



Universidad  
Politécnica  
de Cartagena

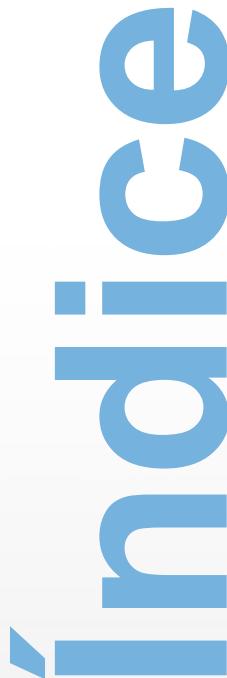
# Tema 1. Circuitos digitales Electrónica Industrial

Andrés Iborra García  
Departamento de Tecnología Electrónica  
Septiembre 2012

**CLOUD**  
INCUBATOR HUB

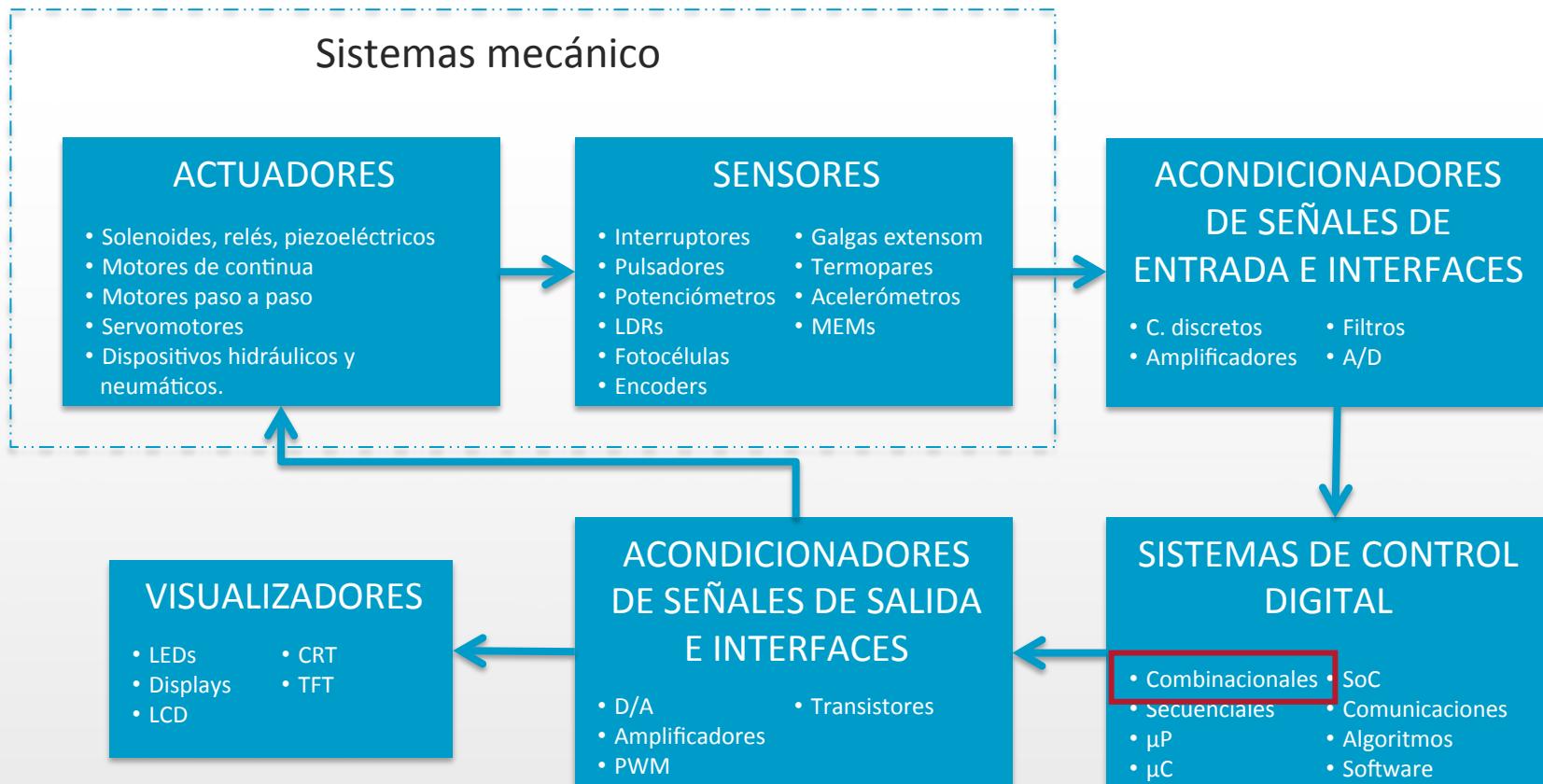


- 1. Circuitos digitales combinacionales.**
  - 1.1. Señales analógicas y digitales.
  - 1.2. Señales digitales.
  - 1.3. Dispositivos lógicos combinacionales.
  - 1.4. Circuitos integrados.
  - 1.5. Álgebra de Boole.
  - 1.6. Diseño de circuitos combinacionales.
- 2. Circuitos digitales secuenciales**
  - 2.1 Flip-Flops y Latches.
  - 2.2 Flip-flops activados por flanco.
  - 2.3 Aplicaciones de los Flip-Flops.
