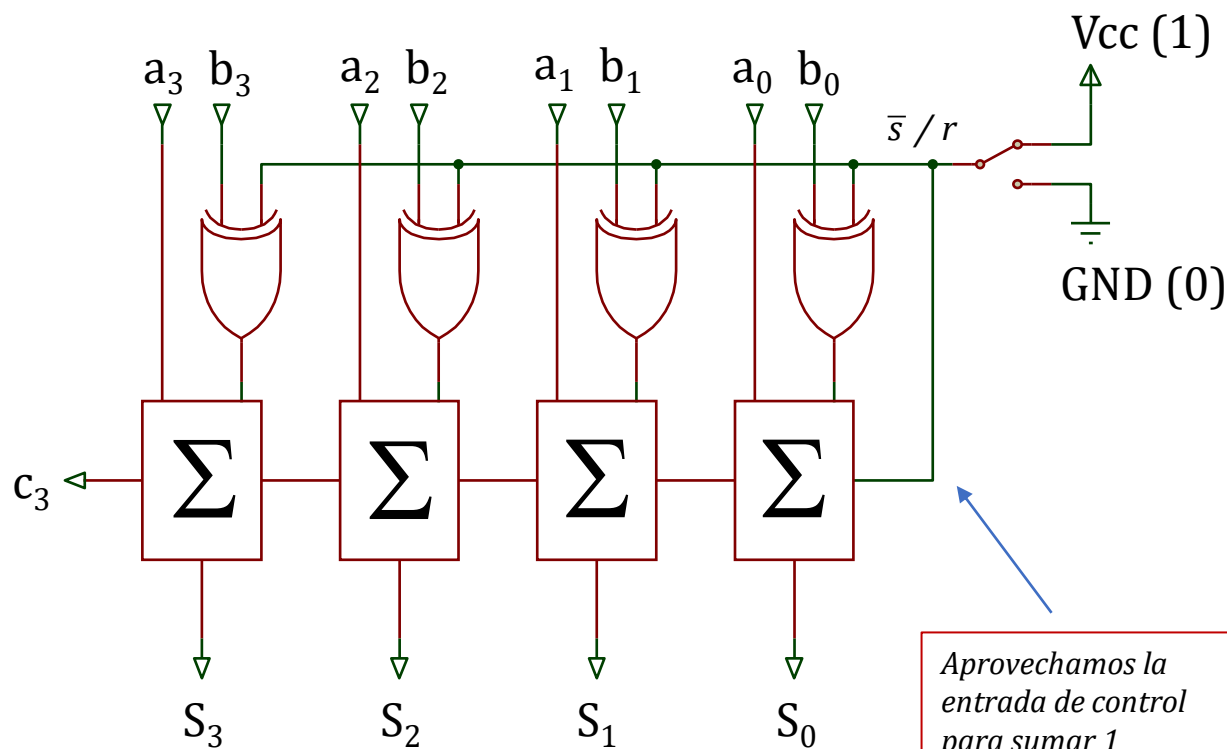


2.6 Sumador/Restador Binario

Es un circuito capaz de realizar las operaciones de suma y resta sobre números con signo. Si los números están representados en **Complemento a 2**, el circuito resultante sería:

\bar{s}/r	b	b'
0	0	0
0	1	1
1	0	1
1	1	0

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



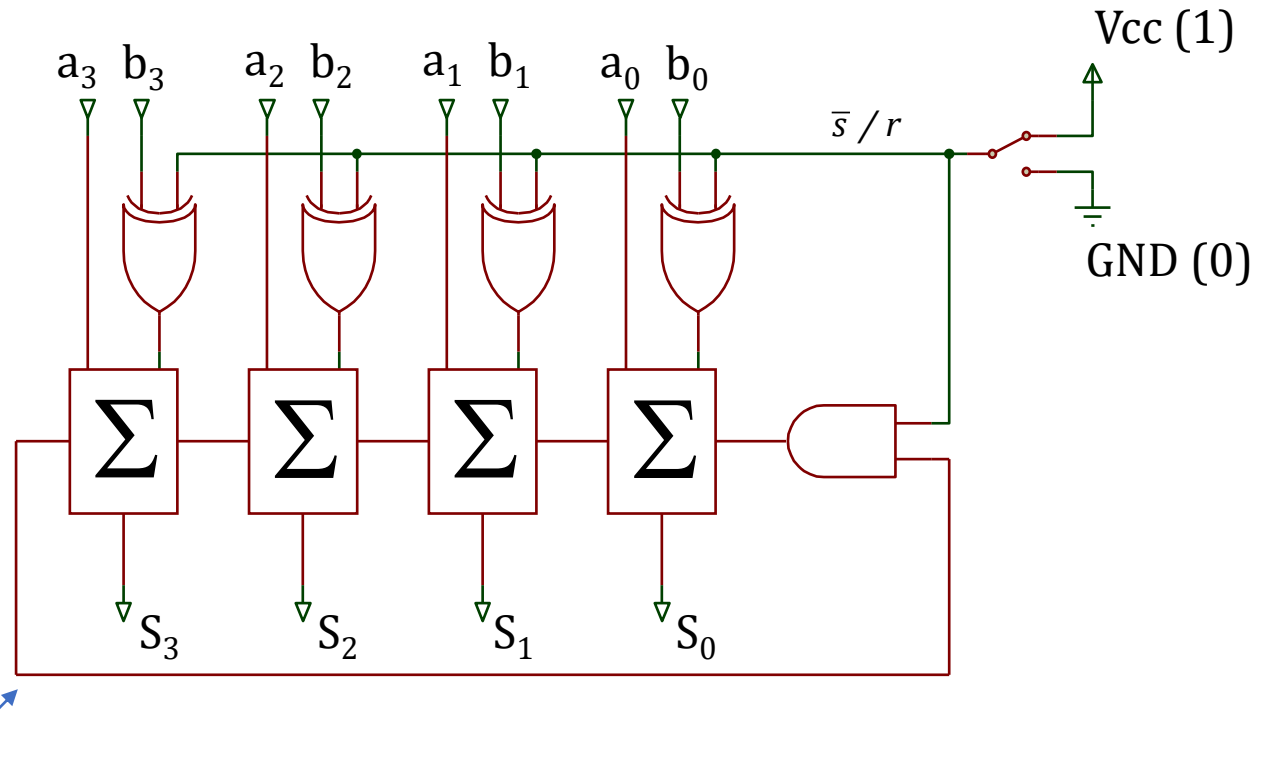
Carece de significado en C2

Aprovechamos la entrada de control para sumar 1

Si los números están representados en **Complemento a 1**, el circuito sería ahora:

\bar{s}/r	b	b'
0	0	0
0	1	1
1	0	1
1	1	0

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



Si estamos restando y hay acarreo final, hay que corregir el resultado sumando 1

3. Decodificador. Definición

Podemos definir al **decodificador** como un circuito combinatorial 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. Se suelen nombrar haciendo referencia al número de entradas y salidas que poseen. Así, 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

$$S_0 = \bar{a}\bar{b}$$

$$S_1 = \bar{a}b$$

$$S_2 = a\bar{b}$$

$$S_3 = ab$$

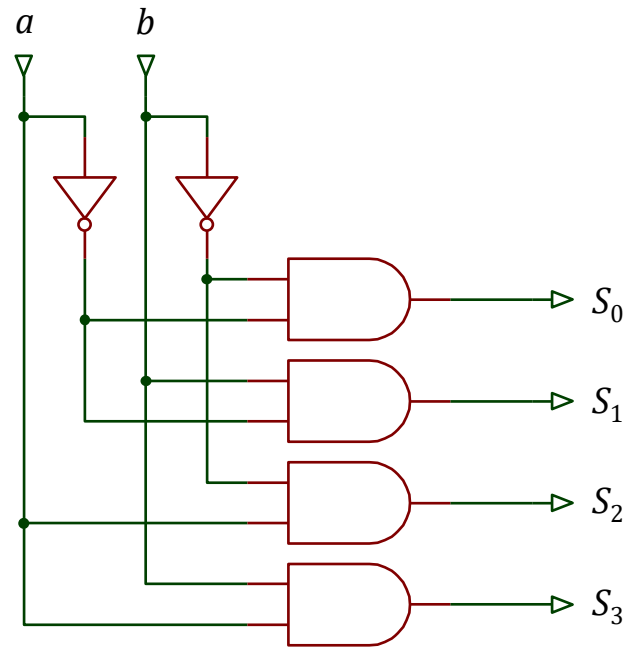
Cada salida hace referencia a uno de los posibles productos canónicos de las variables de entrada

