

# Circuitos Combinacionales

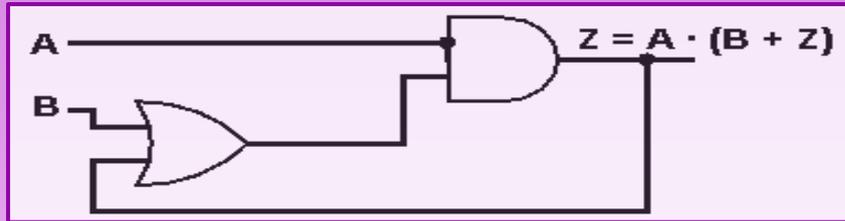
## Apunte N° 5



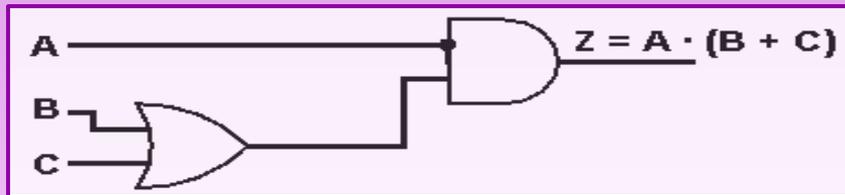
# Qué es un Circuito Combinacional?

Un sistema combinacional es un circuito lógico cuyas salidas están completamente determinadas en cualquier instante por los valores aplicados a sus variables de entrada.

Son circuitos que no tienen bucles de realimentación, es decir una salida no puede usarse como entrada de una etapa anterior.



**NO** es combinacional



**SÍ** es combinacional

# Qué es un Circuito Combinacional?

Un circuito o un sistema lógico combinacional es aquel que:

- ◆ Está formado por funciones lógicas elementales ( AND, OR, NAND, NOR, etc. )
- ◆ Tiene un determinado número de entradas y salidas

**IMPORTANTE:** En cada instante, el valor de la salida (o salidas) depende únicamente de los valores de las entradas. Por lo tanto, en ellos no es necesario tener en cuenta el tiempo.

Ejemplos de sistemas lógicos combinacionales:

- ◆ Codificadores, decodificadores
- ◆ Multiplexores, demultiplexores
- ◆ Comparadores, detectores de paridad...

En los sistemas secuenciales la salida o salidas en un instante de tiempo no solo dependen de los valores de las entradas en ese instante, sino también de los valores que tuvieron en tiempos anteriores. En cierta forma, estos sistemas poseen memoria.

# Qué es un Circuito Combinacional?

Por tanto, un circuito o un sistema lógico combinacional es :

- ◆ La realización ("implementación") de una función lógica tal como se definió en el Tema anterior.
- ◆ Los sistemas o circuitos combinacionales pueden ser representados mediante una tabla de verdad o mediante las expresiones ya vistas anteriormente (formas canónicas, como suma de productos o producto de sumas).
- ◆ Toda función lógica puede implementarse en un circuito o sistema combinacional.

Dos son los aspectos a tratar de los sistemas combinacionales:

- ◆ **Análisis del circuito:** dado el esquema circuital obtener sus ecuaciones para simplificarlas y obtener la especificación literal de su funcionamiento. Debe conducir a una solución única.
- ◆ **Síntesis o diseño del circuito:** Dada una especificación literal, obtener un circuito que la satisfaga (generalmente bajo unos criterios de optimización). Varios circuitos pueden satisfacer la misma tabla de verdad. SOLUCIÓN no ÚNICA.

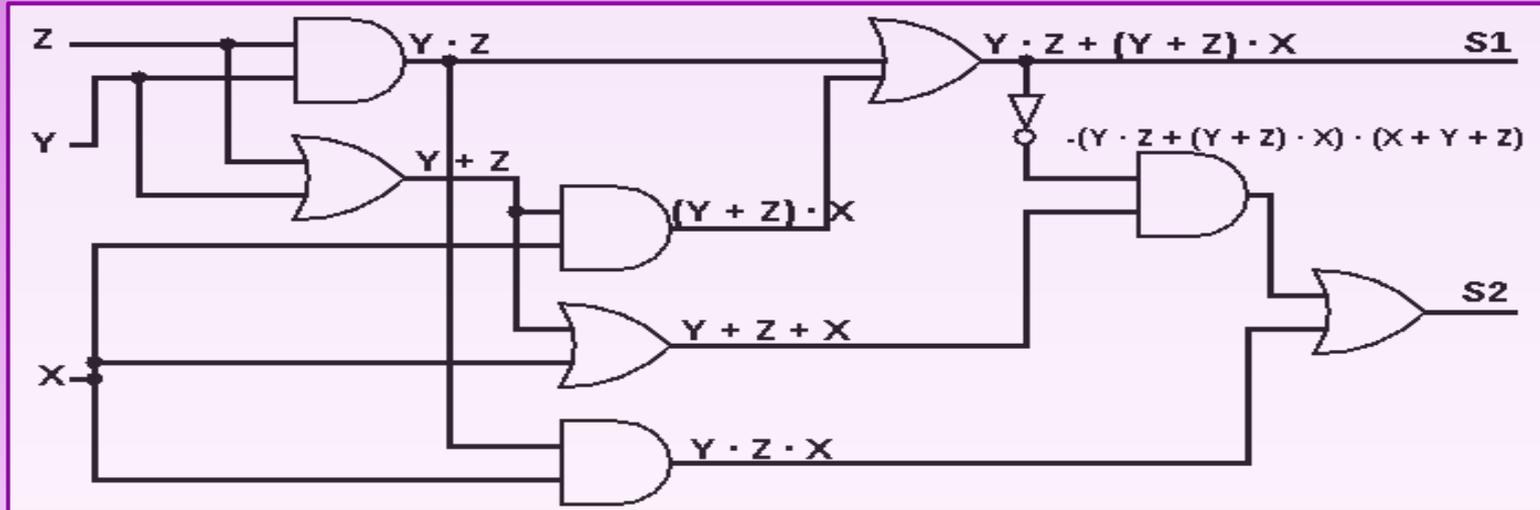
En ambos casos aplicaremos las técnicas de obtención de funciones booleanas y su simplificación, que ya hemos estudiado.

