

# Sistemas automáticos, circuitos digitales y combinacionales: Circuitos combinacionales

---



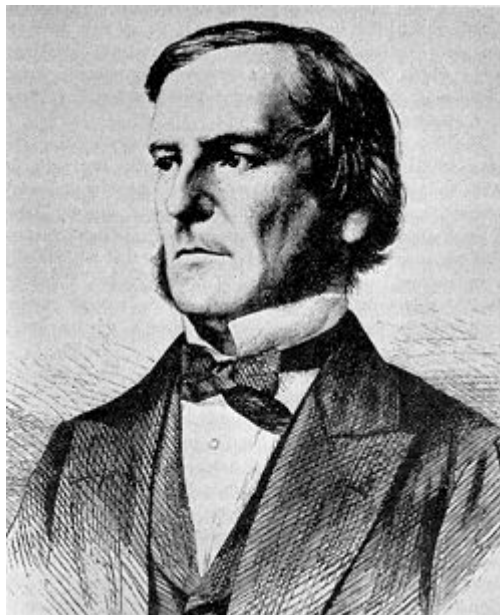
**PAC**  
**Preparación Acceso a CFGS**

**Tecnología Industrial**  
**Contenidos**

**Sistemas automáticos, circuitos digitales y combinacionales:**  
**Circuitos combinacionales.**

El álgebra de Boole se denomina así en honor a **George Boole**, matemático inglés 1815 - 1864, que fue el primero en definirla como parte de un sistema lógico, a mediados del siglo XIX. El álgebra de Boole fue un intento de utilizar las técnicas algebraicas para tratar expresiones de la lógica proposicional.

El álgebra opera con **variables booleanas**, que son aquellas que sólo pueden tomar dos valores (0 y 1), estos valores no representan números si no estados. Ejemplo: pueden simbolizar si un interruptor está abierto (0), o cerrado (1), si conduce o no conduce, si hay tensión o no.



George Boole

Imagen de Hask en [Wikimedia](#). Dominio público

En la actualidad, esta herramienta se aplica de forma generalizada en el ámbito del diseño de control electrónico de procesos industriales. Estos métodos de control se basan en el uso de sistemas denominados puertas lógicas.

Hemos estado estudiando anteriormente las características generales de los circuitos digitales.

También hemos presentado un protocolo de actuación para su diseño basándonos en unas determinadas condiciones de funcionamiento, a partir de la definición del problema lógico, su tabla de verdad, su función canónica, su simplificación por métodos algebraicos o por los diagramas de Karnaugh, y su implementación por medio de cualquier tipo de puertas o bien por medio de puertas universales por aplicación de los teoremas y postulados del Álgebra de Boole.

Estos circuitos que hemos estado aprendiendo a construir se llaman **combinacionales** porque el estado de sus salidas depende única y exclusivamente de la combinación que toman sus variables de entrada, sin que importen los estados anteriores de las variables ni el tiempo.

En este tema vamos a estudiar una serie de **circuitos combinacionales** que son muy **comunes** y aparecen o bien aisladamente o formando parte de otros circuitos más complejos de aplicación general, que se repiten un número de veces tan considerable que se hace aconsejable su fabricación en serie y que son los siguientes:

1. Codificadores.
2. Decodificadores.
3. Convertidores de código.
4. Multiplexores.
5. Demultiplexores.
6. Comparadores.
7. Sumadores.
8. Restadores.
9. Detectores/generadores de paridad.
10. Otros circuitos lógicos: ALU

# 1. Codificadores

Un codificador es un circuito combinacional integrado que tiene hasta  $2^n$  entradas y  $n$  salidas y la función que desempeña es mostrar en la salida la combinación correspondiente al código binario de la entrada activada.

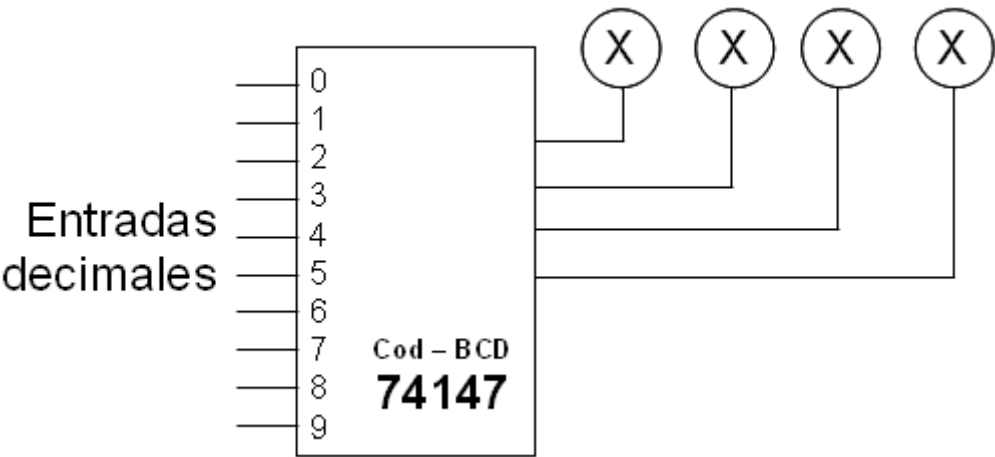


Imagen de elaboración propia

Se entenderá mejor con un ejemplo: En una calculadora cuando pulsamos cualquiera de las diez teclas numéricas de una calculadora estamos marcando un número decimal, pero la calculador opera con número en binario. Para expresar en binario del 1 al 10, necesitamos al menos cuatro bits, ya que con tres solamente podríamos establecer  $2^3 = 8$  combinaciones posibles (es decir del 0 al 7) y no podríamos codificar los diez dígitos necesarios (faltarían el 8 y el 9).

Por tanto emplearemos 4 salidas. Como con 4 salidas (4 bits) tenemos 16 combinaciones y empleamos 10 (del 0 al 9), o bien dejaremos seis combinaciones sin emplear, o las utilizaremos para codificar cualquier otra función representada en alguna de las teclas de la calculadora (el +, el -, el ·, el ÷, el = y la √; por ejemplo)

La tabla de verdad del codificador será:

Entradas										Salidas			
A <sub>0</sub>	A <sub>1</sub>	A <sub>2</sub>	A <sub>3</sub>	A <sub>4</sub>	A <sub>5</sub>	A <sub>6</sub>	A <sub>7</sub>	A <sub>8</sub>	A <sub>9</sub>	S <sub>0</sub>	S <sub>1</sub>	S <sub>2</sub>	S <sub>3</sub>
0	0	0	0	0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0	1	0	0	0	0	1
0	0	0	0	0	0	0	1	0	0	0	0	1	0
0	0	0	0	0	0	1	0	0	0	0	0	1	1
0	0	0	0	0	1	0	0	0	0	0	1	0	0
0	0	0	0	1	0	0	0	0	0	0	1	0	1
0	0	0	1	0	0	0	0	0	0	0	1	1	0
0	0	1	0	0	0	0	0	0	0	0	1	1	1
0	1	0	0	0	0	0	0	0	0	1	0	0	0
1	0	0	0	0	0	0	0	0	0	1	0	0	1

A partir de la tabla se deduce que la salida S<sub>1</sub> será 1 si lo es la entrada A<sub>9</sub>, ó la A<sub>7</sub>, ó la A<sub>5</sub>, ó la A<sub>3</sub>, ó la A<sub>1</sub>, de ahí que la ecuación lógica que corresponde a esta salida sea la suma de las entradas 1, 3, 5, 7 y 9. Si seguimos analizando la tabla obtendremos, de forma análoga, las ecuaciones que tienen que cumplir las salidas S<sub>2</sub>, S<sub>3</sub> y S<sub>4</sub>.

En el caso de se activasen más de una entrada estaríamos ante el dilema de ¿qué entrada debería codificarse?, o se produciría una señal de error en la salida, por ello los codificadores pueden ser sin prioridad, (no suelen emplearse), y los codificadores con prioridad, generalmente a la entrada más significativa, en este caso la tabla de verdad sería:

Entradas										Salidas			
A <sub>0</sub>	A <sub>1</sub>	A <sub>2</sub>	A <sub>3</sub>	A <sub>4</sub>	A <sub>5</sub>	A <sub>6</sub>	A <sub>7</sub>	A <sub>8</sub>	A <sub>9</sub>	S <sub>0</sub>	S <sub>1</sub>	S <sub>2</sub>	S <sub>3</sub>
0	0	0	0	0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0	1	x	0	0	0	1
0	0	0	0	0	0	0	1	x	x	0	0	1	0
0	0	0	0	0	0	1	x	x	x	0	0	1	1
0	0	0	0	0	1	x	x	x	x	0	1	0	0
0	0	0	0	1	x	x	x	x	x	0	1	0	1
0	0	0	1	x	x	x	x	x	x	0	1	1	0
0	0	1	x	x	x	x	x	x	x	0	1	1	1
0	1	x	x	x	x	x	x	x	x	1	0	0	0
1	x	x	x	x	x	x	x	x	x	1	0	0	1

Es decir si por cualquier circunstancia se activase más de una entrada simultáneamente, el codificador presentará en la salida la correspondiente al código de la entrada que tenga asignado un mayor peso, es decir la más significativa, resultando indiferente los valores que tomasen las otras entradas menos significativas.

En la figura adjunta se muestra el circuito integrado combinacional correspondiente a un codificador con prioridad de 9 entradas y cuatro salidas.

