

5. OTROS CIRCUITOS COMBINACIONALES

5. OTROS CIRCUITOS COMBINACIONALES

5.1 Circuitos y subsistemas combinacionales

- Concepto de circuito combinacional

- Multiplexor digital:

 - Concepto general.

 - Extensión de capacidad en multiplexores.

 - Aplicaciones de los multiplexores: conversión paralelo-serie y generación de funciones.

- Codificadores:

 - Codificadores sin prioridad.

 - Codificadores con prioridad.

- Decodificadores:

 - Decodificadores de salidas mutuamente excluyentes (no excitadores),

 - Decodificadores excitadores (Drivers).

 - Extensión de la capacidad de un decodificador.

 - Aplicaciones de decodificadores: Conversión serie-paralelo (demultiplexor) y generar funciones.

- Convertidores de código.

- Generador-comprobador de paridad:

 - Extensión de la capacidad de un generador-comprobador de paridad.

- Comparador binario:

 - Concepto general, Extensión de la capacidad de un comparador.

Circuitos y subsistemas combinacionales

- Concepto de circuito combinacional**

Aquel en el que los estados lógicos de las variables de salida en cada instante dependen únicamente de los estados lógicos de las variables de entrada en el mismo instante



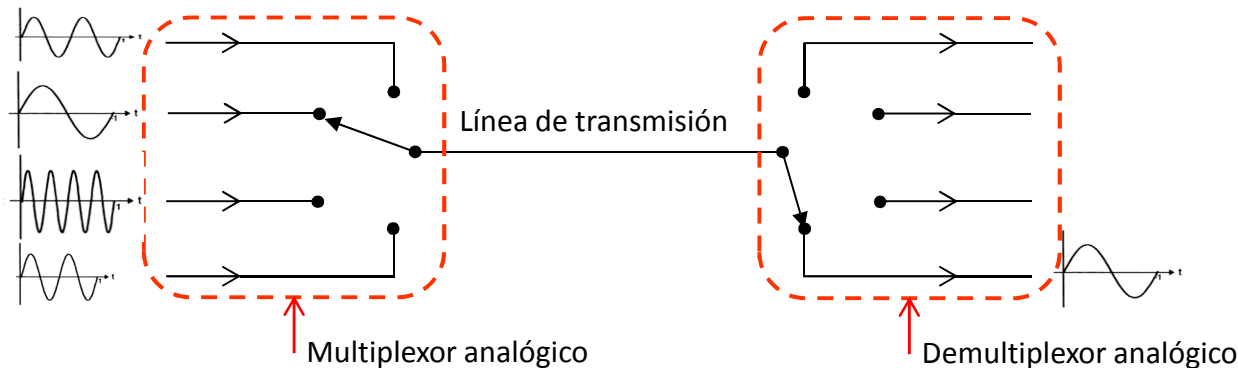
En el siguiente capítulo se abordarán circuitos más complejos, donde la salida también depende de datos que hay almacenados en memoria. Se trata del otro gran grupo: los circuitos secuenciales.

En este capítulo se verán circuitos combinacionales relacionados con transferencia de información y generación de códigos para presentar la información de salida de los sistemas digitales: multiplexores y demultiplexores, codificadores y decodificadores, conversores de código, generadores y detectores de bit de paridad, y comparadores

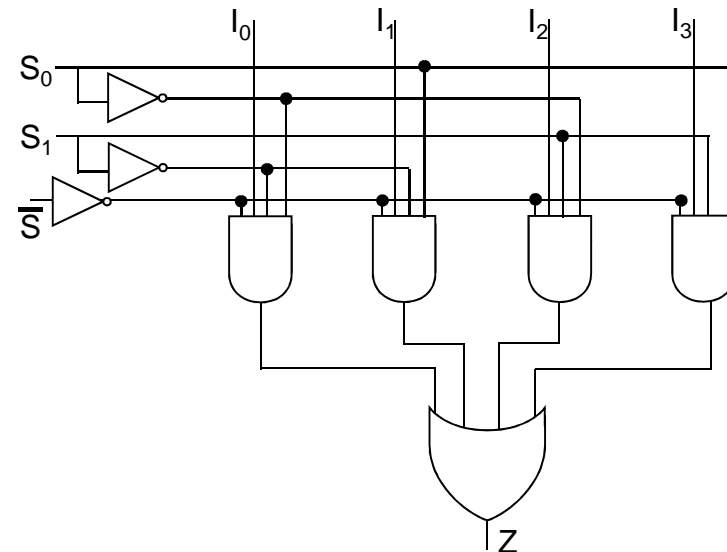
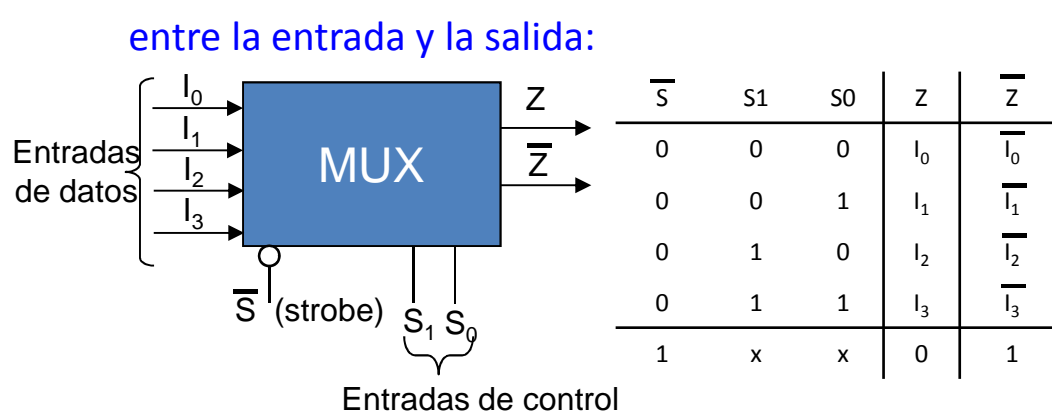
Circuitos y subsistemas combinacionales

• Multiplexor digital

En electrónica analógica el multiplexor analógico conecta una señal seleccionada de entre varias entradas a una sola salida:



En el multiplexor analógico existe contacto directo entre la línea de transmisión y la línea seleccionada. El multiplexor digital funciona distinto. Una serie de entradas de control eligen una de las entradas de datos digitales, y el valor de la entrada seleccionada se transmite a la salida sin que haya contacto entre la entrada y la salida:



$$Z = \overline{S} \cdot \overline{S_1} \cdot \overline{S_0} \cdot I_0 + \overline{S} \cdot \overline{S_1} \cdot S_0 \cdot I_1 + \overline{S} \cdot S_1 \cdot \overline{S_0} \cdot I_2 + \overline{S} \cdot S_1 \cdot S_0 \cdot I_3$$

$$\overline{Z} = \overline{S} \cdot \overline{S_1} \cdot \overline{S_0} \cdot \overline{I_0} + \overline{S} \cdot \overline{S_1} \cdot S_0 \cdot \overline{I_1} + \overline{S} \cdot S_1 \cdot \overline{S_0} \cdot \overline{I_2} + \overline{S} \cdot S_1 \cdot S_0 \cdot \overline{I_3}$$

Circuitos y subsistemas combinacionales

La relación entre las entradas de control y las de datos en multiplexor debe cumplir:

- Si un multiplexor tiene N entradas de datos y n entradas de control, $2^n \geq N$

En el multiplexor comercial CD54ACT151 (<http://bit.ly/1o8I5nq>), hay 8 entradas de datos y 3 control

Cumple la anterior relación ya que: $2^3=8$:

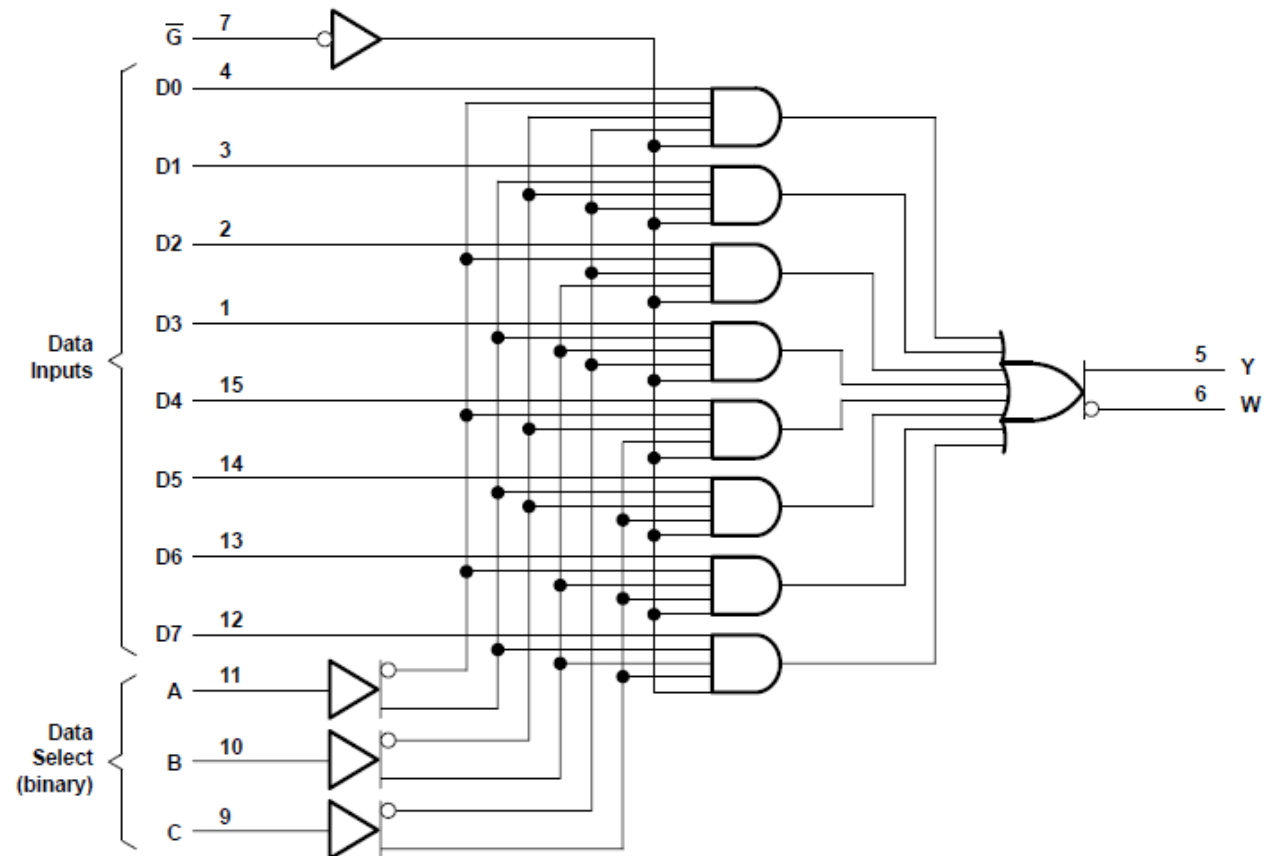
TABLA DE VERDAD

FUNCTION TABLE					
INPUTS				OUTPUTS	
SELECT			STROBE \overline{G}	Y	W
C	B	A			
X	X	X	H	L	H
L	L	L	L	D0	$\overline{D0}$
L	L	H	L	D1	$\overline{D1}$
L	H	L	L	D2	$\overline{D2}$
L	H	H	L	D3	$\overline{D3}$
H	L	L	L	D4	$\overline{D4}$
H	L	H	L	D5	$\overline{D5}$
H	H	L	L	D6	$\overline{D6}$
H	H	H	L	D7	$\overline{D7}$