# Análisis de Circuitos Combinacionales

Consiste en determinar la expresión algebraica de la función implementada por el circuito.

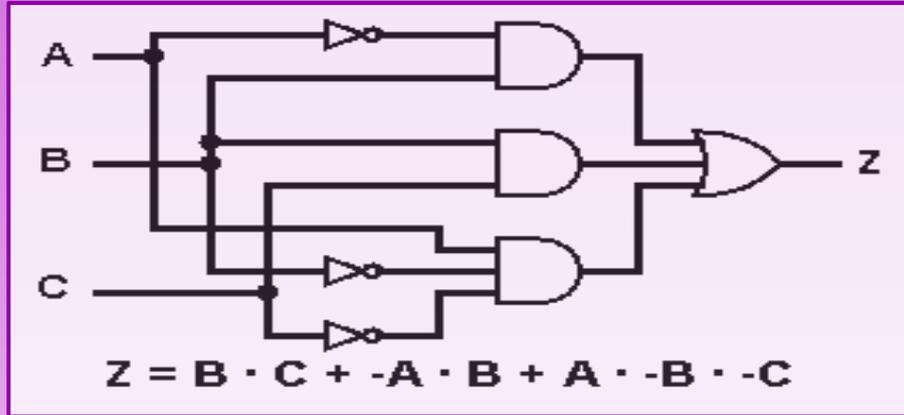
Obtención de las funciones de conmutación que describen el comportamiento del circuito, expresando la salida en función de las entradas.

Se evalúan las expresiones generadas por cada puerta desde sus entradas hasta sus salidas.

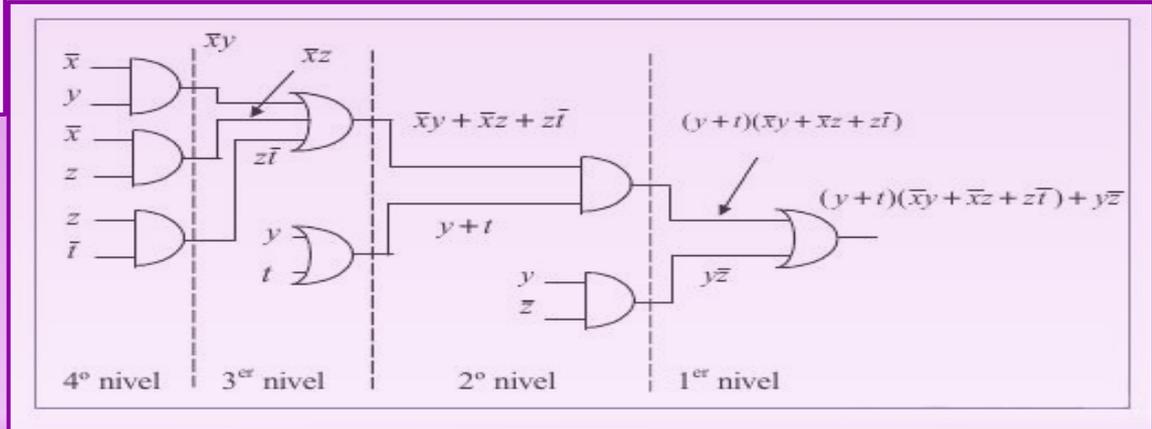


# Análisis de Circuitos Combinacionales

Un circuito combinacional se analiza determinando la salida de los elementos lógicos que lo constituyen (normalmente puertas lógicas), partiendo de las variables de entrada y avanzando en el sentido de la señal hacia la salida.



Un nivel representa cada paso por una compuerta y la de salida es el primer nivel. Las puertas cuya salida alimentan al primer nivel, constituyen el segundo nivel, y así sucesivamente.