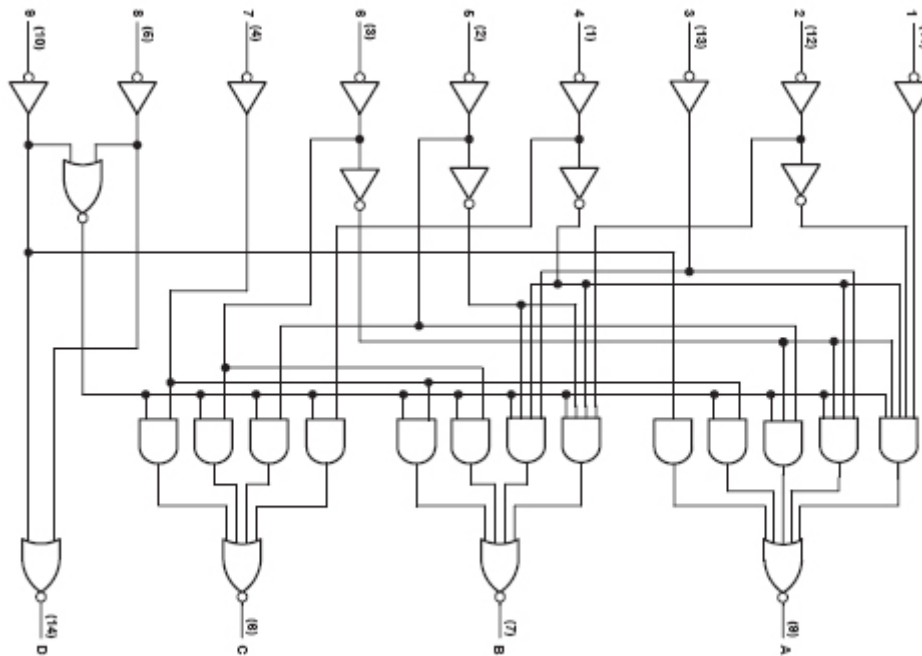


Imagen de elaboración propia

Como ya explicamos en el tema anterior las puertas lógicas y los circuitos que explicaremos en este tema se comercializan en circuitos integrados (CI), que son como una "pastilla de plástico" de la que salen unas patillas de conexión llamadas "pines", cada uno de los cuales corresponde a una entrada o salida de datos, alimentación o tierra; y cuya identificación se hace a partir de una pequeña muesca en la cápsula que marca el número de patilla. En las llamadas datasheet de los fabricantes de chips, dibujos como el siguiente indican la y correspondencia entre los pins de CI y las entradas y salidas del circuito, en este caso el codificador.

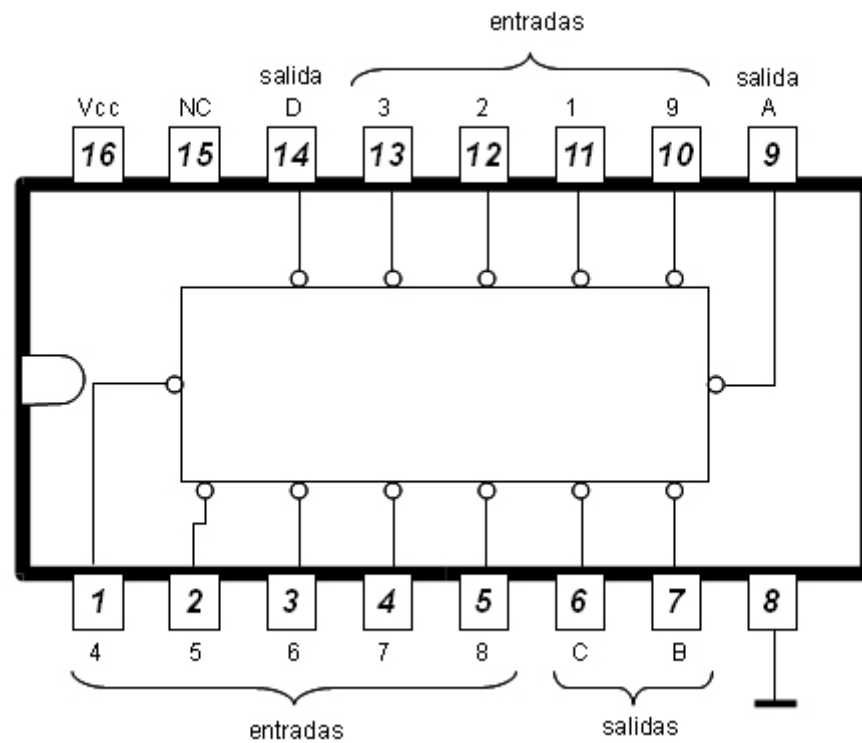


Imagen de elaboración propia

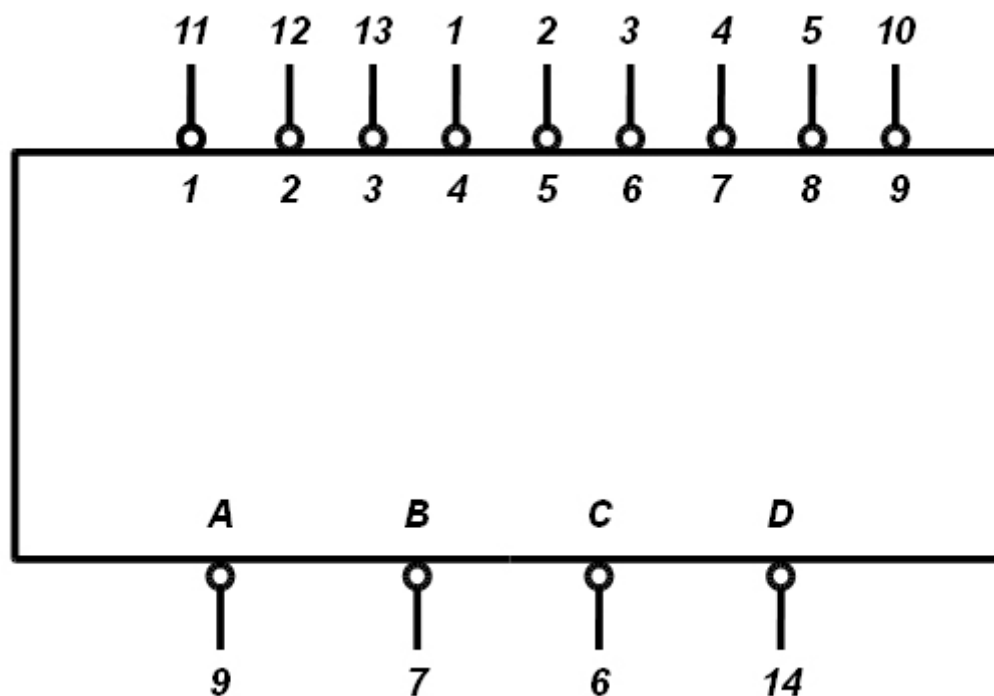


Imagen de elaboración propia

Este tipo de codificadores se emplean en la codificación de los teclados convencionales, así mismo en los circuitos conversores analógico-digital, y para controlar posibles perturbaciones en los ordenadores.

Aunque la aplicación más significativa de este tipo de circuitos integrados es en la construcción de multiplexadores, que son unos circuitos combinacionales que estudiaremos posteriormente.

## 2. Decodificadores

Son circuitos combinatoriales integrados que disponen de **n** entradas y un número de salidas igual o menor a **2<sup>n</sup>**, actúan de modo que según cual sea la combinación de las variables de entrada se activa una única salida, permaneciendo el resto de ellas desactivada.

Suelen disponer de una entrada adicional denominada de inhibición o strobe de modo que cuando esta entrada se encuentra activada, pone todas las salidas a 0.

Por ejemplo un decodificador de 2 entradas y  $2^2=4$  salidas, tendría la siguiente tabla de verdad:

Entradas		Salidas			
E1	E0	S3	S2	S1	S0
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	0	0	0

Sus ecuaciones lógicas serían:

$$S_0 = \overline{E_1} \cdot \overline{E_0}$$

$$S_1 = \overline{E_1} \cdot E_0$$

$$S_2 = E_1 \cdot \overline{E_0}$$

$$S_3 = E_1 \cdot E_0$$

Que una vez implementado con puertas NOT y AND, quedaría:

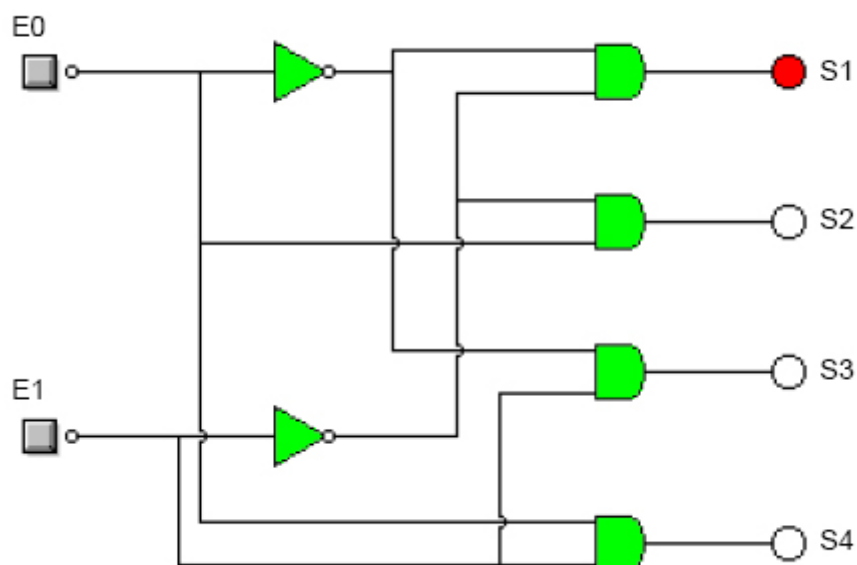


Imagen de elaboración propia

Un ejemplo típico es el **decodificador BCD a decimal**, cuya tabla de verdad será:

E3	E2	E1	E0	S9	S8	S7	S6	S5	S4	S3	S2	S1	S0
0	0	0	0	0	0	0	0	0	0	0	0	0	1
0	0	0	1	0	0	0	0	0	0	0	0	1	0
0	0	1	0	0	0	0	0	0	0	0	1	0	0
0	0	1	1	0	0	0	0	0	0	1	0	0	0
0	1	0	0	0	0	0	0	0	1	0	0	0	0
0	1	0	1	0	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	0	1	0	0	0	0	0	0
0	1	1	1	0	0	1	0	0	0	0	0	0	0
1	0	0	0	0	1	0	0	0	0	0	0	0	0
1	0	0	1	1	0	0	0	0	0	0	0	0	0

Los decodificadores pueden ser de dos tipos:

- **No Excitadores.** Se denominan así a un tipo de decodificadores cuyas salidas solo pueden acoplarse a otros circuitos digitales de la misma familia integrada, ya que dan una corriente muy pequeña en dichas salidas, incapaz de activar ningún otro componente.
- **Decodificadores Excitadores.** Son aquellos cuyas salidas dan suficiente corriente como para atacar, no solo a otros circuitos integrados de la misma familia, sino también a otros dispositivos, tales como displays, lámparas, relés, transductores,...