- 3. Tecnologías TTL y CMOS**
  - 3.1 Familias Lógicas.
  - 3.2 Parámetros característicos.
  - 3.3 Hojas de datos de los fabricantes de CI.
  - 3.4 Conectividad entre CI de diferentes familias.



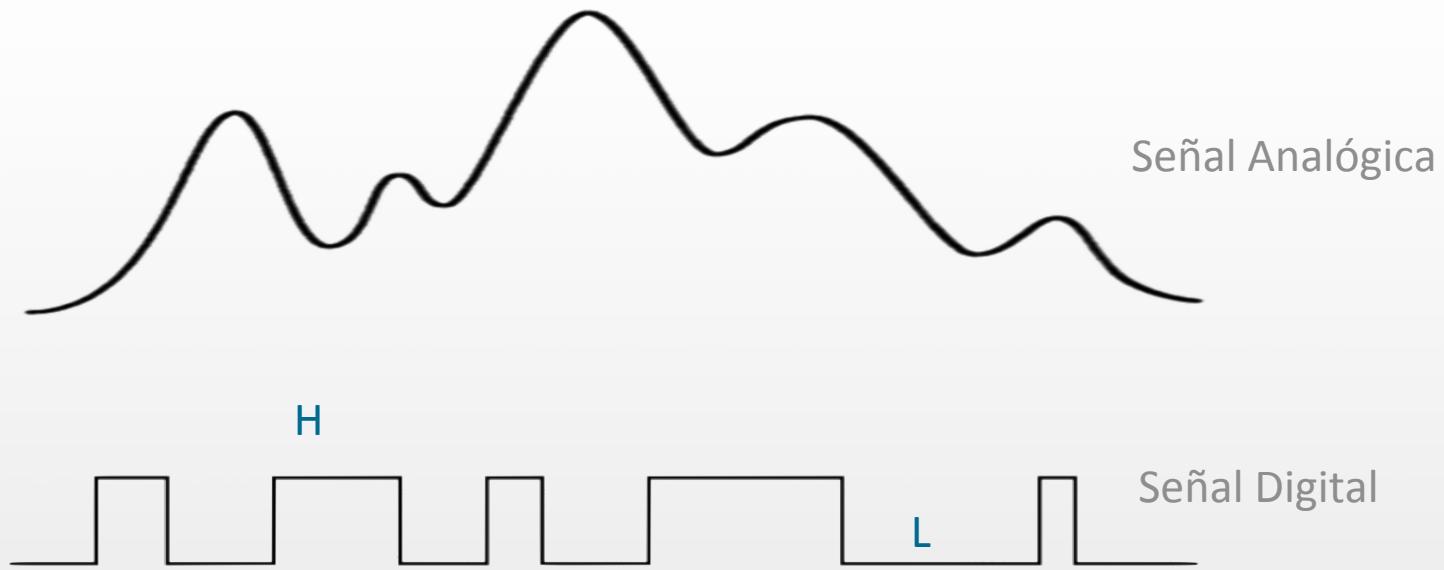
- 4. Circuitos integrados MSI.**
  - 4.1 Niveles de integración de los CI digitales.
  - 4.2 Operaciones aritméticas.
  - 4.3 Comparadores.
  - 4.4 Codificadores.
  - 4.5 Decodificadores.
  - 4.6 Multiplexores.
  - 4.7 Demultiplexores.
- 5. Diseño de Sistemas Digitales con Circuitos Integrados LSI y MSI**
  - 5.1 Habilitación de puertos de E/S.
  - 5.2 Sistema de recuento de votos.
  - 5.3 Display digital.
  - 5.4 Teclado digital.
  - 5.5 Display multiplexado.