D0, D1 . . . D7 = the level of the respective D input

CIRCUITO DIGITAL

logic diagram (positive logic)

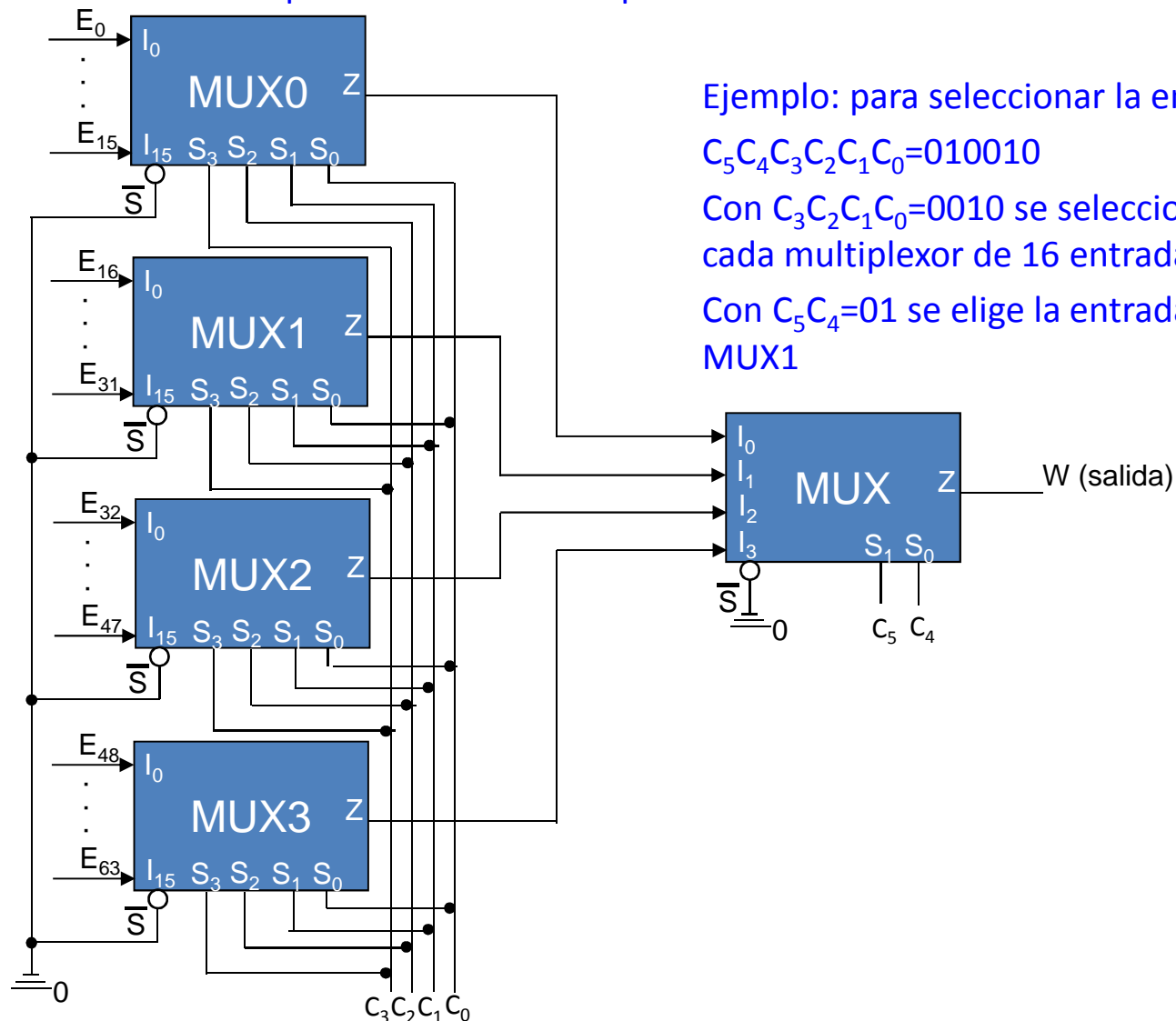


Circuitos y subsistemas combinacionales

- Extensión de la capacidad de los multiplexores**

No existen multiplexores comerciales de más de 16 entradas de datos.

¿Cómo hacer entonces para diseñar un multiplexor de más entradas de control?



Ejemplo: para seleccionar la entrada E₁₈:

$C_5C_4C_3C_2C_1C_0=010010$

Con $C_3C_2C_1C_0=0010$ se selecciona la entrada I₂ de cada multiplexor de 16 entradas

Con $C_5C_4=01$ se elige la entrada I₁ del multiplexor MUX1

• Aplicaciones de los multiplexores

a) Generación de funciones

La ecuación general de un multiplexor de dos entradas de control genera los productos canónicos de dichas entradas de control:

$$Z = \overline{S_1} \cdot \overline{S_0} \cdot I_0 + \overline{S_1} \cdot S_0 \cdot I_1 + S_1 \cdot \overline{S_0} \cdot I_2 + S_1 \cdot S_0 \cdot I_3$$

Por tanto, si los I_0 , I_1 , I_2 e I_3 corresponden con otra variable, con un multiplexor se puede obtener una función de n entradas de control más una variable. Es decir, de $n+1$ variables siendo n el número de entradas de control.

Ejemplo: obtener la siguiente función de 4 variables con un multiplexor (tendrá 3 de control):

$$F = \overline{B} \cdot C \cdot D + A \cdot \overline{B} \cdot D + \overline{A} \cdot \overline{B} \cdot C$$

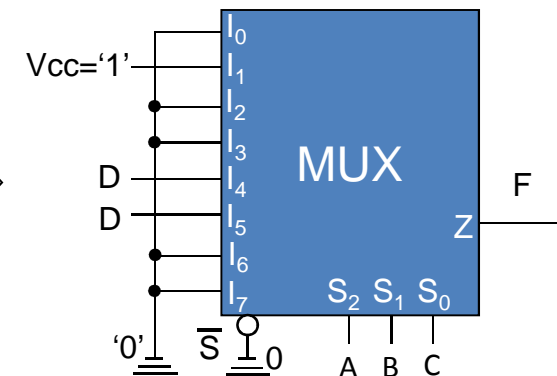
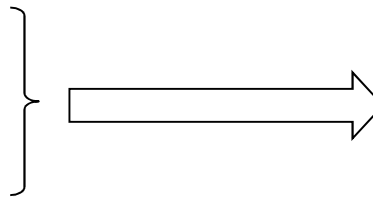
El primer paso es poner la función en forma canónica (suma de minterms):

$$F = (A + \overline{A}) \cdot \overline{B} \cdot C \cdot D + A \cdot \overline{B} \cdot (C + \overline{C}) \cdot D + \overline{A} \cdot \overline{B} \cdot C \cdot (D + \overline{D}) = A \cdot \overline{B} \cdot C \cdot D + \overline{A} \cdot \overline{B} \cdot C \cdot D + A \cdot \overline{B} \cdot C \cdot \overline{D} + \overline{A} \cdot \overline{B} \cdot C \cdot \overline{D} + \overline{A} \cdot \overline{B} \cdot C \cdot D + \overline{A} \cdot \overline{B} \cdot C \cdot \overline{D}$$

A continuación se eligen tres variables de control (en nuestro caso elegiremos A, B y C).

Después ordenamos los términos de la ecuación para que se corresponda con la ecuación de un Multiplexor (los términos que no existen se multiplican por 0):

$$\begin{array}{cccc} \overbrace{000} & \overbrace{001} & \overbrace{010} & \overbrace{011} \\ \overline{A} \cdot \overline{B} \cdot \overline{C} \cdot 0 + \overline{A} \cdot \overline{B} \cdot \overline{C} \cdot (D + \overline{D}) + \overline{A} \cdot \overline{B} \cdot C \cdot 0 + \overline{A} \cdot \overline{B} \cdot C \cdot 0 + \\ & \underbrace{1} & & \\ + \underbrace{A \cdot \overline{B} \cdot \overline{C} \cdot D}_{100} + \underbrace{A \cdot \overline{B} \cdot C \cdot D}_{101} + \underbrace{A \cdot \overline{B} \cdot \overline{C} \cdot 0}_{110} + \underbrace{A \cdot \overline{B} \cdot C \cdot 0}_{111} \end{array}$$



Circuitos y subsistemas combinacionales

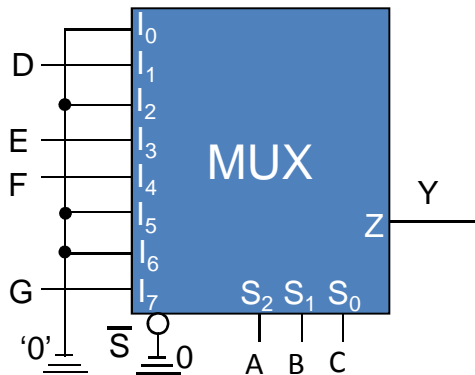
a) Generación de funciones

Aunque con multiplexores habitualmente sólo se pueden representar funciones de $n+1$ variables (siendo n el número de entradas de control), hay casos como el siguiente en que se puede aumentar el número de variables a representar:

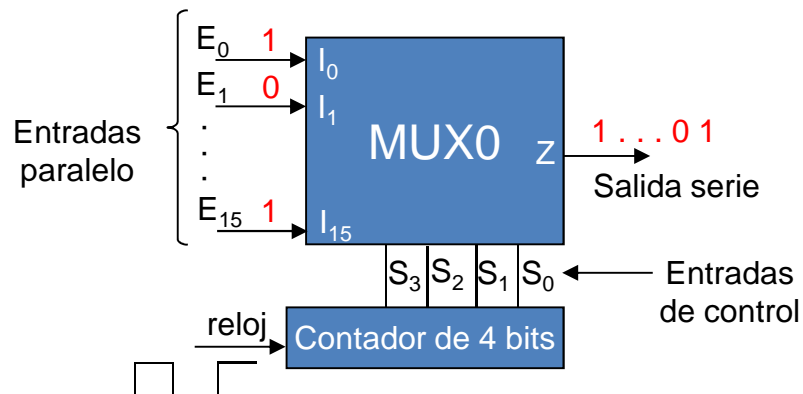
$$Y = \bar{A} \cdot \bar{B} \cdot C \cdot D + \bar{A} \cdot B \cdot C \cdot E + A \cdot \bar{B} \cdot \bar{C} \cdot F + A \cdot B \cdot C \cdot G$$

Lo ponemos en forma de ecuación de multiplexor eligiendo A,B y C como variables de control:

$$Y = \bar{A} \cdot \bar{B} \cdot \bar{C} \cdot 0 + \bar{A} \cdot \bar{B} \cdot C \cdot D + \bar{A} \cdot B \cdot \bar{C} \cdot 0 + \bar{A} \cdot B \cdot C \cdot E + A \cdot \bar{B} \cdot \bar{C} \cdot F + A \cdot \bar{B} \cdot C \cdot 0 + A \cdot B \cdot \bar{C} \cdot 0 + A \cdot B \cdot C \cdot G$$



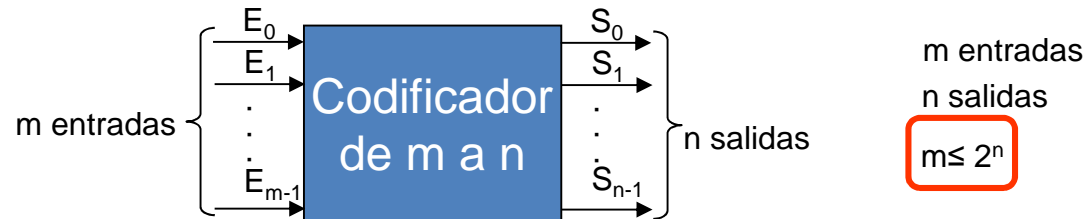
b) Conversión paralelo-serie.