Un decodificador muy común es el de siete segmentos, este circuito combinacional activa simultáneamente varias salidas, decodifica la información de entrada en BCD a un código de siete segmentos adecuado para que se muestre en un **display de siete segmentos**, es el procedimiento empleado en todas las calculadoras, los relojes digitales,...

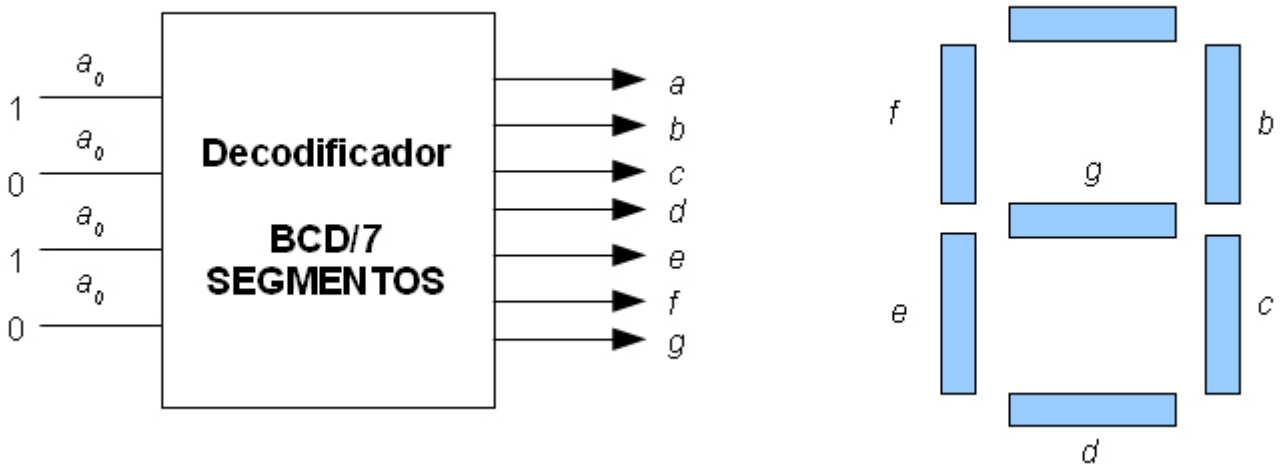


Imagen de elaboración propia

Su tabla de verdad sería:

Entradas				Segmentos							Decimal
A <sub>3</sub>	A <sub>2</sub>	A <sub>1</sub>	A <sub>0</sub>	a	b	c	d	e	f	g	
0	0	0	0	1	1	1	1	1	1	0	0
0	0	0	1	0	1	1	0	0	0	0	1
0	0	1	0	1	1	0	1	1	0	1	2
0	0	1	1	1	1	1	1	0	0	1	3
0	1	0	0	0	1	1	0	0	1	1	4
0	1	0	1	1	0	1	1	0	1	1	5
0	1	1	0	0	0	1	1	1	1	1	6
0	1	1	1	1	1	1	0	0	0	0	7
1	0	0	0	1	1	1	1	1	1	1	8
1	0	0	1	1	1	1	0	0	1	1	9

En las imágenes siguientes puedes ver, qué segmentos están encendidos en dos números, el 6 y el 7. En el 7, como indica la tabla de verdad, están encendidos el a, el b y el c (encendido quiere decir en este caso valor lógico a 1)

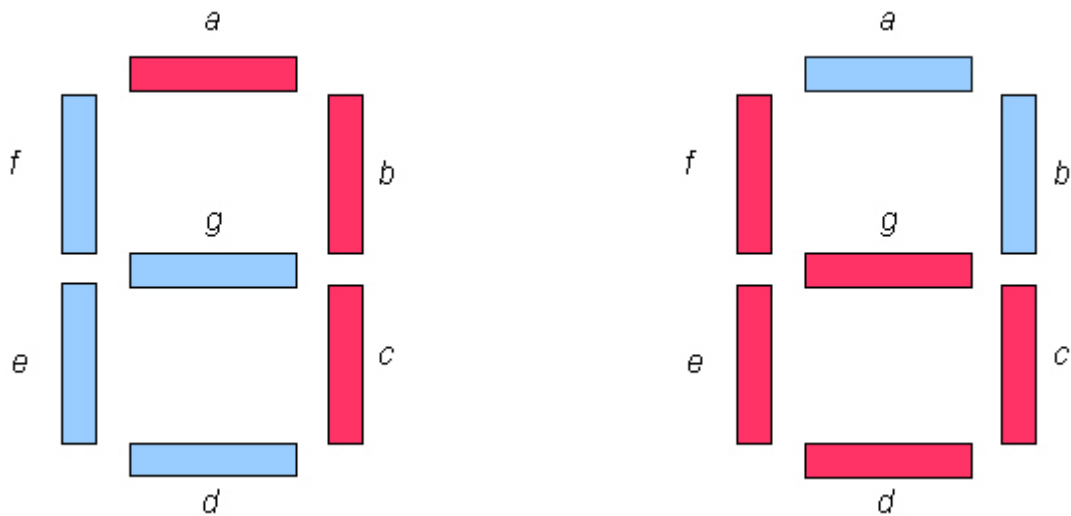


Imagen de elaboración propia

En el 6, están apagados el a y el b (apagado quiere decir valor lógico a cero)

### Aplicacion de los decodificadores: Implementación de funciones lógicas con decodificadores

Una de las principales aplicaciones de los decodificadores es que permite implementar ecuaciones correspondientes al funcionamiento de una función lógica.

*Ejemplo.* A partir de la tabla de la verdad siguiente:



Decimal	Entradas C B A			Salida S
0	0	0	0	0
1	0	0	1	1
2	0	1	0	0
3	0	1	1	1
4	1	0	0	1
5	1	0	1	0
6	1	1	0	0
7	1	1	1	1

La función canónica será

$$S = \overline{C}\overline{B}A + \overline{C}BA + C\overline{B}\overline{A} + CBA$$

Para implementar la función lógica empleando un decodificador, actuaremos del siguiente modo:

En primer lugar se debe utilizar un decodificador que tenga igual o mayor número de líneas de entrada que el número de variables de la función lógica. En nuestro ejemplo un decodificador de cuatro a diez líneas, con salidas activas a nivel bajo y conectando a masa la entrada de mayor peso

Luego, miramos las salidas del decodificador que hacen que la salida de la función sea 1, según la tabla de verdad, corresponde a las salidas:

$$S_1=001, S_3=011, S_4=100 \text{ y } S_7=111$$

Ahora debemos sumar estos términos para conseguir la función lógica deseada, para lo que emplearemos el tipo de puertas adecuado que podrán ser:

- Puertas OR, si hemos utilizado decodificadores con salidas activas en nivel alto, ya que la función se debe activar cuando sea 1, uno o varios de los términos que constituyen la función.
- Puertas NAND, si hemos utilizado decodificadores con salidas activas en nivel bajo, ya que la función se debe activar cuando sea 0 uno o varios de los términos que constituyen la función.

En caso de que alguna combinación de la tabla de verdad que provocan 1 en la salida de la función no tuviera correspondencia con las salidas del decodificador, se confeccionarán mediante puertas lógicas la combinación correspondiente, llevándose la salida de esta combinación junto a la del circuito, a una puerta OR final.

## Ejercicio resuelto

Te proponemos dos ejemplos para intentar mostrar el uso de decodificadores como dispositivos para implementar funciones lógicas

### Ejemplo 1

Implementar utilizando un decodificador el circuito correspondiente a una función lógica  $F(A,B,C)$ , que debe cumplir que será cero cuando las tres variables de entrada estén a nivel bajo, o cuando la variable B se encuentre en estado alto si A no lo está. En los demás casos la función dará un 1 en la salida.

#### Mostrar retroalimentación

Abre la solución en pdf [aquí](#).

### Ejemplo 2

Se desea implementar empleando un decodificador, un circuito correspondiente a una función lógica en la que mediante tres sensores (A,B,C) controlen el estado de tres

avisadores  $(X,Y,Z)$ , de modo que se cumplan las siguientes premisas.

- cuando se active únicamente el sensor A, no habrá indicación alguna.
- cuando se active únicamente el sensor B, actuara el indicador Z.
- cuando se active únicamente el sensor C, actuaran los indicadores X e Y.
- cuando se activen únicamente A y B, actuara X.
- cuando se activen únicamente B y C, actuaran X y Z.
- cuando se activen únicamente A y C, actuara Y.
- cuando se activen simultáneamente A, B y C, los tres actuadores estarán a 0.
- En caso de inactividad de los sensores, la indicación será nula.

### Mostrar retroalimentación

Abre la solución en pdf [aquí](#).

### Un último ejemplo: Control de lámparas de un semáforo

Con 2 elementos para transmitir órdenes, se pueden conseguir 4 ( $2^2$ ) instrucciones distintas, a esto es a lo que llamamos, información codificada en binario.

Un semáforo debe tener tres salidas que serán cada una de las tres lámparas, (se podría añadir una cuarta salida que indicase avería), este lo podemos conseguir con un decodificador de dos entradas y cuatro salidas, con lo que en cada instante solamente puede estar activada una única lámpara, y el circuito de control semafórico que diseñemos solamente deberá disponer de dos salidas.