# Circuitos Digitales Combinacionales



# Señales analógicas y digitales

- En contraste con una señal analógica, en una señal digital solo existen niveles o estados específicos y cambia su nivel en pasos discretos.



En electrónica digital las señales serán de tensión.

$V_H$ , H: Tensión o nivel alto;  $V_L$ , L: Tensión o nivel bajo

# Señales digitales. Niveles de tensión

Voltajes típicos

$V_{Hmax}$

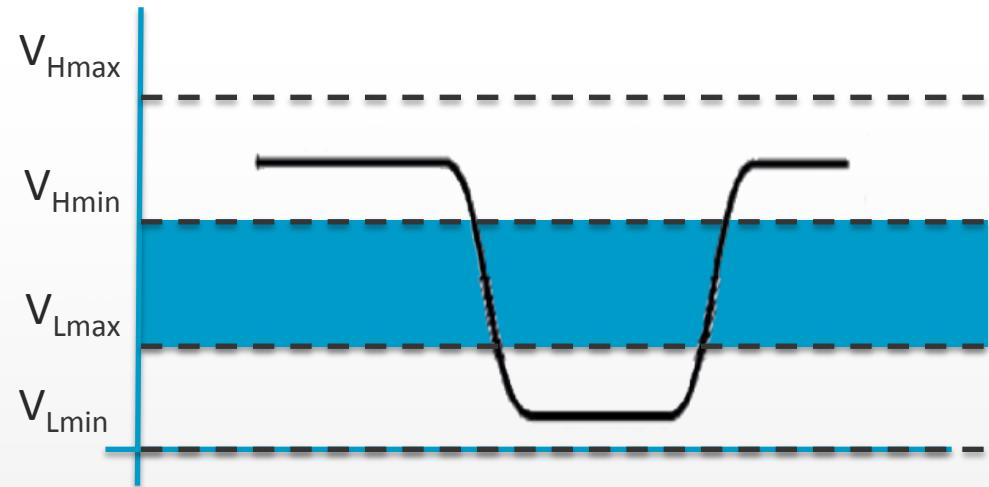
$V_{Hmin}$

Zona incertidumbre

$V_{Lmax}$

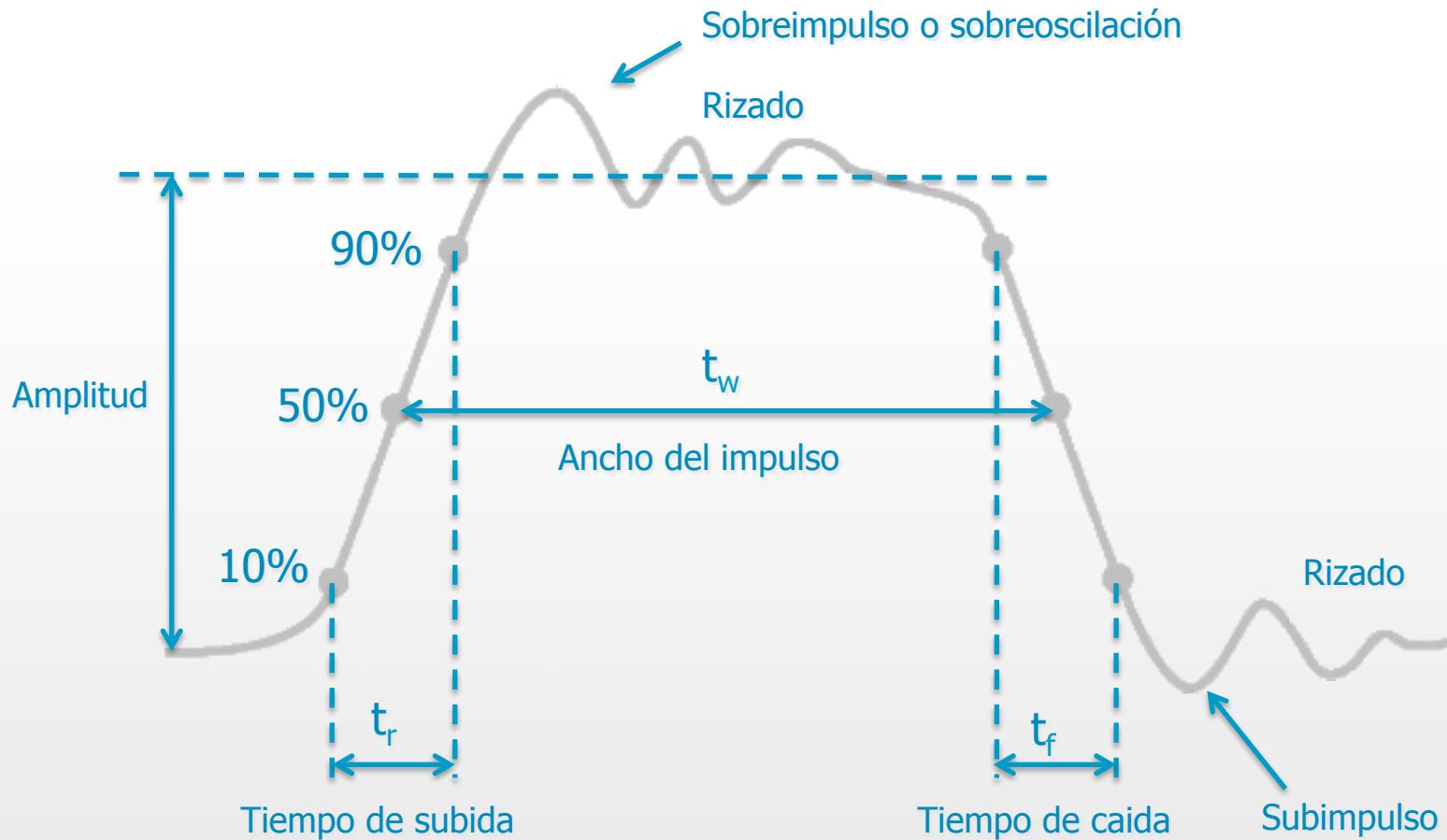
$V_{Lmin}$

	TTL	CMOS*
$V_{Hmax}$	5V	5V
$V_{Hmin}$	2V	3,5V
Zona incertidumbre		
$V_{Lmax}$	0,8V	1V
$V_{Lmin}$	0V	0V



\* Aunque los dispositivos CMOS se pueden alimentar entre 3 y 18V, en este curso solo consideraremos la tensión de 5V para alimentar este tipo de dispositivos.

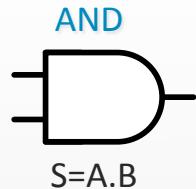