Circuitos y subsistemas combinacionales

- Codificador**

Obtiene un código de salida que se ajusta a los valores introducidos a la entrada



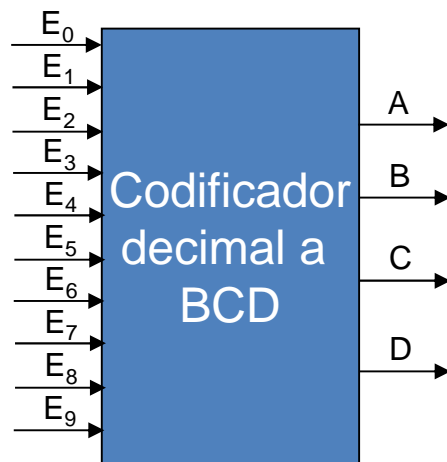
Existen dos versiones:

- a) Codificador sin prioridad**

Sólo se activa una entrada

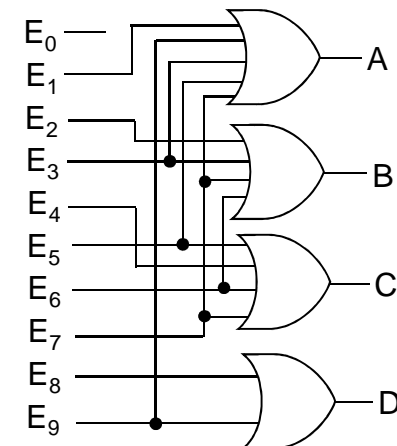
La salida es la codificación de la entrada activa

Ejemplo: Codificador Decimal a BCD



Dígito decimal	Entradas	Salidas			
		D	C	B	A
0	E_0	0	0	0	0
1	E_1	0	0	0	1
2	E_2	0	0	1	0
3	E_3	0	0	1	1
4	E_4	0	1	0	0
5	E_5	0	1	0	1
6	E_6	0	1	1	0
7	E_7	0	1	1	1
8	E_8	1	0	0	0
9	E_9	1	0	0	1

$$\begin{aligned} D &= E_8 + E_9 & A &= E_1 + E_3 + E_5 + E_7 + E_9 \\ C &= E_4 + E_5 + E_6 + E_7 & B &= E_2 + E_3 + E_6 + E_7 \end{aligned}$$



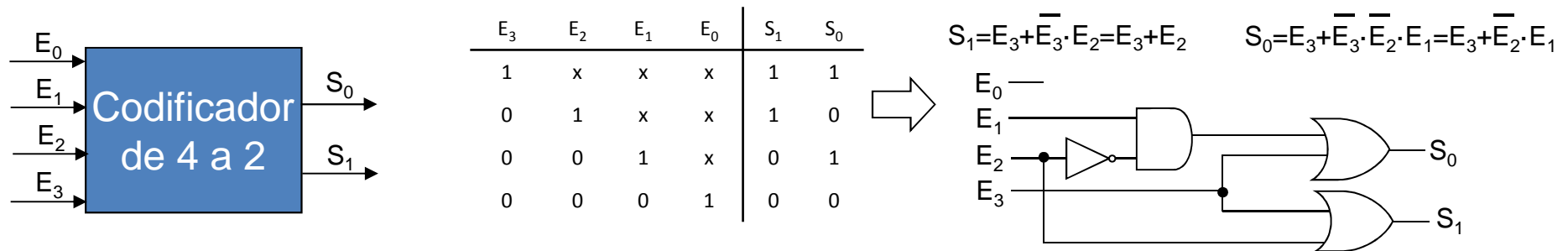
Circuitos y subsistemas combinacionales

b) Codificador con prioridad

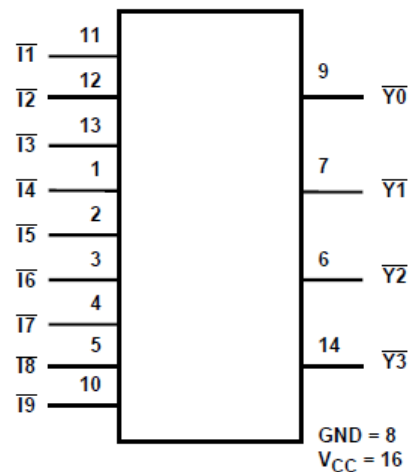
Se puede activar más de una entrada a la vez

Si se activan varias entradas a la vez, la salida codificada corresponde a la entrada de mayor prioridad
(en los codificadores sin prioridad se produce un error si se activa más de una entrada)

Ejemplo 1: Codificador 4 a 2 con prioridad $E_3 > E_2 > E_1 > E_0$



Ejemplo 2: 10 a 4 con prioridad $I_9 > I_8 > I_7 > I_6 > I_5 > I_4 > I_3 > I_2 > I_1$ (las entradas y las salidas son activas a nivel bajo = L) <http://bit.ly/1pvyyLa>



INPUTS									OUTPUTS			
$\bar{I}1$	$\bar{I}2$	$\bar{I}3$	$\bar{I}4$	$\bar{I}5$	$\bar{I}6$	$\bar{I}7$	$\bar{I}8$	$\bar{I}9$	$\bar{Y}3$	$\bar{Y}2$	$\bar{Y}1$	$\bar{Y}0$
H	H	H	H	H	H	H	H	H	H	H	H	H
X	X	X	X	X	X	X	X	L	L	H	H	L
X	X	X	X	X	X	X	L	H	L	H	H	H
X	X	X	X	X	X	L	H	H	H	L	L	L
X	X	X	X	X	L	H	H	H	H	L	L	H
X	X	X	X	L	H	H	H	H	H	L	H	L
X	X	X	L	H	H	H	H	H	H	L	H	H
X	X	L	H	H	H	H	H	H	H	H	L	L
X	L	H	H	H	H	H	H	H	H	H	L	H
L	H	H	H	H	H	H	H	H	H	H	H	L

H = High Logic Level, L = Low Logic Level, X = Don't Care

Circuitos y subsistemas combinacionales

- Decodificadores**

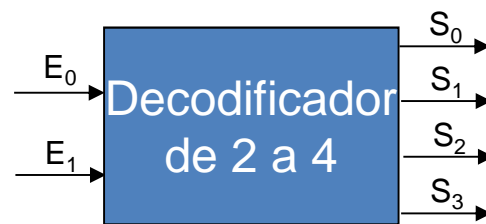
Se traduce un código recibido por las entradas en valores de salida que representan de una forma sencilla dicho código

Existen de dos tipos:

a) Decodificadores con salidas mutuamente excluyentes (no excitadores):

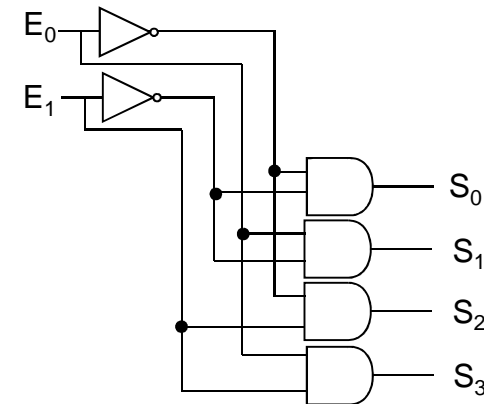
Ante un código de entrada se activa una salida (es la función opuesta al codificador sin prioridad, donde ante una entrada activada se obtenía un código a la salida)

Ejemplo 1: decodificador de 2 entradas (2 a 4)



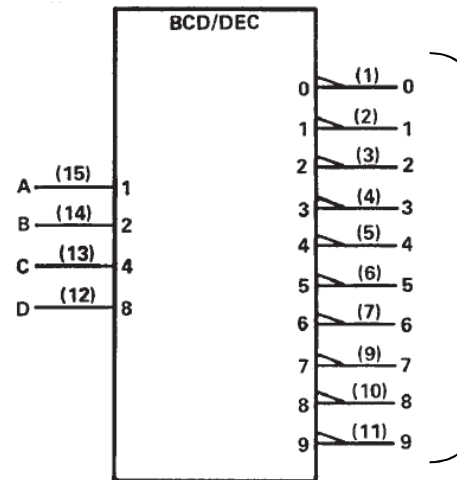
Entradas		Salidas			
E_1	E_0	S_3	S_2	S_1	S_0
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	0	0	0

$$\begin{aligned} S_3 &= E_1 \cdot E_0 & S_1 &= \bar{E}_1 \cdot E_0 \\ S_2 &= E_1 \cdot \bar{E}_0 & S_0 &= \bar{E}_1 \cdot \bar{E}_0 \end{aligned}$$



Circuitos y subsistemas combinacionales

Ejemplo 2: codificador BCD a decimal (circuito sn74ls42 <http://bit.ly/1ohdG23>)

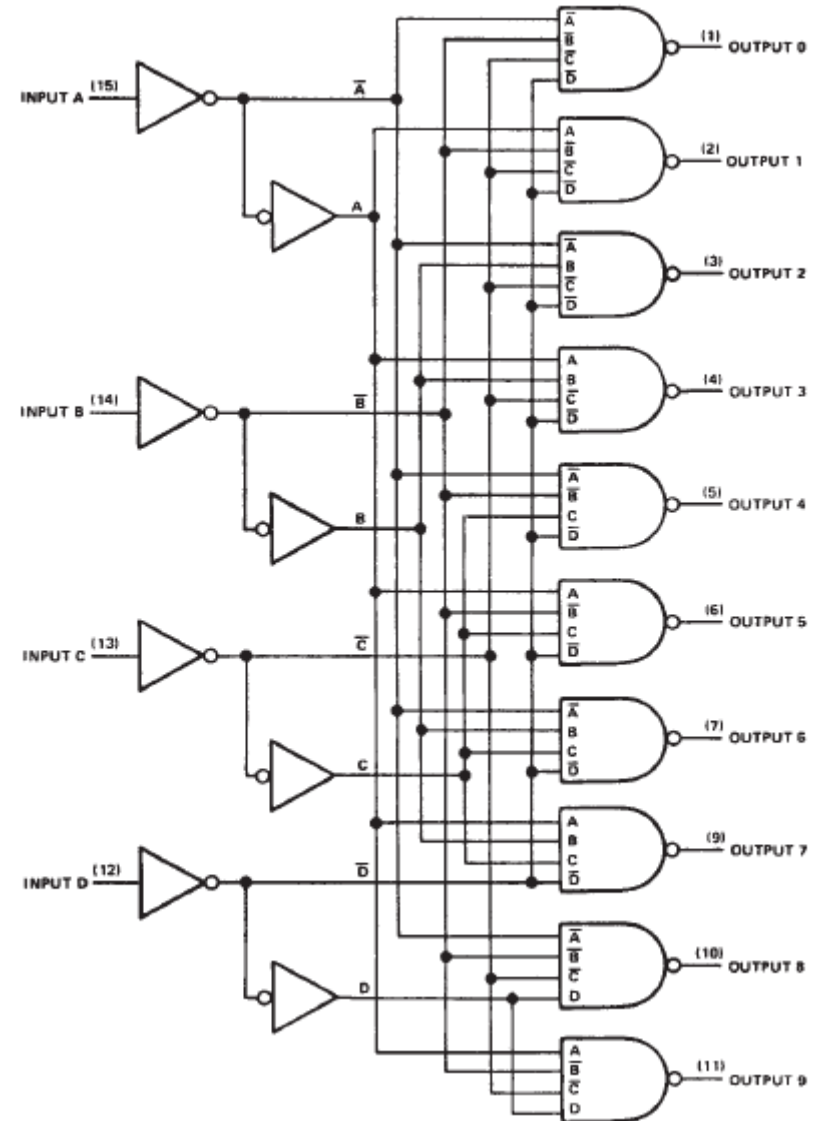


Salidas activas a nivel bajo (low level = L)