# Diseño de Circuitos Combinacionales



*Síntesis*

$F(A, B, C) =$

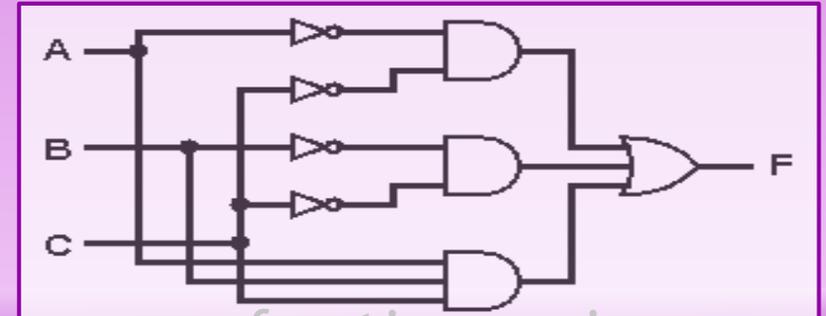
A	B	C	F
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

S  
I  
M  
P  
L  
I  
F  
I  
C  
A  
C  
I  
Ó  
N

\ C	0	1
AB\		
00	1	1
01	1	3
11	6	7
10	1	5

$$F = \sim A \sim C + \sim B \sim C + A B C$$

*Implementación*



# Diseño de Circuitos Combinacionales

La especificación de un sistema combinacional consiste en traducir el enunciado de un problema concreto a variables y funciones booleanas cuya tabla de verdad permita encontrar un circuito lógico que resuelva la situación.

El proceso de diseño de un circuito combinacional consiste en:

- 1) Determinar el número de variables de entrada y de salida necesarias, identificar las variables de entrada, asignarles un nombre y hacer lo mismo con las variables de salida.
- 2) Deducir la tabla de verdad que define las relaciones entre las variables de entrada y de salida.
- 3) Simplificar las funciones representadas en la tabla de verdad.
- 4) Obtener el circuito a partir de las funciones simplificadas.

# Diseño de Circuitos Combinacionales

Conociendo las reglas de análisis combinacional, es muy sencillo el paso de compuertas AND y OR a NAND o NOR.

## ◆ Implementación con compuertas NAND

- a) Realizar el circuito con compuertas AND, OR y NOT.
- b) Para cada compuerta AND, se sustituye una compuerta NAND seguida de un inverso.
- c) Para cada compuerta OR se sustituyen inversores de salida seguidos de una compuerta NAND.

## ◆ Implementación con compuertas NOR

- a) Realizar el circuito con compuertas AND, OR y NOT.
- b) Cambiar todas las compuertas OR por dos compuertas NOR en serie.
- c) Cambiar todas las compuertas AND sustituyendo con inversores en sus entradas y una compuerta NOR.

# Bloques Combinacionales

El progreso de la tecnología electrónica permitió en los 70 ´s la realización de muchas funciones y multifunciones en un solo circuito integrado.

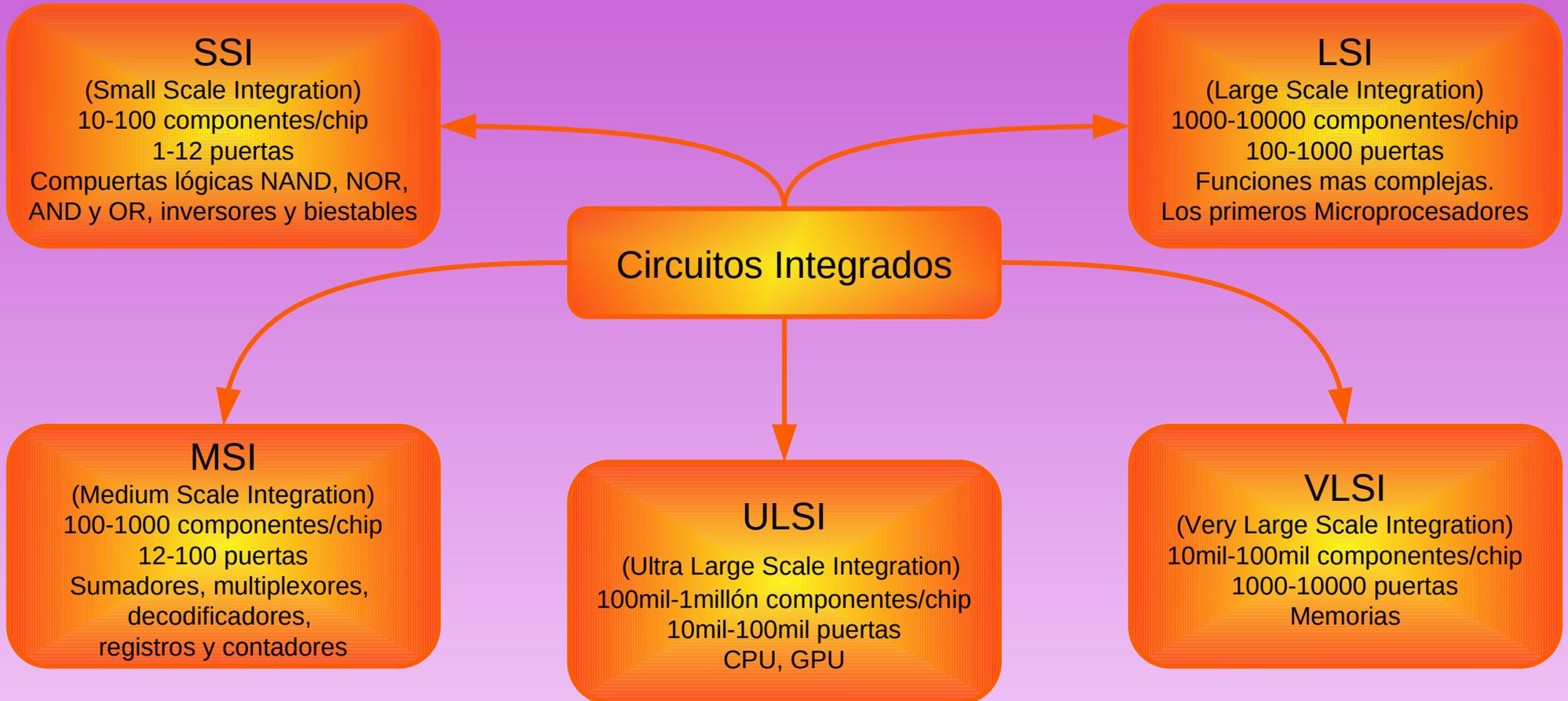
Estos son los que se denominaron circuitos de escala de integración media (MSI).