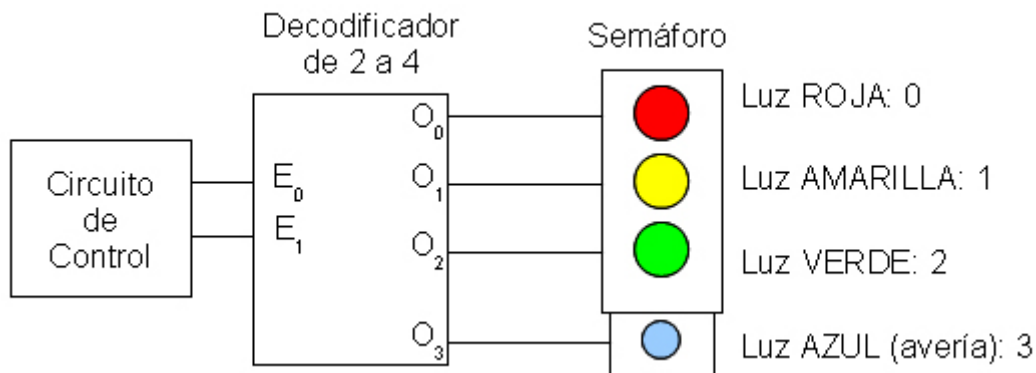


Imagen de elaboración propia

Si el sistema de control envía el número 0 codificado, se activará la luz roja, que está asociada a esa combinación de las variables de entrada, y así sucesivamente.

### 3. Convertidores de código

Son circuitos combinacionales cuya función es cambiar los datos de un código binario a otro, esto es así porque para determinadas operaciones de transmisión y procesamiento de información son más eficaces unos códigos que otros. Se suelen implementar mediante dispositivos lógicos programables.

Vamos a ver un ejemplo de un cambiador de **código de BCD** (8421) a binario exceso 3.

La tabla de verdad será:

Decimal	Entradas				Salidas (Exceso a3)			
	8 E <sub>3</sub>	4 E <sub>2</sub>	2 E <sub>1</sub>	1 E <sub>0</sub>	S <sub>3</sub>	S <sub>2</sub>	S <sub>1</sub>	S <sub>0</sub>
0	0	0	0	0	0	0	1	1
1	0	0	0	1	0	1	0	0
2	0	0	1	0	0	1	0	1
3	0	0	1	1	0	1	1	0
4	0	1	0	0	0	1	1	1
5	0	1	0	1	0	1	1	0
6	0	1	1	0	1	0	0	1
7	0	1	1	1	1	0	1	0
8	1	0	0	0	1	0	1	1
9	1	0	0	1	1	1	0	0

Las funciones canónicas serían:

$$\begin{aligned}S_0 &= \overline{E_3}\overline{E_2}\overline{E_1}\overline{E_0} + \overline{E_3}\overline{E_2}E_1\overline{E_0} + \overline{E_3}E_2\overline{E_1}\overline{E_0} + \overline{E_3}E_2E_1\overline{E_0} + E_3\overline{E_2}\overline{E_1}\overline{E_0} \\S_1 &= \overline{E_3}\overline{E_2}\overline{E_1}E_0 + \overline{E_3}\overline{E_2}E_1E_0 + \overline{E_3}E_2\overline{E_1}\overline{E_0} + \overline{E_3}E_2\overline{E_1}E_0 + \overline{E_3}E_2E_1E_0 + E_3\overline{E_2}\overline{E_1}E_0 \\S_2 &= \overline{E_3}\overline{E_2}\overline{E_1}E_0 + \overline{E_3}\overline{E_2}E_1\overline{E_0} + \overline{E_3}\overline{E_2}E_1E_0 + \overline{E_3}E_2\overline{E_1}\overline{E_0} + \overline{E_3}E_2\overline{E_1}E_0 + E_3\overline{E_2}\overline{E_1}E_0 \\S_3 &= \overline{E_3}E_2E_1\overline{E_0} + \overline{E_3}E_2E_1E_0 + E_3\overline{E_2}\overline{E_1}\overline{E_0} + E_3\overline{E_2}\overline{E_1}E_0\end{aligned}$$

Ahora deberíamos simplificar e implementar la función, pero en este caso es un proceso largo que no vamos a hacer aquí y que la razón por la que estos circuitos se implementan en circuitos combinacionales integrados

## SN74LS151

### LOGIC DIAGRAM

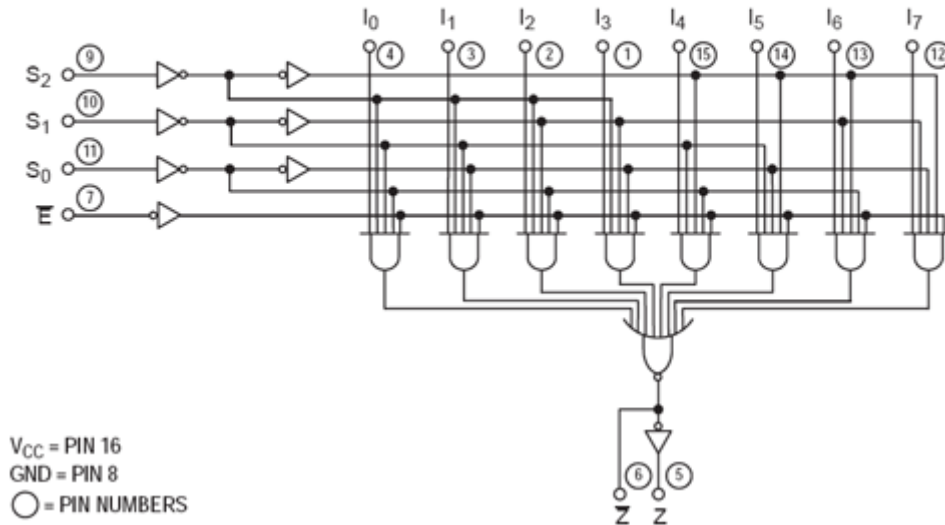


Imagen de la hoja de datos del multiplexor 74151

El encapsulado de este circuito, es decir la forma comercial del circuito integrado que lo aloja, es la de la imagen siguiente:

## SN74LS151

### CONNECTION DIAGRAM DIP (TOP VIEW)

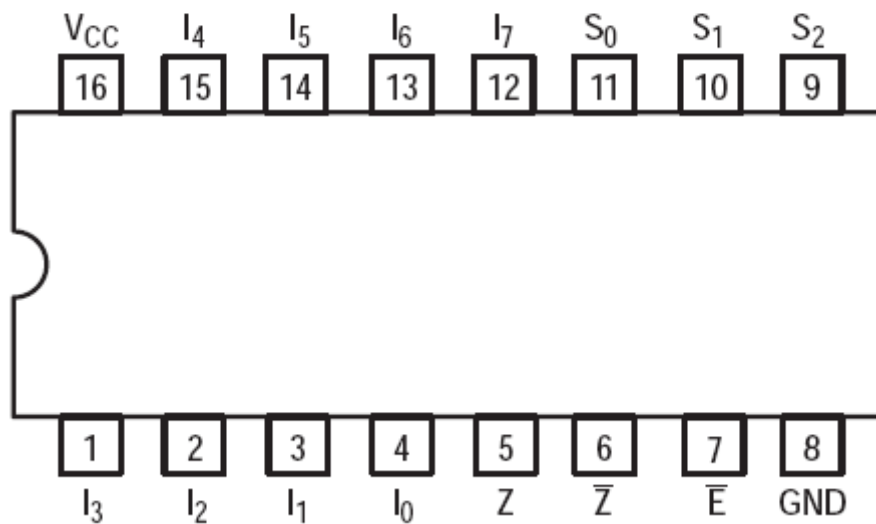


Imagen del encapsulado del multiplexor 74151

### Curiosidad

El multiplexor más sencillo es el bplexer, utilizado en la transmisión-recepción por radar, para conectar alternativamente una única antena al circuito emisor de ondas y al receptor de ondas reflejadas, como se muestra en la figura.

Empleado para transmitir y recibir alternadamente

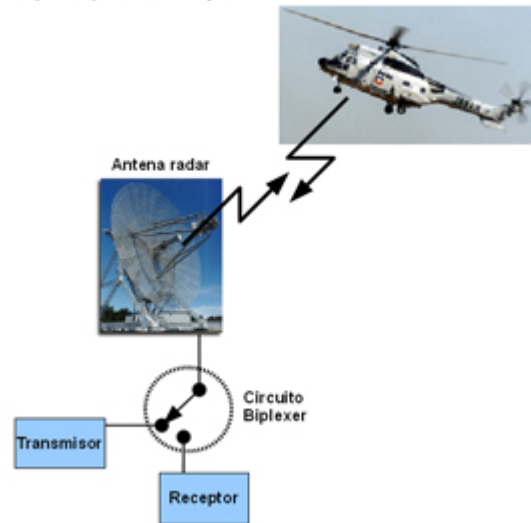


Imagen de elaboración propia

## Curiosidad

Son muy utilizados los multiplexores en los displays de calculadoras y relojes electrónicos, ya que consumen mucha menos potencia, lográndose disminuir el consumo de corriente y reduciéndose el número de pins que deben llegar hasta el circuito activador.

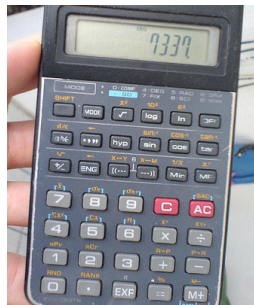


Imagen de E. Posadas en [Flickr](#). CC

### Aplicaciones de los multiplexores: Implementación de funciones lógicas con multiplexores

Una de las principales aplicaciones de los multiplexores es que permite implementar ecuaciones correspondientes al funcionamiento de una función lógica, reemplazando con un solo chip gran cantidad de cableado y de circuitos integrados.

Para lo que se debe seguir el siguiente protocolo:

Se conectan a las entradas de selección las variables de entrada del problema.

Se conectan las entradas de dato a 1 o a 0, según convenga a la configuración escogida con la entrada de selección.