FUNCTION TABLE

NO.	BCD INPUT				DECIMAL OUTPUT									
	D	C	B	A	0	1	2	3	4	5	6	7	8	9
0	L	L	L	L	L	H	H	H	H	H	H	H	H	H
1	L	L	L	H	H	L	H	H	H	H	H	H	H	H
2	L	L	H	L	H	H	L	H	H	H	H	H	H	H
3	L	L	H	H	H	H	H	L	H	H	H	H	H	H
4	L	H	L	L	H	H	H	H	L	H	H	H	H	H
5	L	H	L	H	H	H	H	H	H	L	H	H	H	H
6	L	H	H	L	H	H	H	H	H	H	L	H	H	H
7	L	H	H	H	H	H	H	H	H	H	H	L	H	H
8	H	L	L	L	H	H	H	H	H	H	H	H	L	H
9	H	L	L	H	H	H	H	H	H	H	H	H	H	L
INVALID	H	L	H	L	H	H	H	H	H	H	H	H	H	H
	H	L	H	H	H	H	H	H	H	H	H	H	H	H
	H	H	L	L	H	H	H	H	H	H	H	H	H	H
	H	H	L	H	H	H	H	H	H	H	H	H	H	H
	H	H	H	L	H	H	H	H	H	H	H	H	H	H

H = high level, L = low level

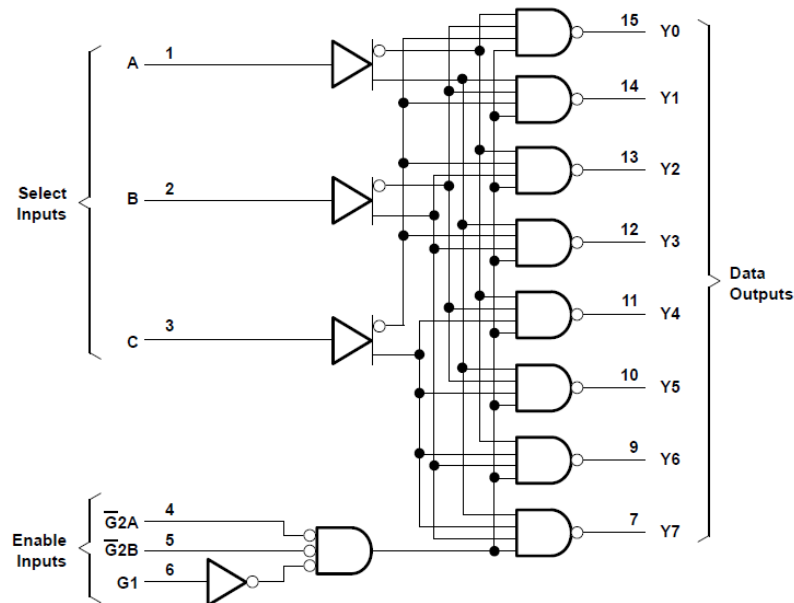


Circuitos y subsistemas combinacionales

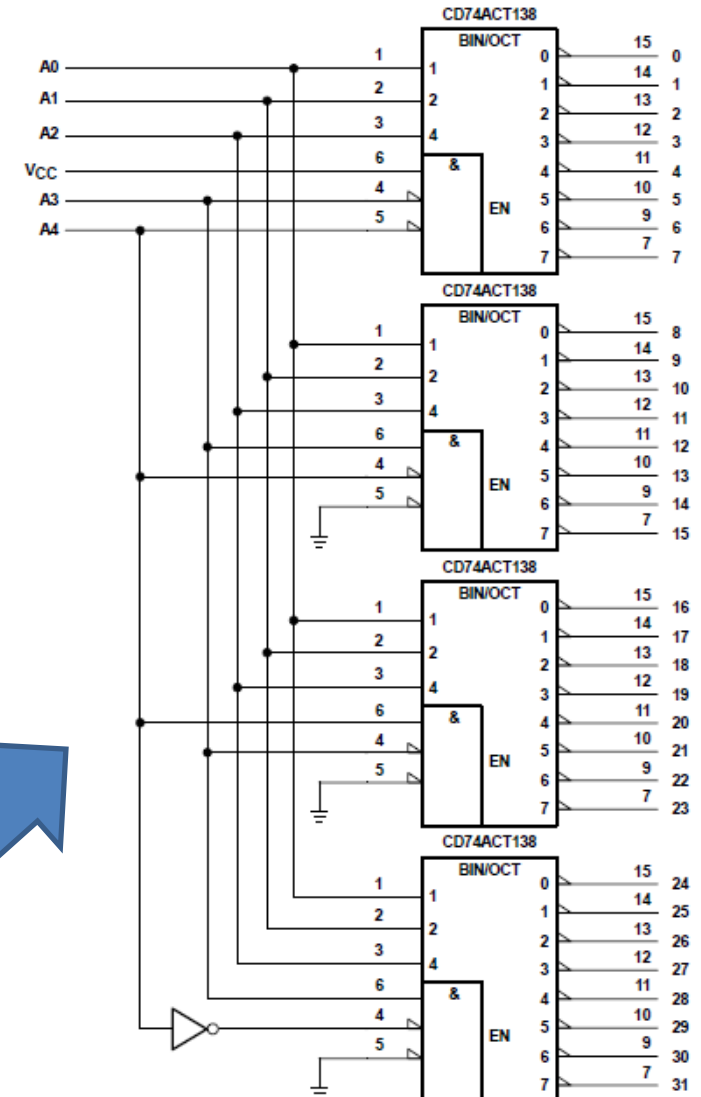
Ejemplo 3: extensión de decodificadores

Análogamente que con los multiplexores, si queremos hacer un decodificador de más de 32 salidas se combinan circuitos decodificadores más sencillos como el CD54ACT138 (<http://bit.ly/1u7uAW9>):

FUNCTION TABLE													
ENABLE INPUTS			SELECT INPUTS			OUTPUTS							
G1	$\overline{G2A}$	$\overline{G2B}$	C	B	A	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
X	H	X	X	X	X	H	H	H	H	H	H	H	H
X	X	H	X	X	X	H	H	H	H	H	H	H	H
L	X	X	X	X	X	H	H	H	H	H	H	H	H
H	L	L	L	L	L	L	H	H	H	H	H	H	H
H	L	L	L	L	H	H	L	H	H	H	H	H	H
H	L	L	L	H	L	H	H	L	H	H	H	H	H
H	L	L	L	H	H	H	H	L	H	H	H	H	H
H	L	L	H	L	L	H	H	H	H	L	H	H	H
H	L	L	H	L	H	H	H	H	H	H	L	H	H
H	L	L	H	H	L	H	H	H	H	H	H	L	H
H	L	L	H	H	H	H	H	H	H	H	H	H	L



El decodificador tiene 3 entradas A, B y C, 8 salidas Y0...Y7 y tres entradas de habilitación que se usarán para la extensión



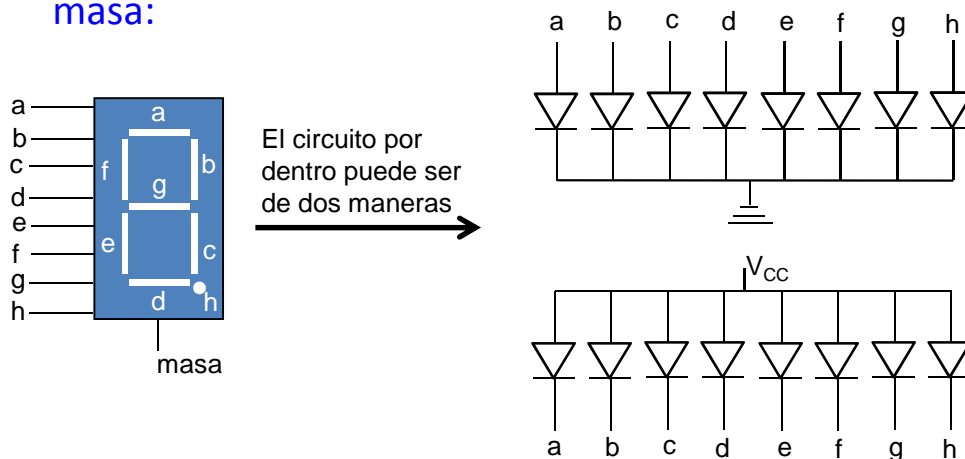
Circuitos y subsistemas combinacionales

b) Decodificadores excitadores (Drivers)

El código de entrada se traduce en la activación de una serie de salidas que excitarán un dispositivo visual como puede ser un display de 7 segmentos o una pantalla LCD. Para excitar el dispositivo visual se permite, a diferencia del excitador con salidas mutuamente excluyentes, activar más de una salida

Ejemplo: Decodificador excitador para display 7 segmentos

Un display de 7 segmentos es un dispositivo electrónico con 9 pines: los 7 segmentos, el punto y la masa:

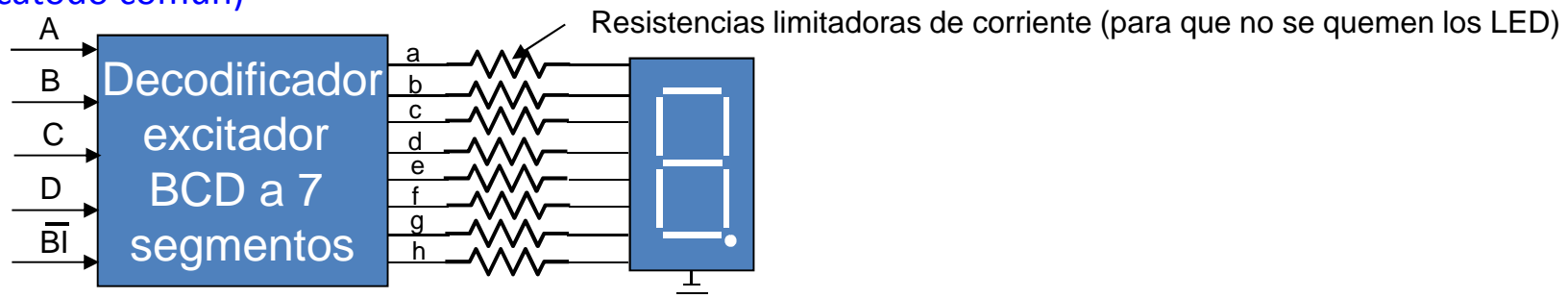


Esta configuración se llama cátodo común y el cátodo se conecta a tierra.