La combinación de estos circuitos con otros, incluso de carácter secuencial, generaron bloques funcionales complejos que al ser integrados en un chip han dado lugar a los que se conocen como circuitos de gran escala de integración (LSI) y de muy gran escala de integración (VLSI).

Vamos a estudiar algunos bloques funcionales combinacionales que se encuentran como circuitos integrados y cuyo conocimiento resulta imprescindible para diseñar circuitos digitales y comprender el funcionamiento de los más complejos.

- \* Codificadores
- \* Decodificadores
- \* Multiplexores
- \* Demultiplexores
- \* Detectores-generadores de paridad
- \* Sistemas combinacionales programables.

# Bloques Combinacionales



# Bloques Combinacionales

## CODIFICADORES

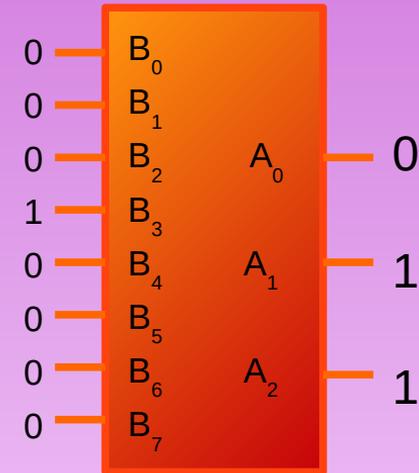
Son sistemas digitales combinacionales con:

- ◆ 2<sup>n</sup> entradas
- ◆ n salidas

**Funcionamiento:** Permite que se le introduzca en una de sus entradas un nivel activo que representa un dígito (decimal u octal) y lo convierte en una salida codificada (como BCD o binario)

Estos circuitos pueden ser diseñados con prioridad o sin ella.

B <sub>0</sub>	B <sub>1</sub>	B <sub>2</sub>	B <sub>3</sub>	B <sub>4</sub>	B <sub>5</sub>	B <sub>6</sub>	B <sub>7</sub>	A <sub>0</sub>	A <sub>1</sub>	A <sub>2</sub>
1	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	0	1	1
0	0	0	0	1	0	0	0	1	0	0
0	0	0	0	0	1	0	0	1	0	1
0	0	0	0	0	0	1	0	1	1	0
0	0	0	0	0	0	0	1	1	1	1



Codificador Octal a Binario

# Bloques Combinacionales

## CODIFICADORES

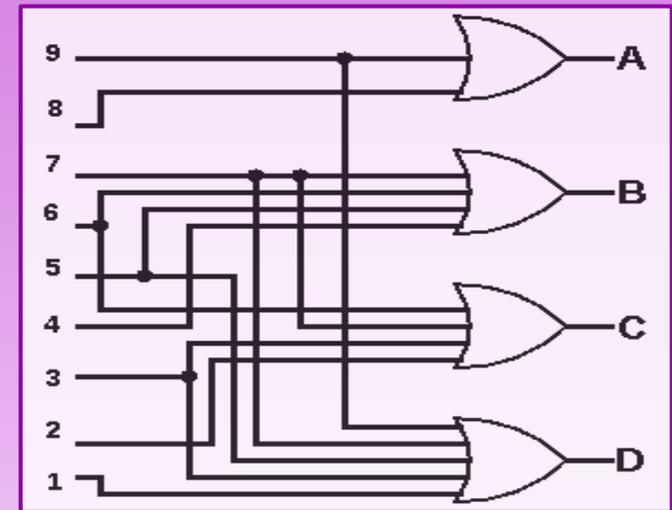
Codificador decimal-BCD (sin prioridad)

Para realizar esta codificación (de 0 a 9 en decimal) necesitamos 4 dígitos binarios.

Tiene 9 entradas y 4 salidas.