Cuando no disponemos de suficientes entradas de selección en un multiplexor para conectar con las entradas del problema, podemos continuar empleándolos, estableciendo en las entradas de dato los valores adecuados correspondientes con ayuda de lógica adicional y en función de la variable que no conectamos a la entrada de selección.

*Ejemplo: Empleo de multiplexores de igual número de entradas de control que de variables a implementar.*

Supongamos, por ejemplo, que queremos implementar la siguiente función:

$$F = \overline{A}\overline{B}\overline{C}D + \overline{A}\overline{B}CD + \overline{A}B\overline{C}\overline{D} + \overline{A}BCD + A\overline{B}\overline{C}\overline{D} + A\overline{B}C\overline{D} + AB\overline{C}\overline{D} + ABC\overline{D}$$

La función tiene cuatro variables de entradas, A, B, C, D, con lo que existen combinadas, dan lugar a 16 combinaciones posibles. Empleando un multiplexor de 4 entradas de control, se dispondrán de 16 canales de datos, por lo tanto uno por cada posible combinación de las variables de entrada de la función lógica.

Cada término que constituye la función corresponde a las de cada combinación de las variables de entrada que hacen 1 dicha función, por lo que si aplicamos las variables de la función a las entradas de selección y conectamos a 1 los canales de entrada que se corresponden con las combinaciones que intervienen en la función, poniendo a 0 el resto de los canales, tendremos la función implementada.

*Ejemplo: Empleo de multiplexores en funciones con un número de entradas de control inferior en una unidad al de variables de la función a implementar.*

Es posible implementar funciones lógicas de n variables con multiplexores de n-1 entradas de control, lo que producirá el consiguiente ahorro económico.

Con el ejemplo del apartado anterior, confeccionamos la siguiente tabla, donde se agrupan por columnas todas las posibles combinaciones de tres de las variables de entrada B, C y D, dejando en las filas las posibilidades de la variable que resta A.

BCD \ A	000	001	010	011	100	101	110	111
0	0	1	0	1	1	1	1	1
1	0	1	0	0	1	0	1	0
	0	1	0	$\overline{A}$	1	$\overline{A}$	1	$\overline{A}$

Por tanto, la implementación del circuito se consigue aplicando las variables b, c y d a las tres entradas de selección del multiplexor y conectando las entradas de los canales de la siguiente forma:

- Canales 0 y 2 conectado a 0.
- Canales 1, 4 y 6 conectado a 1.
- Canales 3, 5 y 7 a través de un inversor a la variable a, ya que su valor es siempre el contrario del de dicha variable.

## Ejercicio resuelto

Abre los siguientes enlaces, en los que podrás ver cuatro ejercicios resueltos para acabar de explicar la implementación de funciones lógicas con multiplexores.

- [Multiplexores. Ejercicio resuelto 1](#)
- [Multiplexores. Ejercicio resuelto 2](#)
- [Multiplexores. Ejercicio resuelto 3](#)
- [Multiplexores. Ejercicio resuelto 4](#)

### **Mostrar retroalimentación**

Puedes descargar los cuatro ejercicios en un mismo [archivo aquí](#).

## 4. Multiplexores

El multiplexor es el circuito lógico combinacional equivalente a un interruptor mecánico giratorio de varias posiciones, tal como el componente que sirve para seleccionar las bandas de un receptor de radio.

Permite dirigir la información binaria procedente de diversas fuentes a una única línea de salida, para ser transmitida a través de ella, a un destino común.

Disponen de: hasta  $2^n$  líneas de entrada de datos, una **única** de salida y **n** entradas de selección; que habilitan y ponen en contacto uno de los terminales de entrada de datos con el de salida.

El circuito combinacional integrado multiplexor, suele tener: 8 entradas de datos (bits), 3 entradas de selección (address) y una única salida e datos.

Por ejemplo cuando en las entradas de selección está activa la combinación 010 Equivalente a la entrada de información número 2, en la salida aparecerá el bit que en ese momento haya en la entrada 2 es decir un 1, ya que esta es la entrada que hemos seleccionado para comunicarla con la salida.

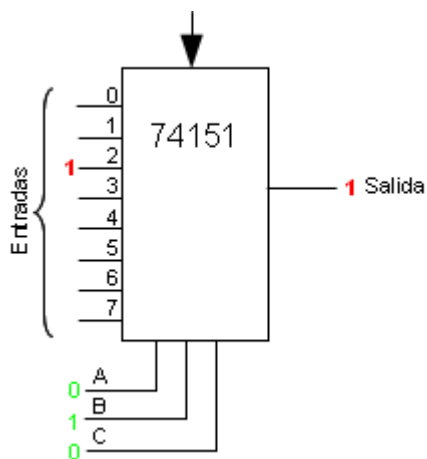


Imagen de elaboración propia

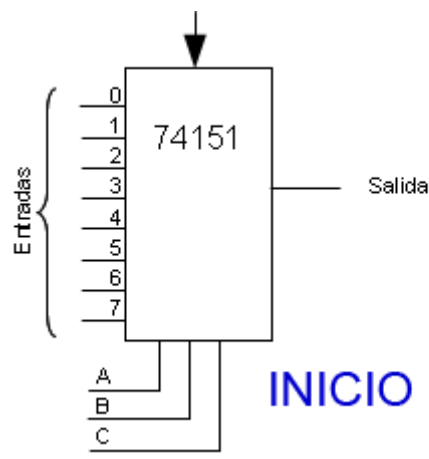


Imagen de elaboración propia



Se puede observar el conexionado en la figura siguiente , obtenida de su hoja de datos, del multiplexor 74151, con las tres entradas de selección ( $S_2$ ,  $S_1$  y  $S_0$ ), las ocho entradas de datos ( $I_7$ ,  $I_6$ ,  $I_5$ ,  $I_4$ ,  $I_3$ ,  $I_2$ ,  $I_1$  e  $I_0$ ), Una entrada de inhibición ( $\bar{E}$ ) una salida ( $Y$ ) y otra salida más que es la negada de la anterior ( $\bar{Y}$ )



## 6. Comparadores

Son circuitos integrados combinacionales con uno o más pares de entradas que tienen como función comparar dos magnitudes binarias para determinar su relación.

El comparador más básico, que determina si dos números son iguales, se consigue mediante una puerta XOR (or exclusiva), ya que su salida es 1 si los dos bits de entrada son diferentes y 0 si son iguales.

Muchos comparadores poseen además de la salida de igualdad, dos salidas más que indican cual de los números colocados a la entrada es mayor (M) que el otro, o bien es menor (m) que el otro.

Vamos a implementar un circuito comparador de dos bits empleando puertas elementales, para lo que, en primer lugar escribiremos su tabla de verdad.

Entradas A B	Salidas I M m
0 0	1 0 0
0 1	0 0 1
1 0	0 1 0
1 1	1 0 0

Con lo que las funciones canónicas serán:

$$\begin{aligned}
 M &= A\bar{B} \\
 m &= \bar{A}B \\
 I &= AB + \bar{A}\bar{B}
 \end{aligned}$$

O bien como  $I=1$  solo cuando  $M=m=0$

$$I = \overline{M+m} = \overline{A\bar{B} + \bar{A}B}$$

Que equivale a leer los 0 de la tabla de verdad de I.

Con lo que al implementar la función lógica quedará el siguiente circuito:

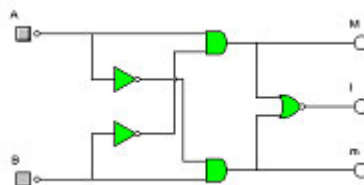


Imagen de elaboración propia

Generalmente estos circuitos combinacionales no suelen cablearse, vienen en [circuitos integrados](#) como por ejemplo el CI 7485, que es un comparador de 4 bits. Posee 3 entradas en cascada que permiten utilizar varios comparadores para comparar números binarios de más de 4 bits:

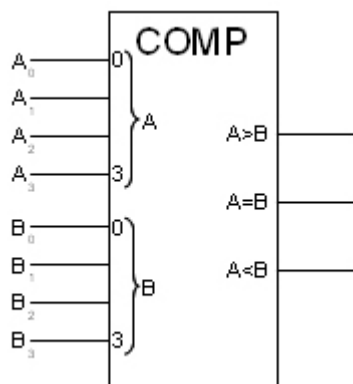


Imagen de elaboración propia

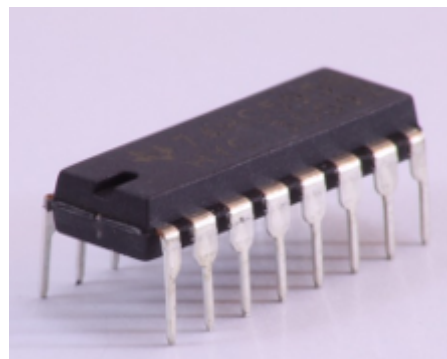


Imagen en [Pixabay](#). Dominio público

Se usa un comparador para los 4 bits menos significativos de los 2 números y se aplica su salida a la entrada en cascada del siguiente que compara los bits de mayor significación, proporcionando el resultado final.

## 7. Sumadores

Un sumador es un circuito que realiza la suma de dos palabras binarias. Es distinta de la operación OR, con la que no nos debemos confundir. La operación suma de números binarios tiene la misma mecánica que la de números decimales.

Por lo que en la suma de números binarios con dos o más bits, puede ocurrir el mismo caso que podemos encontrar en la suma de números decimales con varias cifras: cuando al sumar los dos primeros dígitos se obtiene una cantidad mayor de 9, se da como resultado el dígito de menor peso y "me llevo" el anterior a la siguiente columna, para sumarlo allí.

En la suma binaria de los dígitos  $1 + 1$ , el resultado es 0 y me llevo 1, que debo sumar en la columna siguiente y pudiéndose escribir 10, solamente cuando sea la última columna a sumar. A este bit más significativo de la operación de sumar, se le conoce en inglés como carry (acarreo), equivalente al "me llevo una" de la suma decimal.