Si las entradas a,b,c,d,e,f,g ó h tienen tensión alta el LED se enciende, y si tienen tensión baja el LED está apagado

Esta configuración se llama ánodo común y el ánodo se conecta a tensión de alimentación del circuito (V_{CC}). Si las entradas a,b,c,d,e,f,g ó h tienen tensión baja el LED se enciende, y si tienen tensión alta el LED está apagado

El decodificador excitador se conecta al display de la siguiente manera (suponiendo configuración cátodo común)

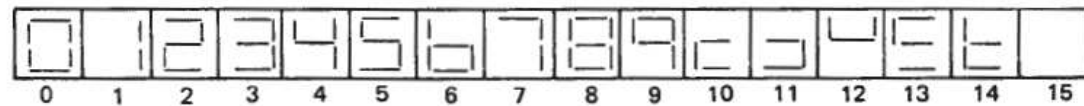
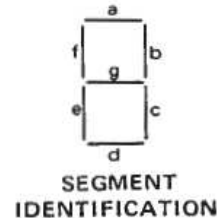
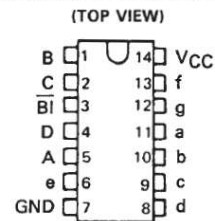


Circuitos y subsistemas combinacionales

b) Decodificadores excitadores (Drivers)

Un ejemplo de circuito excitador comercial para displays 7 segmentos es el SN7446A (<http://bit.ly/1zEive8>). No utiliza el punto (segmento h):

SN54LS49 . . . J OR W PACKAGE
SN74LS49 . . . D OR N PACKAGE



NUMERICAL DESIGNATIONS AND RESULTANT DISPLAYS

'LS49

FUNCTION TABLE (T3)

DECIMAL OR FUNCTION	INPUTS					OUTPUTS							NOTE
	D	C	B	A	\overline{BI}	a	b	c	d	e	f	g	
0	L	L	L	L	H	H	H	H	H	H	H	L	1
1	L	L	L	H	H	L	H	H	L	L	L	L	
2	L	L	H	L	H	H	H	L	H	H	L	H	
3	L	L	H	H	H	H	H	H	H	L	L	H	
4	L	H	L	L	H	L	H	H	L	L	H	H	
5	L	H	L	H	H	H	L	H	H	L	H	H	
6	L	H	H	L	H	L	L	H	H	H	H	H	
7	L	H	H	H	H	H	H	H	L	L	L	L	
8	H	L	L	L	H	H	H	H	H	H	H	H	
9	H	L	L	H	H	H	H	H	L	L	H	H	
10	H	L	H	L	H	L	L	L	H	H	L	H	
11	H	L	H	H	H	L	L	H	H	L	L	H	
12	H	H	L	L	H	L	H	L	L	L	H	H	
13	H	H	L	H	H	H	L	L	H	L	H	H	
14	H	H	H	L	H	L	L	L	H	H	H	H	
15	H	H	H	H	H	L	L	L	L	L	L	L	
BI	X	X	X	X	L	L	L	L	L	L	L	L	2

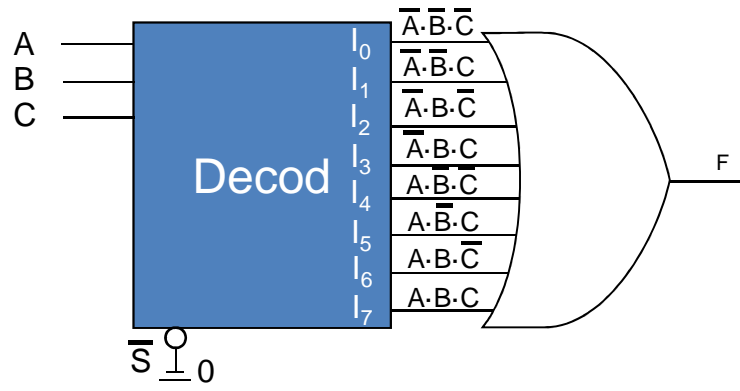
H = high level, L = low level, X = irrelevant

- NOTES:
1. The blanking input (\overline{BI}) must be open or held at a high logic level when output functions 0 through 15 are desired.
 2. When a low logic level is applied directly to the blanking input (BI), all segment outputs are low regardless of the level of any other input.

• Aplicaciones de los decodificadores

a) Generación de funciones

Con un decodificador de 3 entradas y 8 salidas, si se unen las salidas a una puerta OR se obtiene:



$$F = \overline{A} \cdot \overline{B} \cdot \overline{C} + \overline{A} \cdot \overline{B} \cdot C + \overline{A} \cdot B \cdot \overline{C} + \overline{A} \cdot B \cdot C + A \cdot \overline{B} \cdot \overline{C} + A \cdot \overline{B} \cdot C + A \cdot B \cdot \overline{C} + A \cdot B \cdot C$$

Regla general: si la función tiene n variables se necesita un decodificador de n entradas de control para obtener dicha función conectando a la puerta OR las salidas necesarias para obtener dicha función.

Ejemplo: $F = X \cdot Y \cdot Z + X \cdot Y \cdot \overline{Z} + X \cdot \overline{Y} \cdot Z + \overline{X} \cdot \overline{Y} \cdot Z$

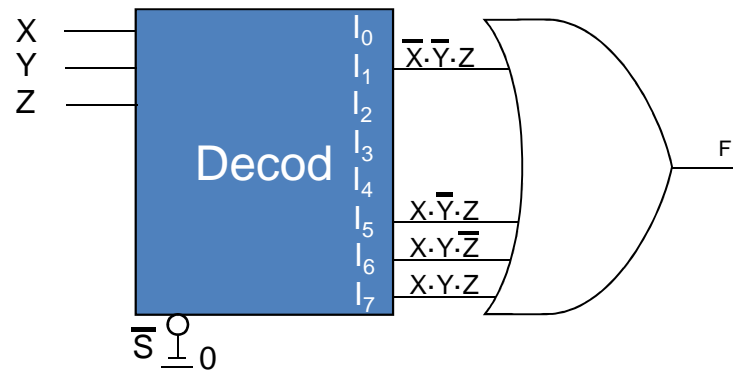
asignando $X=A$, $Y=B$, y $Z=C$

$$X \cdot Y \cdot Z = A \cdot B \cdot C = 7$$

$$X \cdot Y \cdot \overline{Z} = A \cdot B \cdot \overline{C} = 6$$

$$X \cdot \overline{Y} \cdot Z = A \cdot \overline{B} \cdot C = 5$$

$$\overline{X} \cdot \overline{Y} \cdot Z = \overline{A} \cdot \overline{B} \cdot C = 1$$

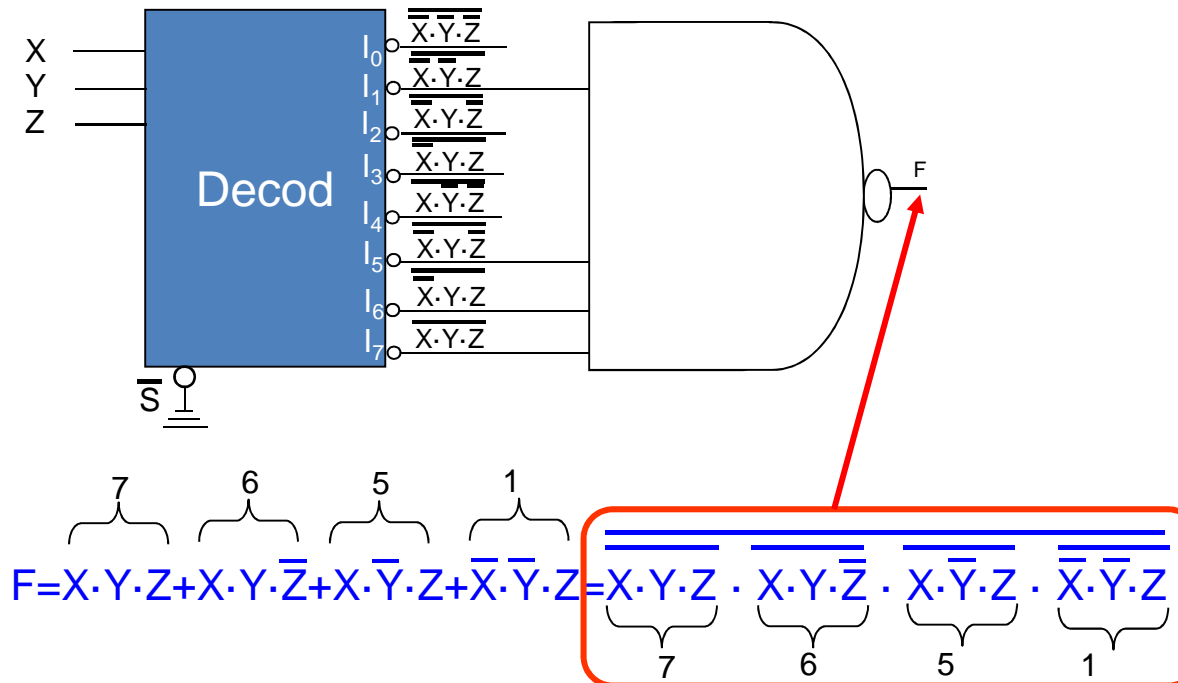


Circuitos y subsistemas combinacionales

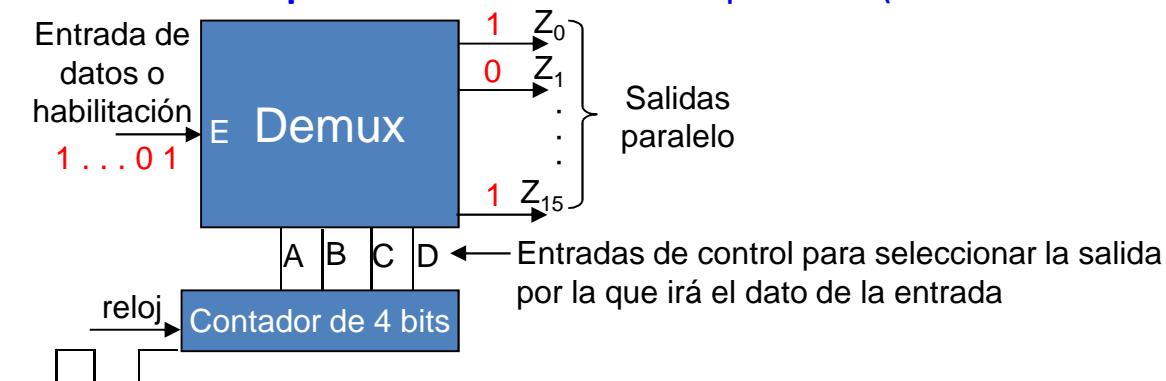
a) Generación de funciones

En cambio si el multiplexor tiene las salidas activa a nivel bajo, éstas se conectan a una puerta NAND

El ejemplo de la página anterior es, con salidas del decodificador activas a nivel bajo:



b) Conversión serie-paralelo: se usan demultiplexores (decodificadores con entrada de habilitación)