No se necesita una entrada para el dígito 0, ya que las salidas BCD están todas a nivel bajo cuando no hay entradas a nivel alto.

$B_1$	$B_2$	$B_3$	$B_4$	$B_5$	$B_6$	$B_7$	$B_8$	$B_9$	A	B	C	D
1	0	0	0	0	0	0	0	0	0	0	0	1
0	1	0	0	0	0	0	0	0	0	0	1	0
0	0	1	0	0	0	0	0	0	0	0	1	1
0	0	0	1	0	0	0	0	0	0	1	0	0
0	0	0	0	1	0	0	0	0	0	1	0	1
0	0	0	0	0	1	0	0	0	0	1	1	0
0	0	0	0	0	0	1	0	0	0	1	1	1
0	0	0	0	0	0	0	1	0	1	0	0	0
0	0	0	0	0	0	0	0	1	1	0	0	1



# Bloques Combinacionales

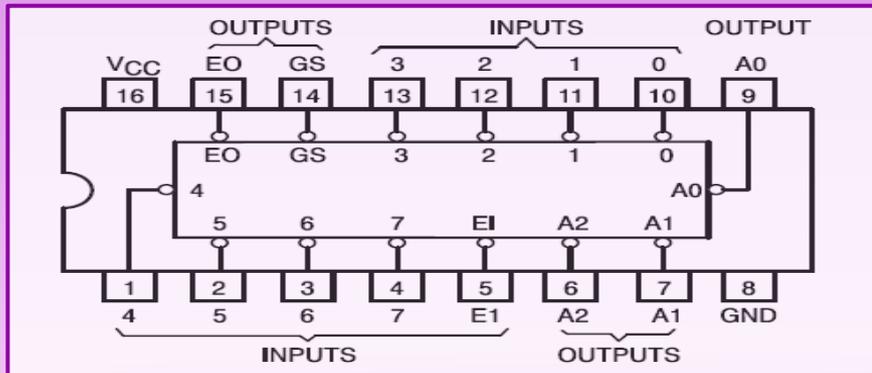
## Codificadores con prioridad

En los codificadores sin prioridad si se activan simultáneamente varias líneas de entrada se genera un código erróneo en la salida, de acuerdo al número de entradas excitadas con el respectivo valor.

La solución de este inconveniente se logra empleando codificadores de prioridad de modo que respondan a una sola señal de entrada activa.

Funcionan codificando la entrada activa de mayor valor decimal sin tener en cuenta las demás.

Ejemplo: Codificador 74148 (de octal a binario) activo por bajos.



Inputs									Outputs				
EI	7	6	5	4	3	2	1	0	A2	A1	A0	GS	EO
1	X	X	X	X	X	X	X	X	1	1	1	1	1
0	1	1	1	1	1	1	1	1	1	1	1	1	0
0	0	X	X	X	X	X	X	X	0	0	0	0	1
0	1	0	X	X	X	X	X	X	0	0	1	0	1
0	1	1	0	X	X	X	X	X	0	1	0	0	1
0	1	1	1	0	X	X	X	X	0	1	1	0	1
0	1	1	1	1	0	X	X	X	1	0	0	0	1
0	1	1	1	1	1	0	X	X	1	0	1	0	1
0	1	1	1	1	1	1	0	X	1	1	0	0	1
0	1	1	1	1	1	1	1	0	1	1	1	0	1

# Bloques Combinacionales

## DECODIFICADORES

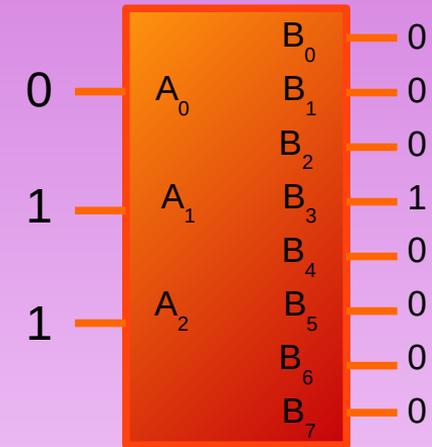
Un decodificador es un circuito combinacional:

- ◆ n entradas y
- ◆  $2^n$  salidas

Funcionamiento: Si la configuración binaria presente en las entradas forma el número binario i entonces se activa la salida i-ésima.

Según esto el funcionamiento de un decodificador es el opuesto al de un codificador.

$A_0$	$A_1$	$A_2$	$B_0$	$B_1$	$B_2$	$B_3$	$B_4$	$B_5$	$B_6$	$B_7$
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1



Decodificador Binario a Octal

# Bloques Combinacionales

## DECODIFICADORES

Decodificador 2:4 (2 líneas de entrada y 4 líneas de salida).