$$S_0 = \bar{a}\bar{b}$$

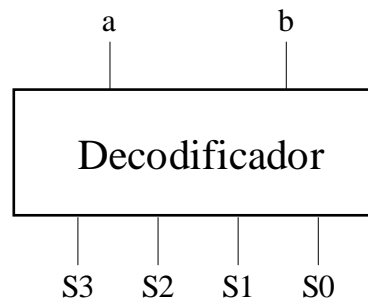
$$S_1 = \bar{a}b$$

$$S_2 = a\bar{b}$$

$$S_3 = ab$$

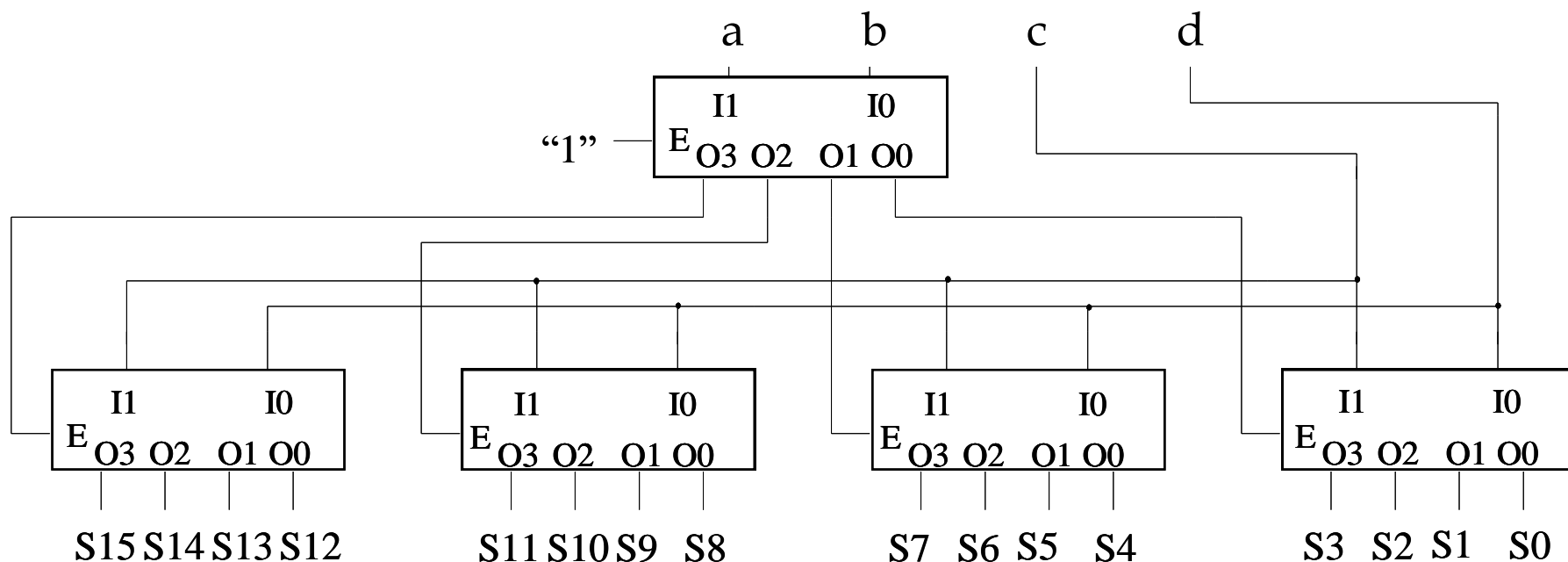


Bloque Funcional:



3.1 Agrupación de Decodificadores

Decodificador de 4 a 16 líneas mediante el uso de la entrada de habilitación E



3.2 Implementación de Funciones

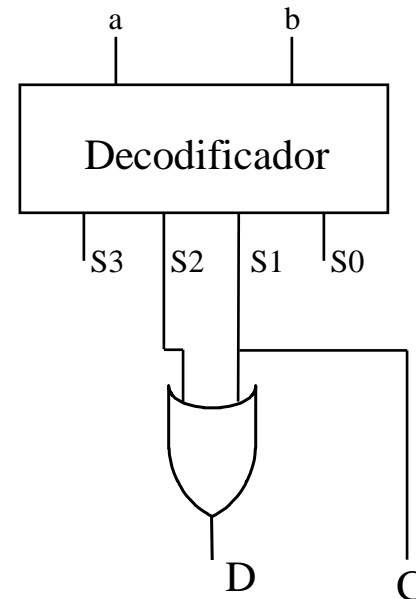
Puede utilizarse el decodificador de n entradas como circuito para implementar funciones lógicas de n variables lógicas. La idea se basa en utilizar los productos canónicos que produzcan un valor lógico 1 en la salida.