**Semisumador.** Es un dispositivo capaz de sumar dos bits y dar como resultado la suma de ambos y el acarreo. La tabla de verdad correspondiente a esta operación sería:

Entradas		Salidas	
A	B	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

Con lo que sus funciones canónicas serán:

$$C = AB$$

$$S = \bar{A}B + A\bar{B} = A \oplus B$$

Que una vez implementado con puertas lógicas, un semisumador tendría el circuito:

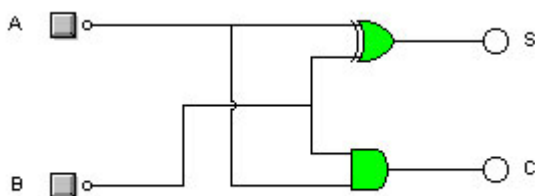


Imagen de elaboración propia

**Sumador completo.** Presenta tres entradas, dos correspondientes a los dos bits que se van a sumar y una tercera con el acarreo de la suma anterior. Y tiene dos salidas, el resultado de la suma y el acarreo producido. Su tabla de verdad será:

Entradas			Salidas	
A	B	C <sup>-1</sup>	C	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

1	1	1	1	0
---	---	---	---	---

Sus funciones canónicas serán:

$$S = \overline{A}\overline{B}C^{-1} + \overline{A}B\overline{C^{-1}} + A\overline{B}C^{-1} + ABC^{-1}$$

$$C = \overline{A}BC^{-1} + A\overline{B}C^{-1} + ABC^{-1}$$

Que una vez simplificadas quedarían:

$$S = A \oplus B \oplus C^{-1}$$

$$C = AB + AC^{-1} + BC^{-1}$$

O bien:

$$C = AB + (A \oplus B)C^{-1}$$

Una vez implementado con puertas lógicas el sumador presentaría cualquiera de los siguientes circuitos:

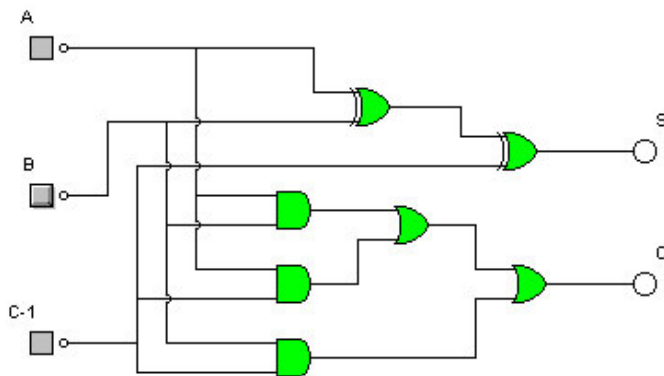


Imagen de elaboración propia

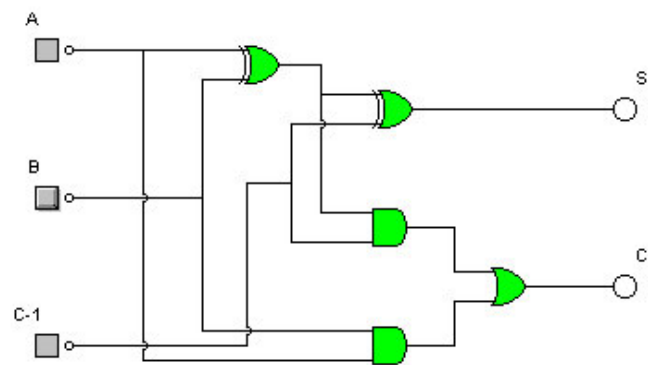
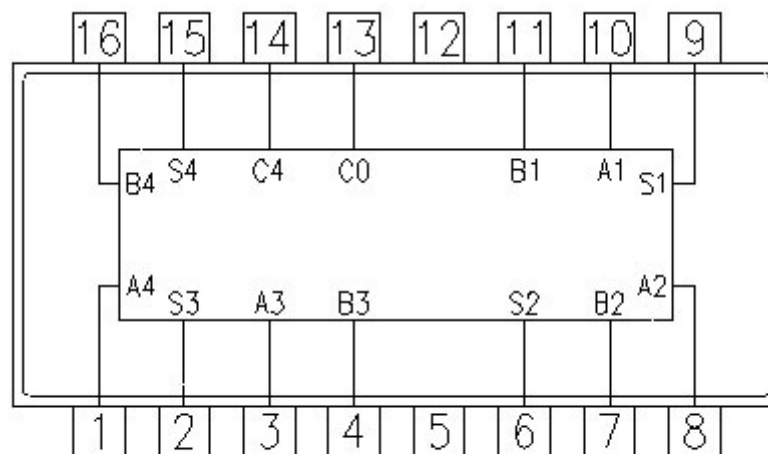


Imagen de elaboración propia

Aunque, como ya hemos dicho en otros casos, en realidad estos circuitos no se cablean con puertas lógicas, si no que forman parte de circuitos integrados como el CI 7483, que es un sumador de cuatro bits.



7483

Imagen de elaboración propia

El esquema

## 5. Demultiplexores

Son circuitos integrados combinacionales que realizan la función opuesta a un multiplexor. Es decir tiene una única entrada de datos,  $n$  entradas de selección y un número de salidas  $<2^n$ , de modo que según introduzcamos una combinación u otra por las entradas de selección, así conseguimos comunicar la entrada de datos con la salida seleccionada.

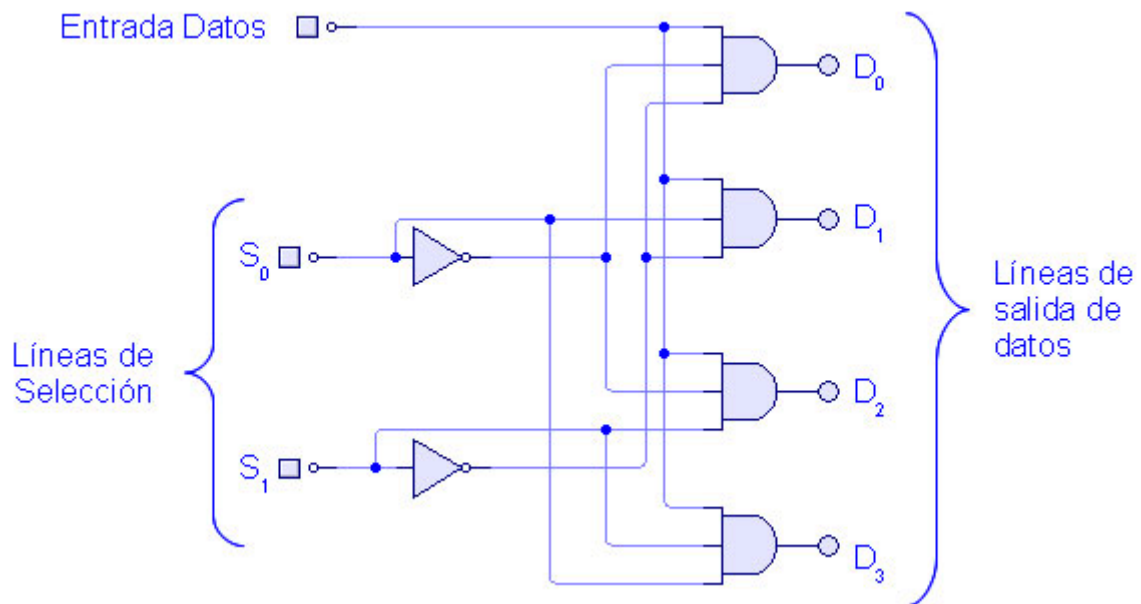


Imagen de elaboración propia

Los demultiplexores se pueden utilizar como decodificadores, de binario a decimal o a hexadecimal. Si se coloca en las entradas de selección (address) un número binario, se obtiene en la salida seleccionada el estado correspondiente que tengamos en la entrada de datos

En el siguiente vídeo, se muestra el funcionamiento del demultiplexor.



*Curiosidad*

### ¿Sabías qué ...?

Los demultiplexores junto con los multiplexores se emplean para poder llevar varias conversaciones telefónicas simultaneas por una misma línea, por lo que se reducen significativamente los costes, y se simplifican muy sensiblemente el cableado de conexiones

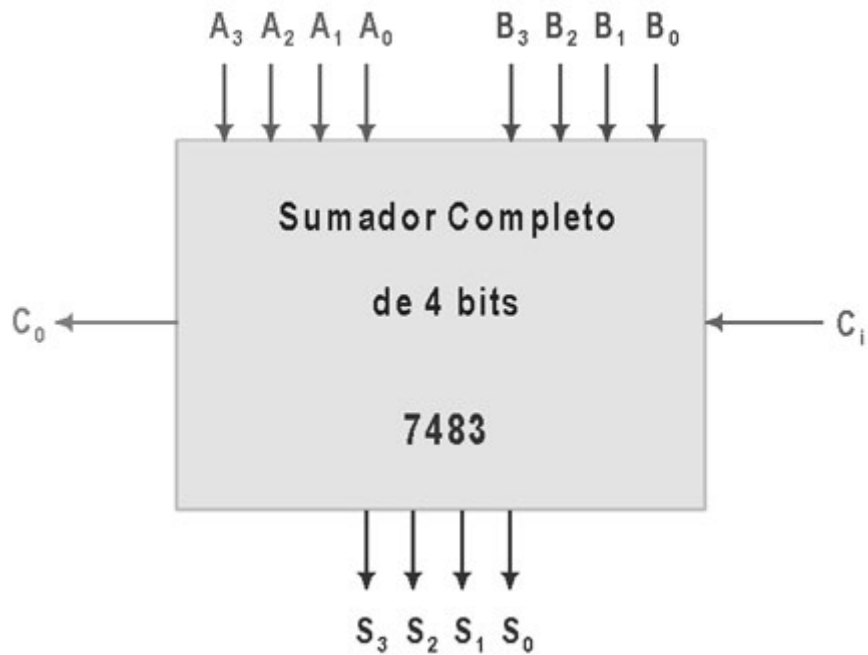


Imagen de elaboración propia

Para sumar números de más de un bit, también se recurre al conexionado de sumadores binarios en paralelo, donde el acarreo de la suma de dos dígitos será una entrada a sumar en el paso siguiente. En este caso se precisan tantos semisumadores como bits tengamos que sumar. El montaje de la figura posterior tiene un funcionamiento idéntico al del CI 7483, aunque presenta incompatibilidades a nivel de pines.