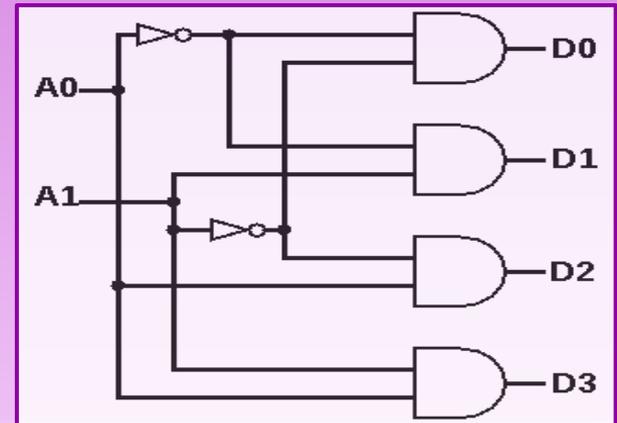
Las entradas del decodificador son  $A_0$  y  $A_1$  y representan un entero de 0 a 3 en código decimal.

Según el valor binario presente en las 2 entradas se activa una de las 4 salidas al valor lógico 1.

Por ejemplo, con el valor  $A_1 A_0 = 10$  se activará la salida  $D_2$ .

Puede comprobarse como este circuito genera los productos canónicos (minterminos) de un conjunto de variables binarias aplicadas a sus entradas.

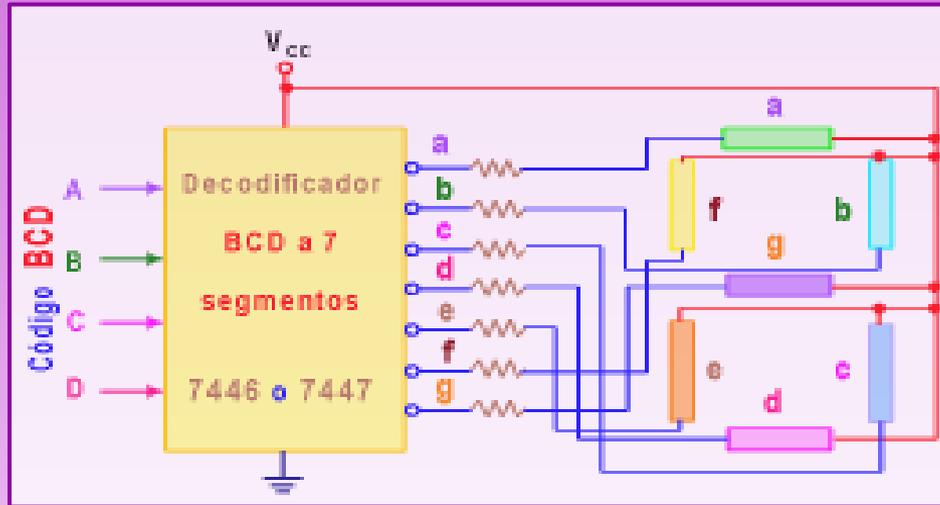
$A_0$	$A_1$	$D_3$	$D_2$	$D_1$	$D_0$
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	0	0	0



# Bloques Combinacionales

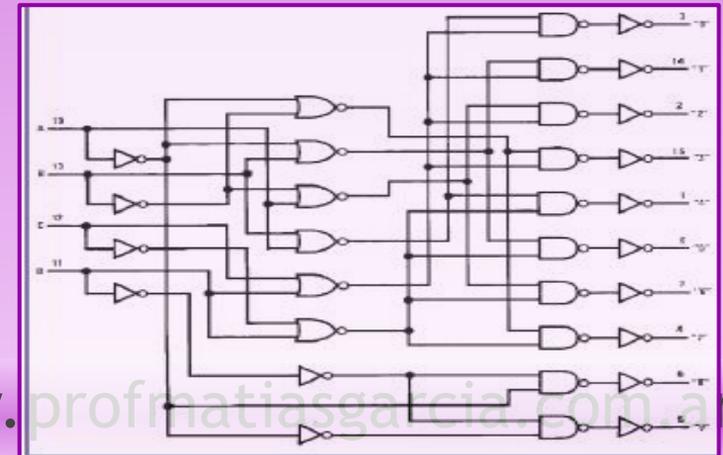
## DECODIFICADORES

Este decodificador permite visualizar números en BCD mediante un display de 7 segmentos.



A la derecha tenemos un decodificador BCD a Decimal

A	B	C	D	0	1	2	3	4	5	6	7	8	9
0	0	0	0	1	0	0	0	0	0	0	0	0	0
0	0	0	1	0	1	0	0	0	0	0	0	0	0
0	0	1	0	0	0	1	0	0	0	0	0	0	0
0	0	1	1	0	0	0	1	0	0	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	0	0	1	0	0	0	0
0	1	1	0	0	0	0	0	0	0	1	0	0	0
0	1	1	1	0	0	0	0	0	0	0	1	0	0
1	0	0	0	0	0	0	0	0	0	0	0	1	0
1	0	0	1	0	0	0	0	0	0	0	0	0	1



# Bloques Combinacionales

## MULTIPLEXORES

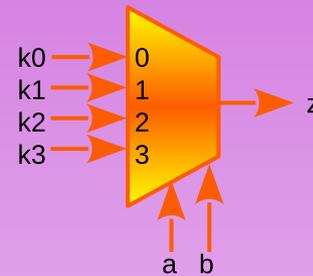
Un multiplexor es un sistema digital que consta de varias entradas y una salida, y mediante un mecanismo de selección, una determinada entrada se transfiere a la salida.