Circuitos y subsistemas combinacionales

c) Combinación de un decodificador con un multiplexor

Uniando las salidas de un decodificador a las entradas de un multiplexor se consigue obtener a la salida del multiplexor una función de número de variables igual al número de entradas de control del multiplexor más el número de entradas del decodificador

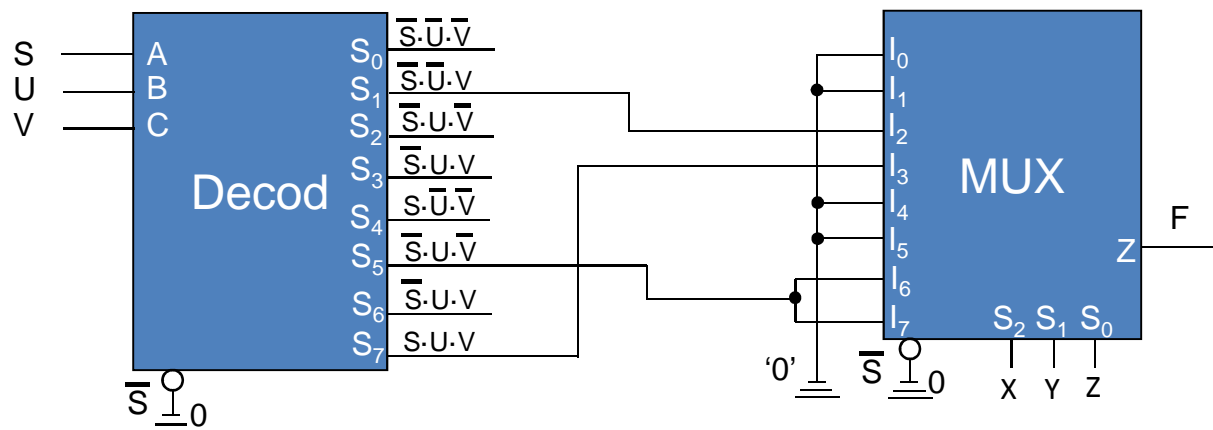
Ejemplo: obtener la siguiente función mediante un decodificador de 3 entradas y un multiplexor de 3 entradas de control:

$$F = S \cdot U \cdot V \cdot \bar{X} \cdot Y \cdot Z + S \cdot \bar{U} \cdot V \cdot X \cdot Y + \bar{S} \cdot \bar{U} \cdot V \cdot \bar{X} \cdot Y \cdot \bar{Z}$$

Se pone la función en forma canónica

$$F = S \cdot U \cdot V \cdot \bar{X} \cdot Y \cdot Z + S \cdot \bar{U} \cdot V \cdot X \cdot Y \cdot (Z + \bar{Z}) + \bar{S} \cdot \bar{U} \cdot V \cdot \bar{X} \cdot Y \cdot \bar{Z}$$

$$F = \underbrace{S \cdot U \cdot V \cdot \bar{X} \cdot Y \cdot Z}_{S_7 I_3} + \underbrace{S \cdot \bar{U} \cdot V \cdot X \cdot Y}_{S_5 I_7} + \underbrace{S \cdot \bar{U} \cdot V \cdot X \cdot Y \cdot Z}_{S_5 I_6} + \underbrace{\bar{S} \cdot \bar{U} \cdot V \cdot \bar{X} \cdot Y \cdot \bar{Z}}_{S_1 I_2}$$

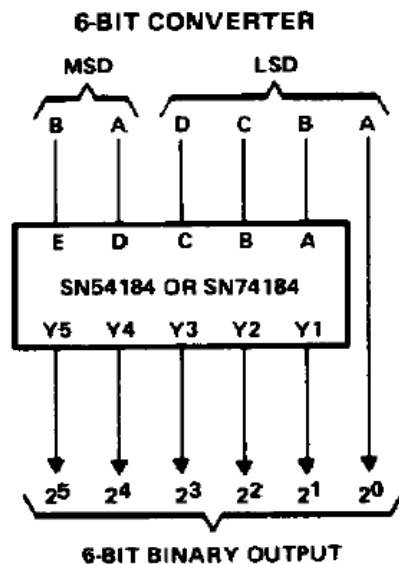


Circuitos y subsistemas combinacionales

- Conversores de código

En el capítulo de circuitos aritméticos se diseñaron conversores de código basados en sumadores. También existen circuitos comerciales que realizan conversión de código

a) Conversor BCD a binario: <http://bit.ly/1zluTdj>



**FUNCTION TABLE
BCD-TO-BINARY
CONVERTER**

BCD WORDS	INPUTS (See Note A)						OUTPUTS (See Note B)					
	E	D	C	B	A	G	Y5	Y4	Y3	Y2	Y1	Y0
0-1	L	L	L	L	L	L	L	L	L	L	L	L
2-3	L	L	L	L	H	L	L	L	L	L	H	L
4-5	L	L	L	H	L	L	L	L	L	H	L	L
6-7	L	L	L	H	H	L	L	L	L	H	H	L
8-9	L	L	H	L	L	L	L	L	H	L	L	L
10-11	L	H	L	L	L	L	L	L	H	L	H	L
12-13	L	H	L	L	H	L	L	L	H	H	L	L
14-15	L	H	L	H	L	L	L	L	H	H	H	L
16-17	L	H	L	H	H	L	L	L	H	L	L	L
18-19	L	H	H	L	L	L	L	L	H	L	L	H
20-21	H	L	L	L	L	L	L	L	H	L	H	L
22-23	H	L	L	L	H	L	L	L	H	L	H	H
24-25	H	L	L	H	L	L	L	L	H	H	L	L
26-27	H	L	L	H	H	L	L	L	H	H	L	H
28-29	H	L	H	L	L	L	L	L	H	H	H	L
30-31	H	H	L	L	L	L	L	L	H	H	H	H
32-33	H	H	L	L	H	L	L	L	L	L	L	L
34-35	H	H	L	H	L	L	L	L	L	L	L	H
36-37	H	H	L	H	H	L	L	L	L	L	H	L
38-39	H	H	H	L	L	L	L	L	L	L	H	H
ANY	X	X	X	X	X	H	L	L	L	L	L	L

H = high level, L = low level, X = irrelevant

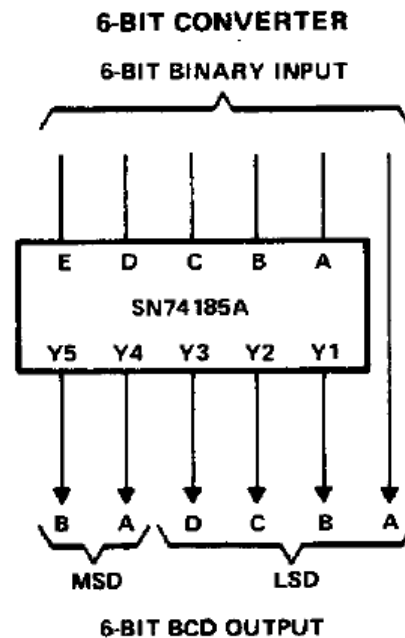
NOTES: A. Input conditions other than those shown produce highs at outputs Y1 through Y5.

B. Outputs Y6, Y7, and Y8 are not used for BCD-to-binary conversion.

Circuitos y subsistemas combinacionales

- Conversores de código

b) Conversor binario a BCD: <http://bit.ly/1zluTdj>



FUNCTION TABLE

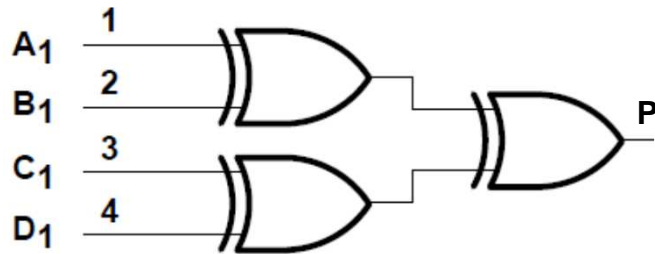
BINARY WORDS	INPUTS						OUTPUTS							
	BINARY SELECT					ENABLE								
	E	D	C	B	A	\overline{G}	Y8	Y7	Y6	Y5	Y4	Y3	Y2	Y1
0-1	L	L	L	L	L	L	H	H	L	L	L	L	L	L
2-3	L	L	L	L	H	L	H	H	L	L	L	L	L	H
4-5	L	L	L	H	L	L	H	H	L	L	L	L	H	L
6-7	L	L	L	H	H	L	H	H	L	L	L	L	H	H
8-9	L	L	H	L	L	L	H	H	L	L	L	H	L	L
10-11	L	L	H	L	H	L	H	H	L	L	H	L	L	L
12-13	L	L	H	H	L	L	H	H	L	L	H	L	L	H
14-15	L	L	H	H	H	L	H	H	L	L	H	L	H	L
16-17	L	H	L	L	L	L	H	H	L	L	H	L	H	H
18-19	L	H	L	L	H	L	H	H	L	L	H	H	L	L
20-21	L	H	L	H	L	L	H	H	L	H	L	L	L	L
22-23	L	H	L	H	H	L	H	H	L	H	L	L	L	H
24-25	L	H	H	L	L	L	H	H	L	H	L	L	H	L
26-27	L	H	H	L	H	L	H	H	L	H	L	L	H	H
28-29	L	H	H	H	L	L	H	H	L	H	L	H	L	L
30-31	L	H	H	H	H	L	H	H	L	H	H	L	L	L
32-33	H	L	L	L	L	L	H	H	L	H	H	L	L	H
34-35	H	L	L	L	H	L	H	H	L	H	H	L	H	L
36-37	H	L	L	H	L	L	H	H	L	H	H	L	H	H
38-39	H	L	L	H	H	L	H	H	L	H	H	H	L	L
40-41	H	L	H	L	L	L	H	H	H	L	L	L	L	L
42-43	H	L	H	L	H	L	H	H	H	L	L	L	L	H
44-45	H	L	H	H	L	L	H	H	H	L	L	L	H	L
46-47	H	L	H	H	H	L	H	H	H	L	L	L	H	H
48-49	H	H	L	L	L	L	H	H	H	L	L	H	L	L
50-51	H	H	L	L	H	L	H	H	H	L	H	L	L	L
52-53	H	H	L	H	L	L	H	H	H	L	H	L	L	H
54-55	H	H	L	H	H	L	H	H	H	L	H	L	H	L
56-57	H	H	H	L	L	L	H	H	H	L	H	L	H	H
58-59	H	H	H	L	H	L	H	H	H	L	H	H	L	L
60-61	H	H	H	H	L	L	H	H	H	H	L	L	L	L
62-63	H	H	H	H	H	L	H	H	H	H	L	L	L	H
ALL	X	X	X	X	X	H	H	H	H	H	H	H	H	H