Ejemplo: Implementación de un semirestador.

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. Codificador. Definición

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. Existen dos tipos: los Codificadores Prioritarios y los No Prioritarios.

Un Codificador Decimal-BCD **no prioritario** tiene la siguiente tabla de verdad:

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

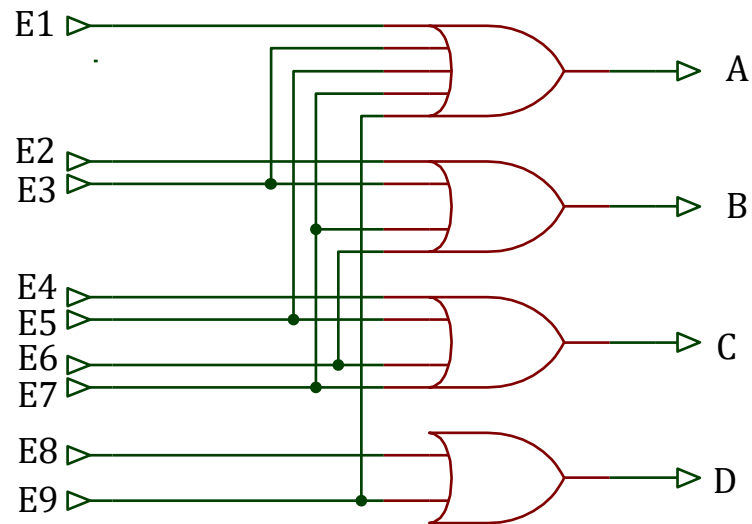
$$D = E8 + E9$$

$$C = E4 + E5 + E6 + E7$$

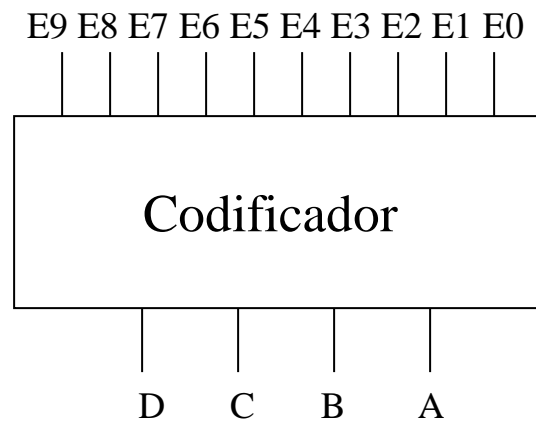
$$B = E2 + E3 + E6 + E7$$

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

Esquema:



Bloque funcional:



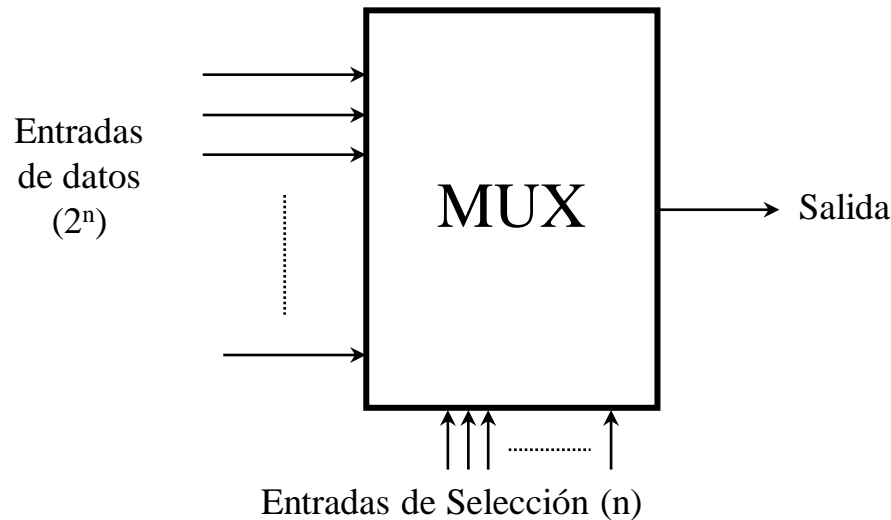
Si activamos más de una entrada simultáneamente en un codificador no prioritario se solaparán las combinaciones de salida que produce cada una de ellas.