Una definición más formal de multiplexor sería la de un circuito combinatorial con:

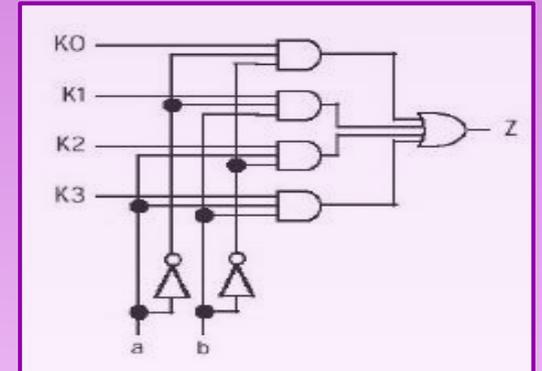
- ◆  $2^n$  entradas de datos ( $k_0, k_1, k_2, \dots$ )
- ◆  $n$  entradas de selección o control ( $a, b, \dots$ )

**Funcionamiento:** permite elegir cuál es el canal de entrada cuya información aparece en el de salida.

La selección del canal de entrada se realiza con el número binario puesto en la entrada de selección.



a	b	Z
0	0	k0
0	1	k1
1	0	k2
1	1	k3



# Bloques Combinacionales

## MULTIPLEXORES

Pueden ser utilizados, para la síntesis de funciones combinacionales en donde las entradas de selección se toman como entradas de las variables.

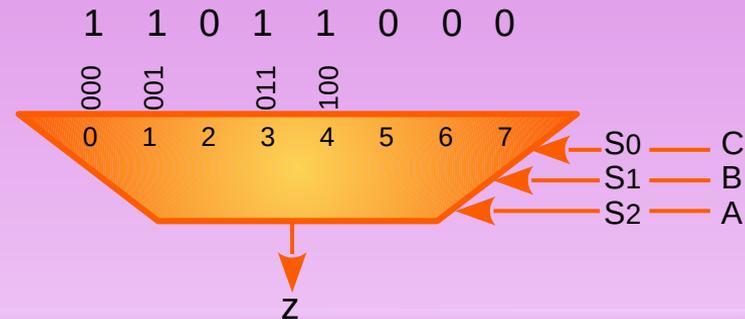
Para una combinación dada de las variables de entrada se establece una "conexión" entre la entrada correspondiente y la salida.

De modo que basta con poner el terminal correspondiente a "0" o a "1" según el valor que ha de tomar la función para la combinación seleccionada de las entradas (según la tabla de verdad).

De modo que para generar una función de tres variables necesitamos un MUX de 8 a 1.

Ejemplo:  $f = (\bar{a} \cdot \bar{b}) + (\bar{a} \cdot b \cdot c) + (a \cdot \bar{b} \cdot \bar{c})$

$f = (\bar{a} \cdot \bar{b} \cdot \bar{c}) + (\bar{a} \cdot \bar{b} \cdot c) + (a \cdot \bar{b} \cdot \bar{c}) + (a \cdot \bar{b} \cdot c)$



# Bloques Combinacionales

## DEMÚLTIPLEXORES

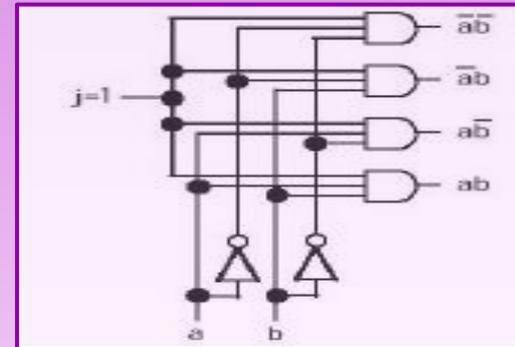
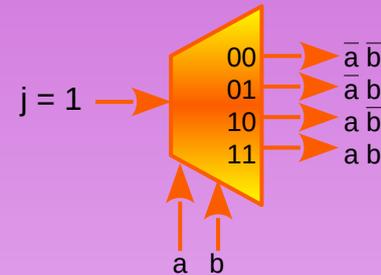
El funcionamiento es el contrario al del multiplexor, encauzando los datos desde una fuente común de entrada hacia uno de  $2^n$  destinos de salida.

Por tanto, un demultiplexor es un circuito combinatorial con:

- ◆ una entrada  $J$
- ◆  $n$  entradas de selección
- ◆  $2^n$  salidas.

La información  $J$  se puede hacer aparecer en cualquiera de las salidas aplicando a las entradas de selección la combinación binaria adecuada.

Es importante comentar que los demultiplexores pueden trabajar como decodificadores.