H = high level, L = low level, X = irrelevant

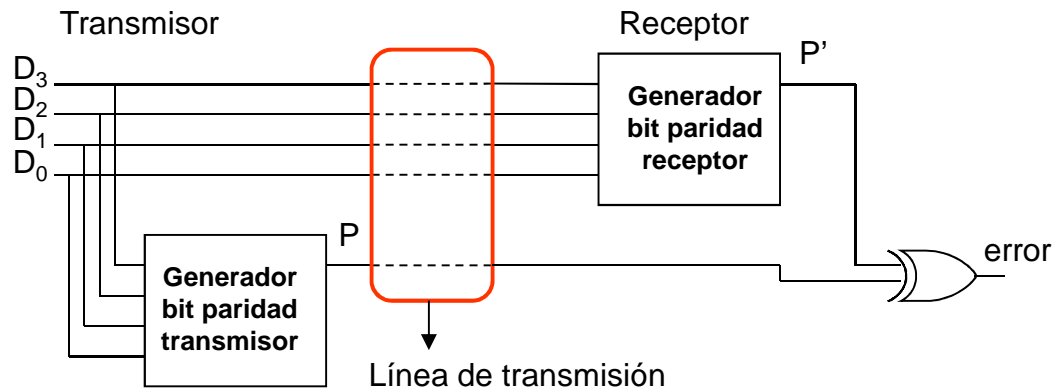
Circuitos y subsistemas combinacionales

- Generador- comprobador de bit de paridad**

Si se tiene un código binario de 4 bit y se desea añadirle un bit de paridad par (se obtiene un 0 cuando el número de 1s es par y un 1 cuando es impar) basta con aplicar la función XOR:



Por otro lado, para detectar un error de bit de paridad en un código de 4 bits + 1 bit de paridad que se recibe, basta con aplicar la función XOR al código recibido sin el bit de paridad y comparar el resultado obtenido con el bit de paridad recibido en el receptor:



Nota: se recomienda revisar la parte final del capítulo 2 sobre detección y corrección de errores

Circuitos y subsistemas combinacionales

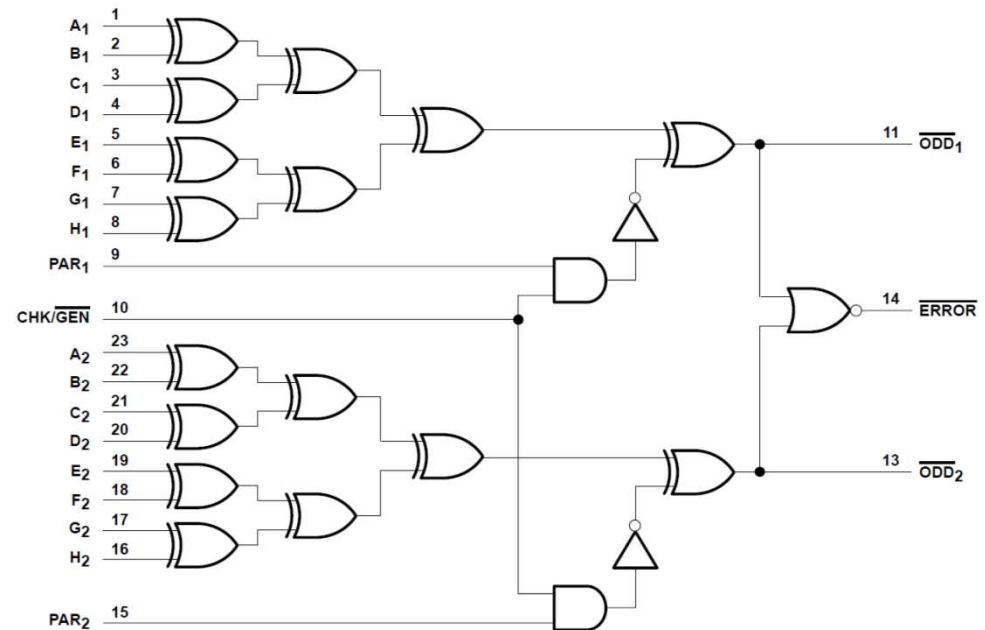
- Circuito comercial CY74FCT480T generador y detector de paridad (<http://bit.ly/1upAj9X>)**
Se emplea para números de 8 bit y la salida ERROR indica cuándo el bit de paridad es incorrecto para los dos códigos introducidos. Esto sirve para avisar de errores en el cálculo del bit de paridad en el transmisor y en el receptor

FUNCTION TABLE

INPUTS					OUTPUTS		
A ₁ -H ₁	A ₂ -H ₂	CHK/GEN	PAR ₁	PAR ₂	$\overline{\text{ODD}}_1$	$\overline{\text{ODD}}_2$	ERROR
Number of A ₁ -H ₁ inputs, high is even	Number of A ₂ -H ₂ inputs, high is even	H	H	H	L	L	H
		H	L	H	H	L	L
		H	H	L	L	H	L
		H	L	L	H	H	L
		L	X	X	H	H	L
	Number of inputs A ₂ -H ₂ , high is odd	H	H	H	L	H	L
		H	L	H	H	H	L
		H	H	L	L	L	H
		H	L	L	H	L	L
		L	X	X	H	L	L
Number of A ₁ -H ₁ inputs, high is odd	Number of A ₂ -H ₂ inputs, high is even	H	H	H	H	L	L
		H	L	H	L	L	H
		H	H	L	H	H	L
		H	L	L	L	H	L
		L	X	X	L	H	L
	Number of A ₂ -H ₂ inputs, high is odd	H	H	H	H	H	L
		H	L	H	L	H	L
		H	H	L	H	L	L
		H	L	L	L	L	H
		L	X	X	L	L	H

H = High logic level, L = Low logic level, X = Don't care

logic diagram

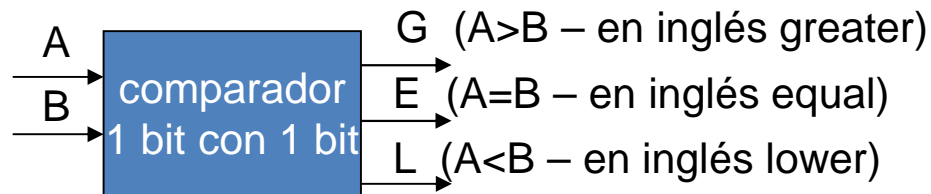


Circuitos y subsistemas combinacionales

- Comparador binario**

a) Concepto general: el comparador binario tiene por objeto determinar si un número es mayor, menor o igual que otro, y se utiliza en la toma de decisiones

Comparador binario de 1 bit con 1 bit:

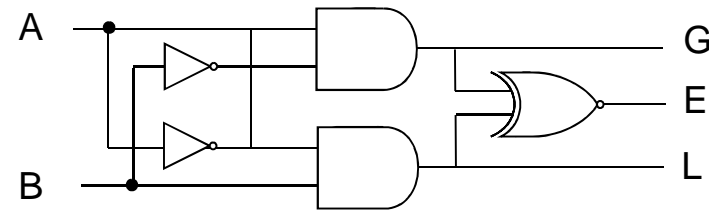


A	B	G	E	L
0	0	0	1	0
0	1	0	0	1
1	0	1	0	0
1	1	0	1	0

$$G = A \cdot \bar{B}$$

$$E = \bar{A} \cdot \bar{B} + A \cdot B = \overline{A \oplus B}$$

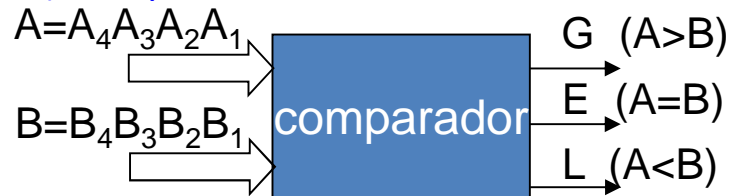
$$L = \bar{A} \cdot B$$



Para números de más bits se van comparando sucesivamente (de más significativo a menos significativo) los dígitos de igual peso de ambos números, hasta encontrar un que es mayor que otro.

Circuitos y subsistemas combinacionales

b) Comparador binario de 4 bit con 4 bit:

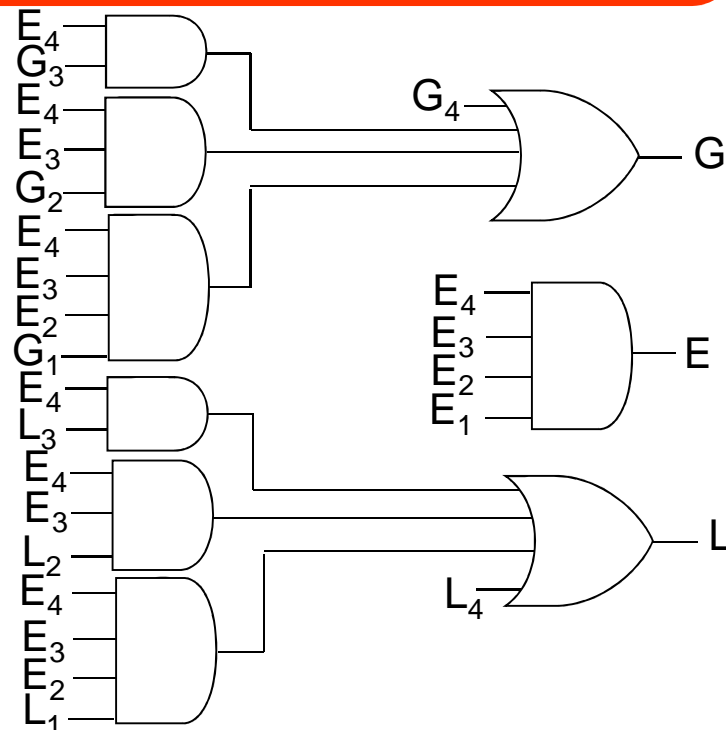
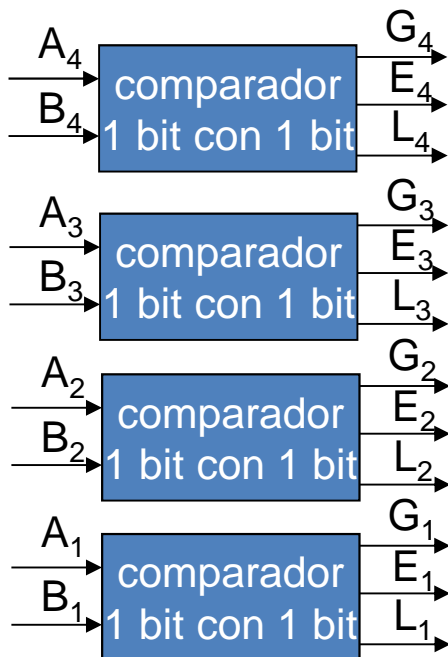


$$G = G_4 + E_4 \cdot G_3 + E_4 \cdot E_3 \cdot G_2 + E_4 \cdot E_3 \cdot E_2 \cdot G_1$$

$$E = E_4 \cdot E_3 \cdot E_2 \cdot E_1$$

$$L = L_4 + E_4 \cdot L_3 + E_4 \cdot E_3 \cdot L_2 + E_4 \cdot E_3 \cdot E_2 \cdot L_1$$