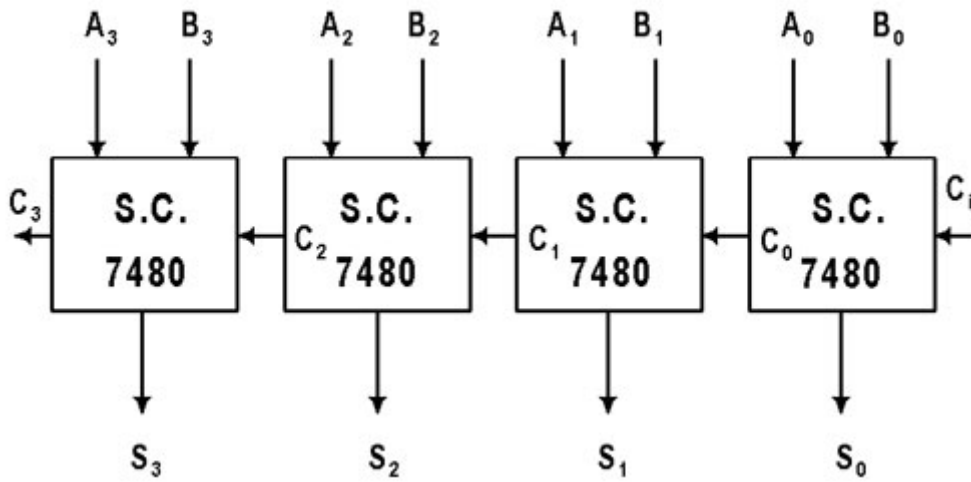


Imagen de elaboración propia

## 8. Restadores

De modo similar a lo comentado con el sumador, podríamos construir un semi-restador en el que las entradas serán M = minuendo, S = sustraendo, y las salidas D = diferencia, P = cifra prestada. Debe cumplir la siguiente tabla de verdad:

Entradas		Salidas	
M	S	D	P
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

Con lo que sus funciones canónicas serán:

$$D = \overline{M}S + M\overline{S} = M \oplus S$$

$$P = \overline{A}B$$

Cuya posible implementación se muestra en la figura:

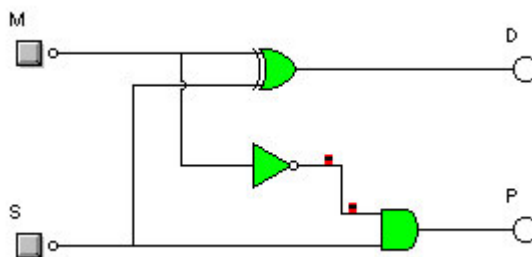


Imagen de elaboración propia

En realidad este circuito no existe ya que para realizar restas se emplean sumadores, puesto que una resta de dos números es igual a la suma de uno con el negativo del otro. Para lo que se utiliza el método de complemento a uno (invertir todos los bits uno a uno, es decir cambiando 1 por 0 y 0 por 1), o bien el método de complemento a dos, añadiéndole un bit de signo aunque no vamos a explicar este método de operar porque no se ajusta a los objetivos del tema.

### Curiosidad

Ni restadores, ni multiplicadores, ni divisores... con sumadores hago de todo!!

Como ya hemos comentado antes, es lo mismo restar, que sumar número de signo opuesto, por lo que no hace falta un restador para la operación resta. Pero, ¿y para multiplicar y dividir?

Lo vemos en un ejemplo muy sencillo,

$$\text{Restar: } 10 - 4 = 10 + (-4)$$

$$\text{Multiplicar: } 10 \times 4 = 10 + 10 + 10 + 10$$

$$\text{Dividir: } \frac{4}{10} = 4 \times \frac{1}{10} = \frac{1}{10} + \frac{1}{10} + \frac{1}{10} + \frac{1}{10}$$

Es decir, con números Reales, podemos sumar, restar, multiplicar y dividir con una sola operación, la suma. Por tanto en cuanto tenemos un sumador, podemos hacer cualquier otra operación sin problemas





## 9. Detectores/generadores de paridad

Los circuitos electrónicos digitales se basan en la transmisión y el procesamiento de información, lo que hace necesario verificar que la información recibida es igual a la emitida; no suelen producirse errores, por lo que cuando ocurren en la mayoría de los casos el error en la transmisión se produce en un único bit.

El método más sencillo y eficaz de comprobación de la transmisión de datos consiste en añadir a la información transmitida un bit más, con la misión de que el número de 1 transmitidos en total sea par (paridad par), o impar (paridad impar).

Detectores/generadores de paridad

Los generadores de paridad par son aquellos circuitos que generan un 0 cuando el número de 1 en la entrada es par y un 1 cuando es impar, en el caso de dos bit, sería como se muestra en la tabla de verdad:

Entradas		Salidas	
A	B	P	I
0	0	0	1
0	1	1	0
1	0	1	0
1	1	0	1

P = paridad par, es decir un número de 1 par.

I = paridad impar, es decir un número de 1 impar.

Las funciones canónicas serán:

$$P = \overline{A}B + A\overline{B} = A \oplus B$$
$$I = AB + \overline{A}\overline{B} = A \otimes B = \overline{P}$$

Cuya posible implementación se muestra en la figura:

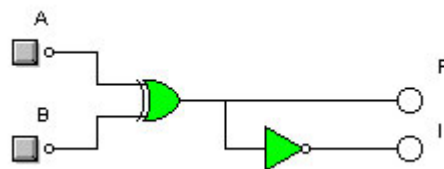


Imagen de elaboración propia

Como venimos comentando a lo largo de todo el tema estos circuitos no se suelen cablear, sino que se presentan como circuitos integrados, un ejemplo de generadores de paridad sería el CI 74180.

## 10. Otros circuitos lógicos: ALU

ALU son las siglas de Arithmetic Logic Unit, es decir, Unidad Lógico Aritmética.

Se trata de un circuito integrado con la capacidad de realizar diferentes operaciones aritméticas y lógicas (es decir, del álgebra de Boole), con dos palabras de  $n$  bits. Se pueden encontrar como circuitos independientes, y también como bloque funcional dentro de los microprocesadores y microcontroladores.

En general, las operaciones matemáticas están codificadas en binario natural y en complemento a 2 para las restas, pero se pueden codificar en otros códigos, como por ejemplo BCD natural.

El más conocido es 74LS181, que es una ALU de 4 bits, que puede realizar hasta 32 funciones diferentes (16 lógicas y 16 aritméticas), trabaja con números binarios de 4 bits, aunque se pueden conectar en cascada para aumentar el número de bits. Este circuito integrado tiene como entradas:

- Los cuatro bits del operando A.
- Los cuatro bits del operando B.
- Entradas de selección (para seleccionar la operación a realizar, entre 16).
- Entrada de acarreo, por si viene de un integrado con el resultado de menor peso.
- Entrada de control, para seleccionar si la operación a realizar debe ser aritmética o lógica.

Como salidas tiene los 4 bits del resultado, más una salida comparador ( $A = B$ ) y salidas de acarreo.

Como curiosidad decir que este circuito integrado trabaja con lógica inversa en las entradas de datos y en las salidas, es decir, que para estos pines se invierte el significado de los 1 y los 0. Aunque es posible hacerlo trabajar con lógica directa.