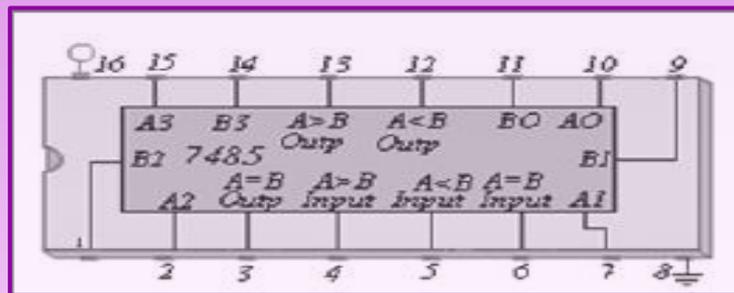
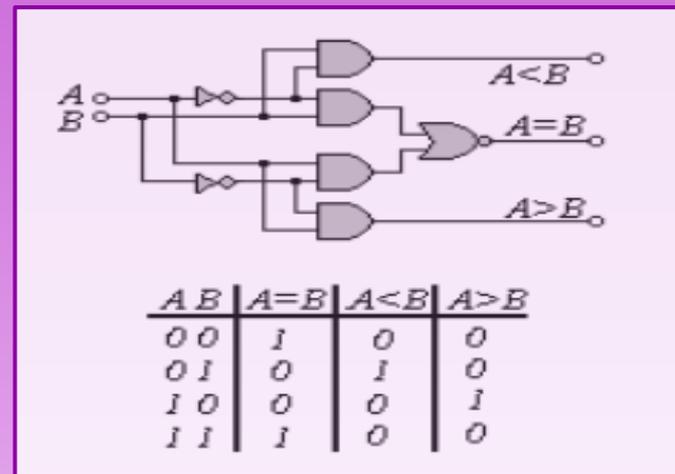
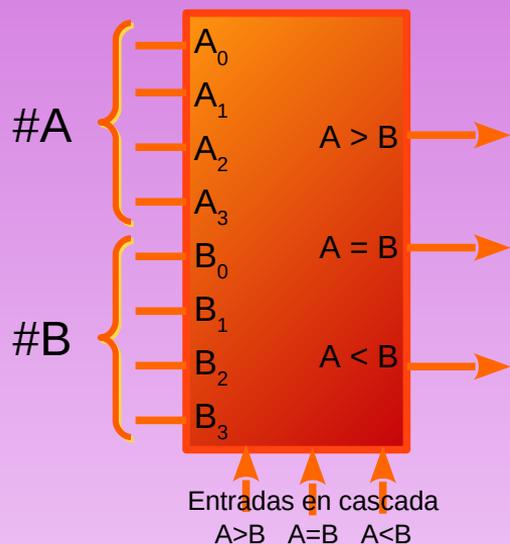
# Bloques Combinacionales

## COMPARADORES BINARIOS

Es un circuito combinatorial que compara números binarios de una cierta cantidad de bits activando a su salida:

**G** (great) si es mayor

**E** (equal) si son iguales.



# Bloques Combinacionales

## COMPARADORES DE PARIDAD

Cuando se transfieren datos digitales de un punto a otro dentro de un sistema digital o cuando se transmiten códigos desde un sistema a otro, se pueden producir errores. Estos errores se manifiestan mediante cambios indeseados en los bits que conforman la información codificada; es decir, un 1 puede cambiar a 0 o un 0 a 1, debido a un mal funcionamiento de los componentes o al ruido eléctrico. En la mayoría de los sistemas digitales, la probabilidad de que haya un bit erróneo es muy pequeña, y la de que haya más de uno es todavía menor. En cualquier caso, cuando no se detecta un error, pueden originarse serios problemas en un sistema digital.

Para poder comprobar o generar la paridad adecuada dentro de un determinado código, se puede aplicar un principio muy sencillo:

**La suma (descartando los acarrees) de un número par de 1s siempre es 0 y la suma de un número impar de 1s siempre es 1.**

Por tanto, para determinar si un cierto código tiene paridad par o paridad impar, se suman todos los bits de ese código. La suma de dos bits se puede generar mediante una puerta OR-exclusiva.

# Bibliografía y Licencia

- ◆ Acha, Santiago, Castro, Manuel, Rioseras, Miguel, "Electrónica Digital Introducción a la Lógica Digital" 2da Ed. (Ra-Ma 2010)
- ◆ Floyd, Thomas, "Fundamentos de sistemas digitales" 9na Ed. (Pearson 2006)
- ◆ Gonzalez Gómez, Juan, "Circuitos y Sistemas Digitales" (Madrid 2002)
- ◆ Este documento se encuentra bajo Licencia Creative Commons Attribution - NonCommercial - ShareAlike 4.0 International (CC BY-NC-SA 4.0), por la cual se permite su exhibición, distribución, copia y posibilita hacer obras derivadas a partir de la misma, siempre y cuando se cite la autoría del Prof. Matías E. García y sólo podrá distribuir la obra derivada resultante bajo una licencia idéntica a ésta.
- ◆ Autor:

***Matías E. García***

Prof. & Tec. en Informática Aplicada  
[www.profmatiasgarcia.com.ar](http://www.profmatiasgarcia.com.ar)  
[info@profmatiasgarcia.com.ar](mailto:info@profmatiasgarcia.com.ar)

 **creative commons**



[www.profmatiasgarcia.com.ar](http://www.profmatiasgarcia.com.ar)