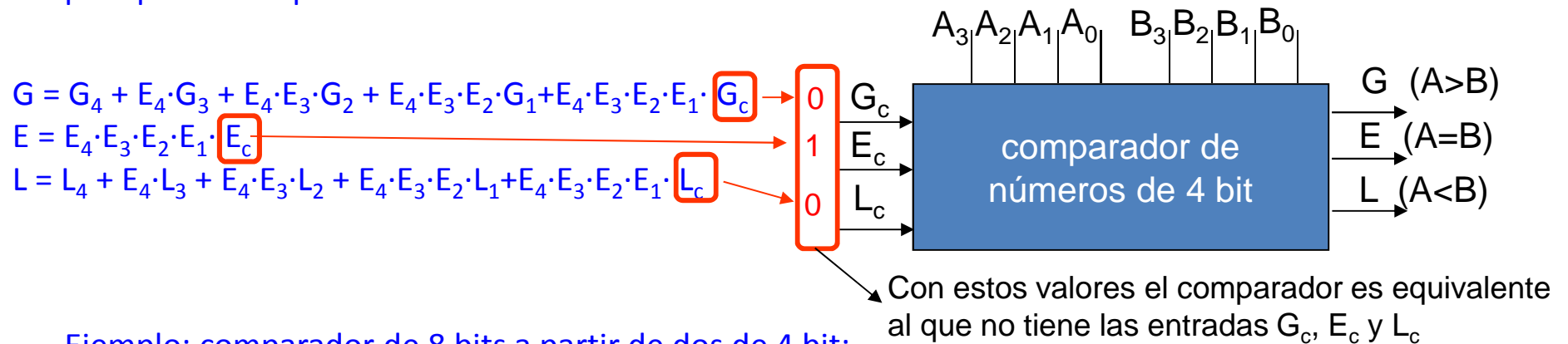
<p>G(A > B) si:</p> <ul style="list-style-type: none"> $A_4 > B_4$ ó $A_4 = B_4$ y $A_3 > B_3$ ó $A_4 = B_4$ y $A_3 = B_3$ y $A_2 > B_2$ ó $A_4 = B_4$ y $A_3 = B_3$ y $A_2 = B_2$ y $A_1 > B_1$ 	<p>L(A < B) si:</p> <ul style="list-style-type: none"> $A_4 < B_4$ ó $A_4 = B_4$ y $A_3 < B_3$ ó $A_4 = B_4$ y $A_3 = B_3$ y $A_2 < B_2$ ó $A_4 = B_4$ y $A_3 = B_3$ y $A_2 = B_2$ y $A_1 < B_1$
---	---



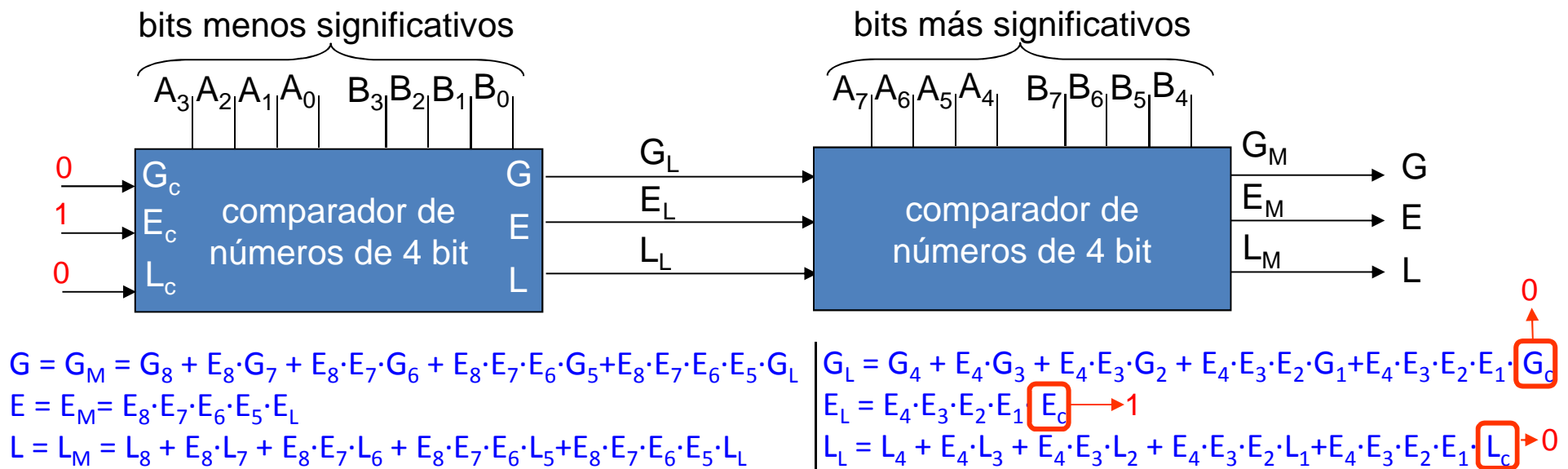
Circuitos y subsistemas combinacionales

Extensión de comparadores con entradas de expansión

Sólo existen comparadores de números de 4 bits. Utilizando tres entradas de expansión G_c , E_c y L_c (análogas a la entrada de acarreo en circuitos sumadores), se pueden conectar comparadores entre sí para poder comparar números de más de 4 bit:



Ejemplo: comparador de 8 bits a partir de dos de 4 bit:

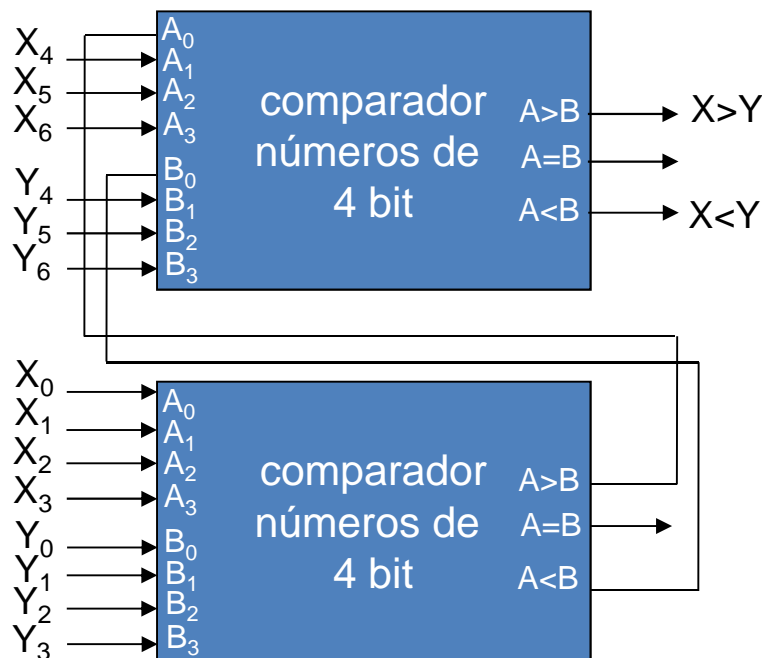


Circuitos y subsistemas combinacionales

- Extensión de comparadores sin entradas de expansión

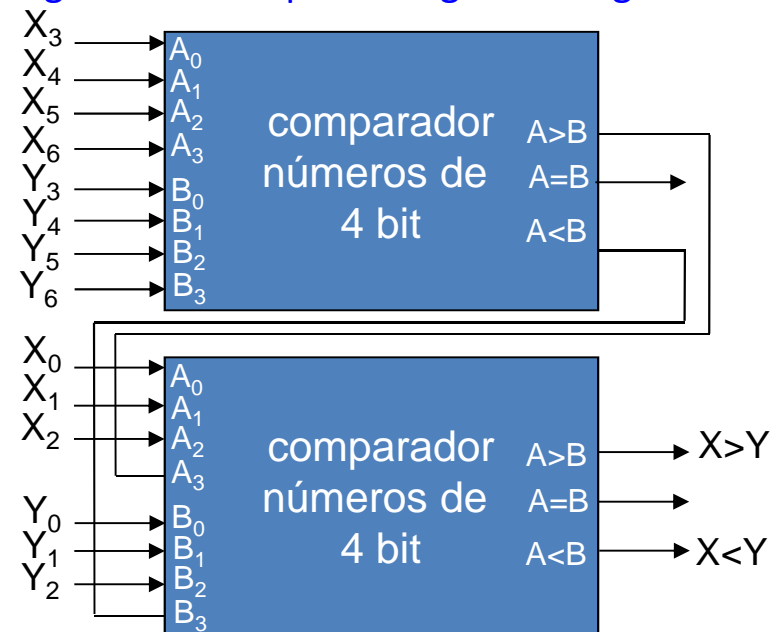
- a) Método ascendente

Se parece al método que utiliza las entradas de expansión. Las salidas del comparador para los 4 bit menos significativos se llevan a los dos bit menos significativos del comparador de los cuatro bit más significativos. De esta manera, sólo se miran las entradas de los bit menos significativos si las de los bit más significativos son iguales



- b) Método descendente

Las salidas del comparador para los 4 bit más significativos se llevan a los dos bit más significativos del comparador de los cuatro bit menos significativos. Es similar al cálculo mental de comparación de números decimales: primero comparamos el dígito más significativo, y si son iguales se compara el siguiente dígito



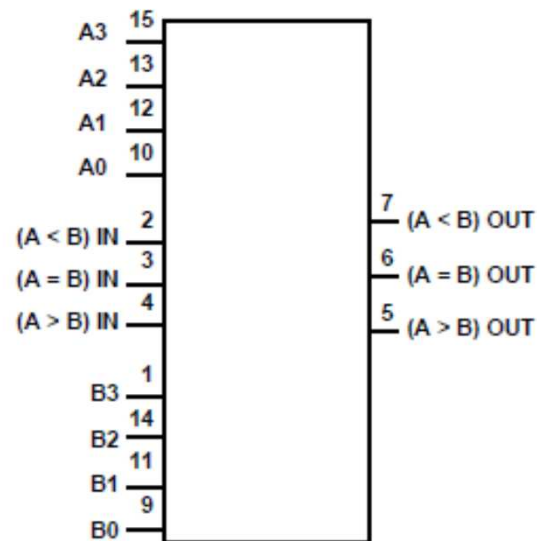
Tamaño máximo de los números binarios con N comparadores:
sin entradas de expansión: $n = 3N + 1$ bits
con entradas de expansión: $n = 4N$ bits

Circuitos y subsistemas combinacionales

- Circuito comparador comercial CD74HC85 (<http://bit.ly/1sfisEK>)**

Pines del chip y tabla de verdad para:

- Un solo dispositivo (comparador de 4 bit con 4 bit)
- Varios dispositivos (conexión mediante entradas de expansión para comparar 8 bit con 8 bit, etc.)
- **Varios dispositivos (conexión mediante entradas de expansión en paralelo)**



TRUTH 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
SINGLE DEVICE OR SERIES CASCADING									
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	L	L	H	L	L	H
PARALLEL CASCADING									
A3 = B3	A2 = B2	A1 = B1	A0 = B0	X	X	H	L	L	H
A3 = B3	A2 = B2	A1 = B1	A0 = B0	H	H	L	L	L	L
A3 = B3	A2 = B2S	A1 = B1	A0 = B0	L	L	L	H	H	L

H = High Voltage Level, L = Low Voltage, Level, X = Don't Care

Circuitos y subsistemas combinacionales

- **Circuito comparador comercial CD74HC85 (<http://bit.ly/1sfisEK>)**

- conexión mediante entradas de expansión en paralelo:

El circuito CD74HC85 está preparado para conexión paralelo, lo que reduce el tiempo de respuesta del dispositivo.

En la figura de la derecha se puede mediante sólo 2 etapas obtener un comparador de 24 bit, mientras que con el método de conexión en serie harían falta 6 comparadores y cada comparador supondría una etapa.

Como ejemplo, si con la familia LS85 el tiempo de respuesta para 24 bits es de 48 ns, mediante conexión en serie el tiempo es el triple (6 etapas frente a 2 etapas), es decir, de 144 ns.