Para evitar este problema se diseñan los **codificadores prioritarios**. En estos, cuando activamos más de una señal de entrada al mismo tiempo, solamente proporcionan la salida correspondiente a la entrada que está dotada de mayor prioridad, que generalmente es la que tiene asignada un mayor número decimal. El codificador Decimal-BCD anterior tendría la siguiente tabla de verdad:

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

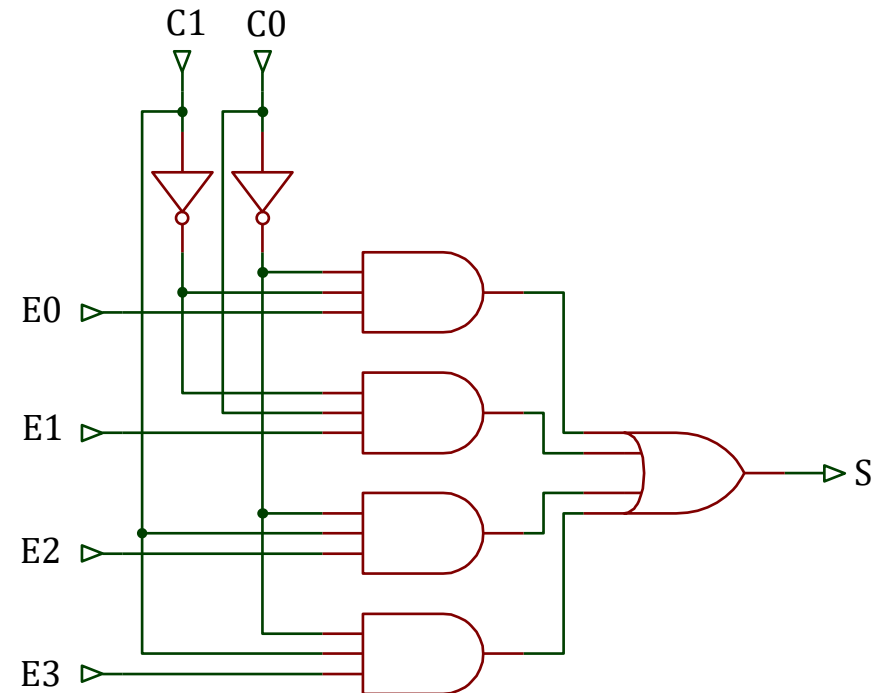
6. Multiplexor. Definición

Podemos definir al **multiplexor** o **selector de datos** 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.



Un multiplexor con dos entradas de selección y cuatro de información presenta la siguiente tabla de verdad:

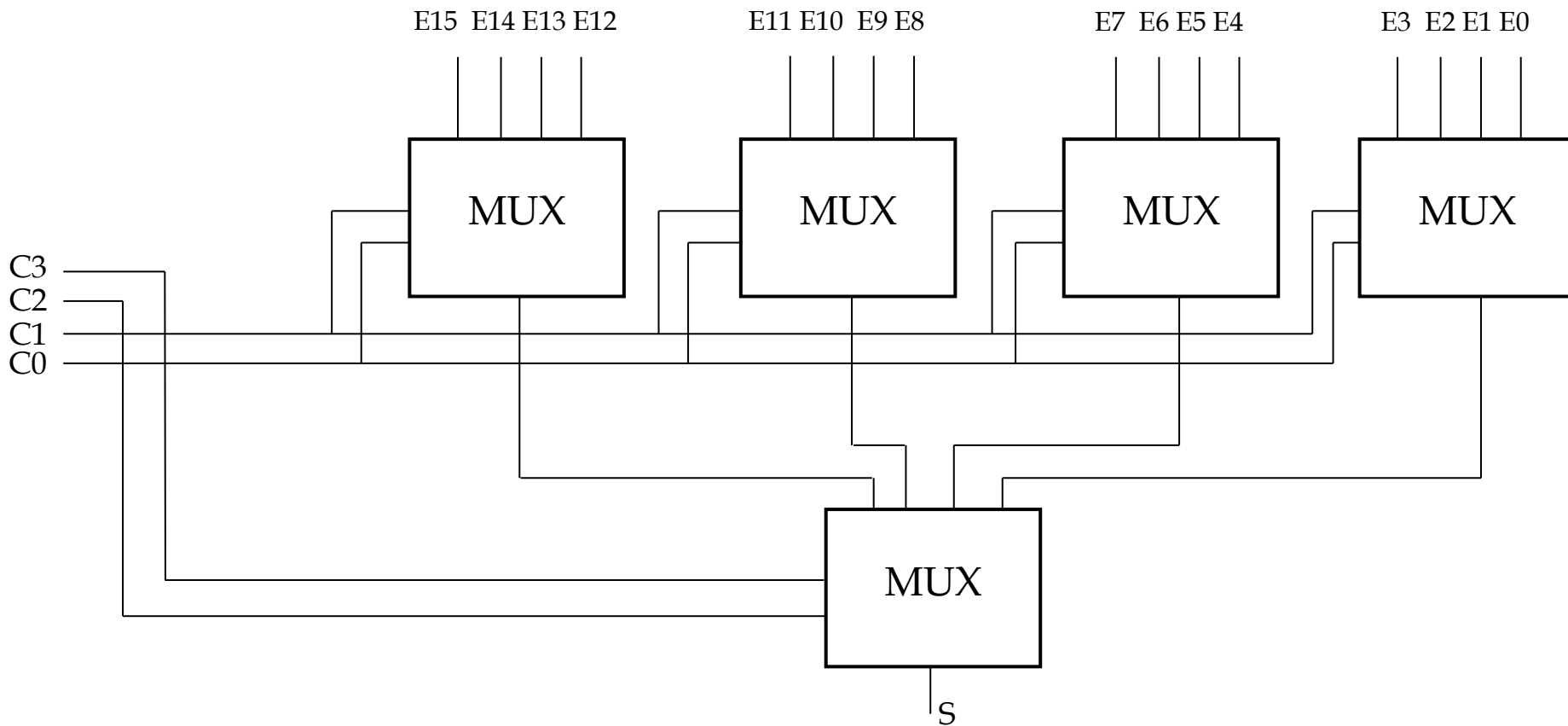
C1	C0	E3	E2	E1	E0	S
0	0	X	X	X	0	0
0	0	X	X	X	1	1
0	1	X	X	0	X	0
0	1	X	X	1	X	1
1	0	X	0	X	X	0
1	0	X	1	X	X	1
1	1	0	X	X	X	0
1	1	1	X	X	X	1



$$S = \overline{C0} \overline{C1} E0 + \overline{C1} C0 E1 + C1 \overline{C0} E2 + C1 C0 E3$$

6.1 Asociación de Multiplexores

Podemos construir multiplexores mayores asociando otros de menos entradas:



6.2 Implementación de Funciones

Puede utilizarse el multiplexor de n entradas de selección como circuito para implementar funciones lógicas de $n+1$ variable. La idea consiste en asociar a las entradas de selección a las variables lógicas, a excepción de una de ellas que se utiliza como entrada de datos.

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

n variables de selección				1 variable de datos		
				0	1	
2^n	{	0	0	...	0	→ selecciona entrada 0
		0	0	...	1	→ selecciona entrada 1
		1	1	...	1	→ selecciona entrada 2^{n-1}

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	→ variable_datos en E_i
Combinación en la entrada i	1	1	→ Siempre 1 en E_i

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

$$C = \sum_2 (1)$$

2 variables como:

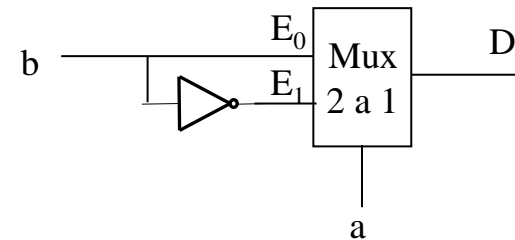
1 variable de selección (n-1) + 1 variable de datos

$$D = \sum_2 (1, 2)$$

a = variable de selección ; b = variable de datos

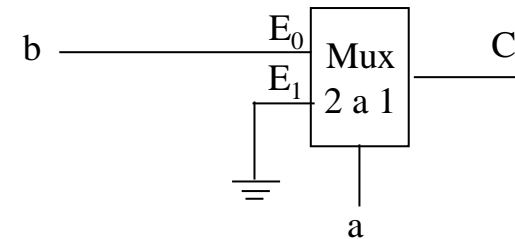
Implementación de D

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



Implementación de C

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



7. Demultiplexor. Definición

Podemos definir al **demultiplexor** como un circuito combinatorial 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.

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

$$S0 = \overline{C0} \overline{C1} E$$

$$S1 = \overline{C1} C0 E$$

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

$$S3 = C1 C0 E$$

Por su comportamiento, los demultiplexores constituyen el mismo bloque combinatorial que los decodificadores