Configuración de pines del CI 74LS181

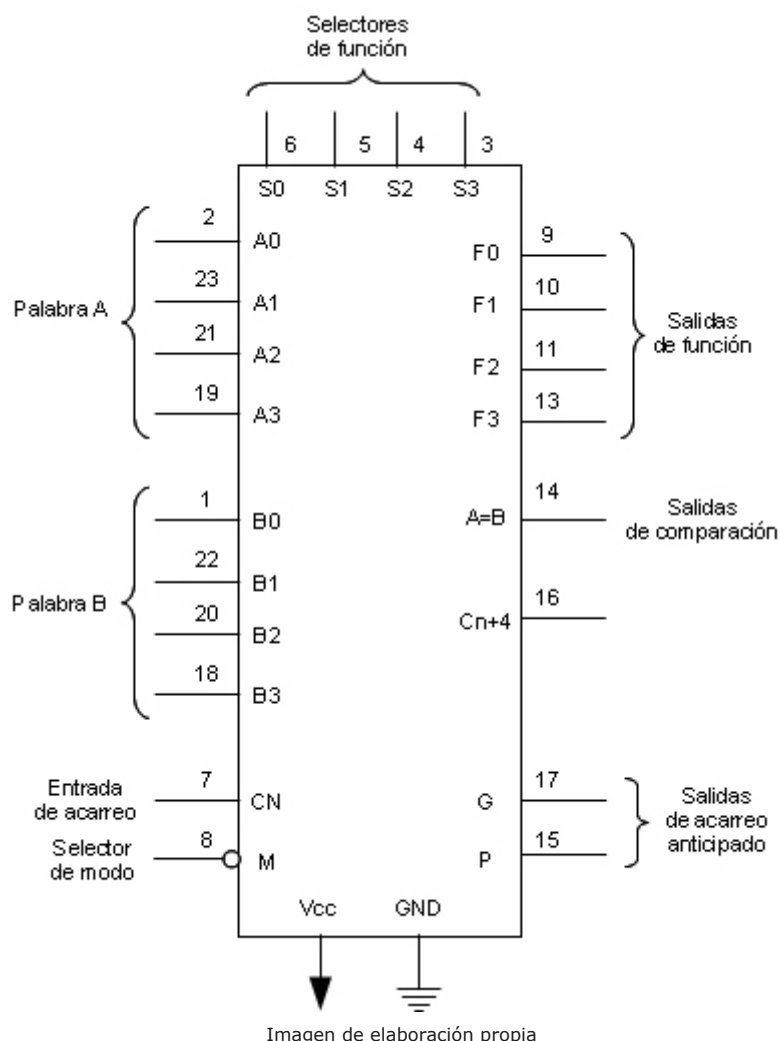


Tabla con las funciones que pueden realizarse con el 74LS181

Códigos de Selección				Funciones lógicas	Funciones aritméticas*	
					M=0	
S3	S2	S1	S0	M=1	Cn=1(sin acarreo)	Cn=0 (con  carreo)
0	0	0	0	$\overline{A}$	$A$	$A+1$
0	0	0	1	$\overline{A+B}$	$A+B$	$(A+B)+1$
0	0	1	0	$\overline{AB}$	$A+\overline{B}$	$(A+\overline{B})+1$
0	0	1	1	0	-1	0
0	1	0	0	$\overline{AB}$	$A+A\overline{B}$	$A+A\overline{B}+1$
0	1	0	1	$\overline{B}$	$(A+B)+A\overline{B}$	$(A+B)+A\overline{B}+1$
0	1	1	0	$A\oplus B$	$A-B-1$	$A-B$
0	1	1	1	$A\overline{B}$	$A\overline{B}-1$	$A\overline{B}$
1	0	0	0	$\overline{A}+B$	$A+AB$	$A+AB+1$
1	0	0	1	$\overline{A\oplus B}$	$A+B$	$A+B+1$
1	0	1	0	$B$	$(A+\overline{B})+AB$	$(A+\overline{B})+AB+1$
1	0	1	1	$AB$	$AB-1$	$AB$
1	1	0	0	1	$A+A$	$A+A+1$
1	1	0	1	$A+\overline{B}$	$(A+B)+A$	$(A+B)+A+1$
1	1	1	0	$A+B$	$(A+\overline{B})+A$	$(A+\overline{B})+A+1$
1	1	1	1	$A$	$A-1$	$A$

\*: *Expresadas en complemento a 2*

- + : Operador OR en funciones lógicas y signo más en operaciones aritméticas
- : Signo aritmético menos
- $\overline{\phantom{x}}$  : Barra de inversión lógica

conexiones.

Cada persona escucha únicamente una fracción de lo que le dice la otra, pero debido a características fisiológicas del oído humano, se consigue tener la sensación de que la comunicación es continua cuando se hace rotar el equipo multiplexor-demultiplexor con una cadencia adecuada por encima de los 20 ciclos por segundo

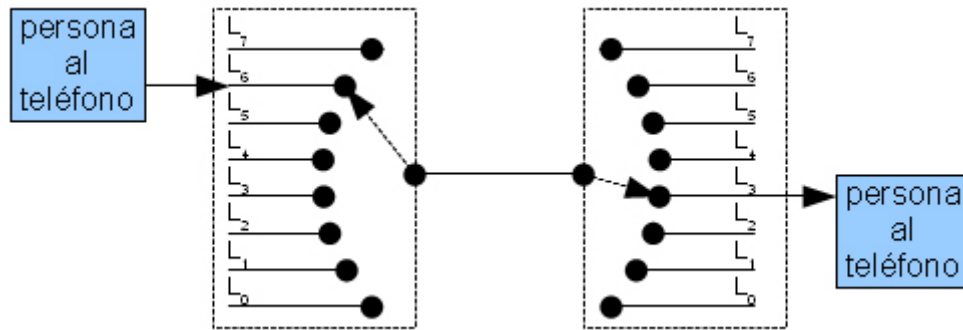
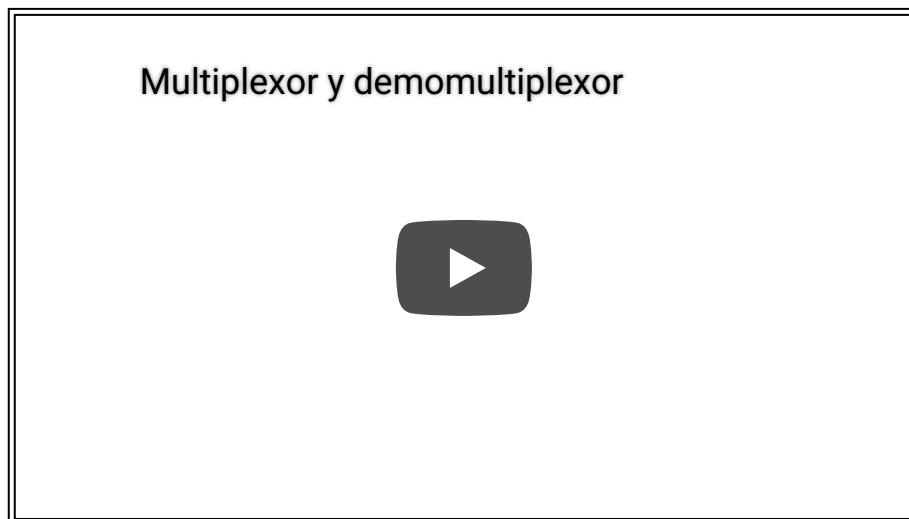


Imagen de elaboración propia

Multiplexor y demultiplexor son circuitos, que operando juntos pueden simplificar la transmisión de datos (como te hemos mostrado en la "curiosidad" anterior, respecto a las líneas telefónicas)

En el siguiente vídeo vamos a ver cómo "unir" el multiplexor y demultiplexor que hemos explicado en anteriores vídeos.



# Fuentes para el profesorado

---

Descargar [CMAP](#).

# Resumen

---

## Importante

Llamamos circuitos combinacionales a los circuitos en los que el estado de sus salidas depende única y exclusivamente de la combinación que toman sus variables de entrada, sin que importen los estados anteriores de las variables ni el tiempo.

Los circuitos combinacionales que se presentan en el tema son de uso común y aparecen o bien aisladamente o formando parte de otros circuitos más complejos de aplicación general.

Se repiten un número de veces tan considerable que se hace aconsejable su fabricación en serie.

## Importante

Los circuitos combinacionales que se ven en el tema son los siguientes:

- **Codificadores:** circuito combinacional integrado que tiene hasta  $2^n$  entradas y  $n$  salidas y la función que desempeña es mostrar en la salida la combinación correspondiente al código binario de la entrada activada.
- **Decodificadores:** Son circuitos combinacionales integrados que disponen de  $n$  entradas y un número de salidas igual o menor a  $2^n$ , actúan de modo que según cual sea la combinación de las variables de entrada se activa una única salida, permaneciendo el resto de ellas desactivada.
- **Convertidores de código:** Son circuitos combinacionales cuya función es cambiar los datos de un código binario a otro, esto es así porque para determinadas operaciones de transmisión y procesamiento de información son más eficaces unos códigos que otros. Se suelen implementar mediante dispositivos lógicos programables.
- **Multiplexores:** Son circuitos combinacionales de hasta  $2^n$  líneas de entrada de datos, una **única** de salida y  $n$  entradas de selección; que habilitan y ponen en contacto uno de los terminales de entrada de datos con el de salida.
- **Demultiplexores:** Son circuitos integrados combinacionales que realizan la función opuesta a un multiplexor. Es decir tiene una única entrada de datos,  $n$  entradas de selección y un número de salidas  $< 2^n$ , de modo que según introduzcamos una combinación u otra por las entradas de selección, así conseguimos comunicar la entrada de datos con la salida seleccionada.
- **Comparadores:** Son circuitos integrados combinacionales con uno o más pares de entradas que tienen como función comparar dos magnitudes binarias para determinar su relación.
- **Sumadores:** Un sumador es un circuito que realiza la suma de dos palabras binarias. Es distinta de la operación OR, con la que no nos debemos confundir. La operación suma de números binarios tiene la misma mecánica que la de números decimales.
- **Restadores:** De modo similar a lo comentado con el sumador, podríamos construir un semi-restador en el que las entradas serán  $M$  = minuendo,  $S$  = sustraendo, y las salidas  $D$  = diferencia,  $P$  = cifra prestada.
- **Detectores/generadores de paridad:** Los generadores de paridad par son

aquellos circuitos que generan un 0 cuando el número de 1 en la entrada es par y un 1 cuando es impar.

# Imprimible

---

Mapa imprimible



# Mapa Conceptual

---

Mapa conceptual (pdf - 306366\_B).

TI\_U4\_T4\_mapa\_conceptual.pdf

1 / 1

# Ejercicios Resueltos

---

A lo largo del tema hemos ido dando algunos ejemplos de cada uno de los circuitos que íbamos estudiando.

Para acabar el tema y con el objetivo, por un lado de afianzar conceptos y por otro de ver que cada circuito no es un "compartimento estanco", más bien todo lo contrario, por lo que se utilizan en combinación con otros para conseguir un objetivo (objetivo que es la función o funciones correspondientes a implementar), te proponemos practiques con los ejercicios resueltos que te proponemos a continuación.

## Aviso Legal

---

El presente texto (en adelante, el "**Aviso Legal**") regula el acceso y el uso de los contenidos desde los que se enlaza. La utilización de estos contenidos atribuye la condición de usuario del mismo (en adelante, el "**Usuario**") e implica la aceptación plena y sin reservas de todas y cada una de las disposiciones incluidas en este Aviso Legal publicado en el momento de acceso al sitio web. Tal y como se explica más adelante, la autoría de estos materiales corresponde a un trabajo de la **Comunidad Autónoma Andaluza, Consejería de Educación y Deporte (en adelante Consejería de Educación y Deporte)**.

Con el fin de mejorar las prestaciones de los contenidos ofrecidos, la Consejería de Educación y Deporte se reserva el derecho, en cualquier momento, de forma unilateral y sin previa notificación al usuario, a modificar, ampliar o suspender temporalmente la presentación, configuración, especificaciones técnicas y servicios del sitio web que da soporte a los contenidos educativos objeto del presente Aviso Legal. En consecuencia, se recomienda al Usuario que lea atentamente el presente Aviso Legal en el momento que acceda al referido sitio web, ya que dicho Aviso puede ser modificado en cualquier momento, de conformidad con lo expuesto anteriormente.

**Régimen de Propiedad Intelectual e Industrial sobre los contenidos del sitio**

---