# Señales digitales. Aspecto real.



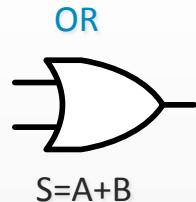
# Dispositivos lógicos combinacionales

Los **dispositivos lógicos combinacionales** son dispositivos digitales que convierten entradas binarias en salidas binarias en base en las reglas del Álgebra de Boole. Los dispositivos más sencillos son las **Puertas Lógicas**.

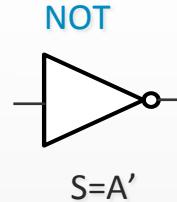
## Puertas básicas



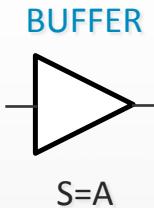
A	B	S
0	0	0
0	1	0
1	0	0
1	1	1



A	B	S
0	0	0
0	1	1
1	0	1
1	1	1



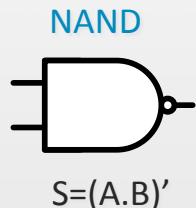
A	S
0	1
1	0



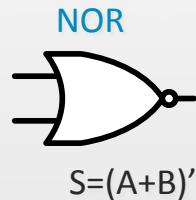
A	S
0	0
1	1

Mejora el fan-out

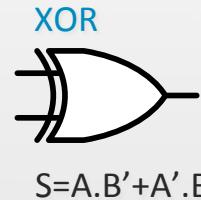
## Puertas compuestas



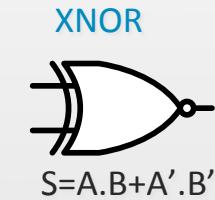
A	B	S
0	0	1
0	1	1
1	0	1
1	1	0



A	B	S
0	0	1
0	1	0
1	0	0
1	1	0

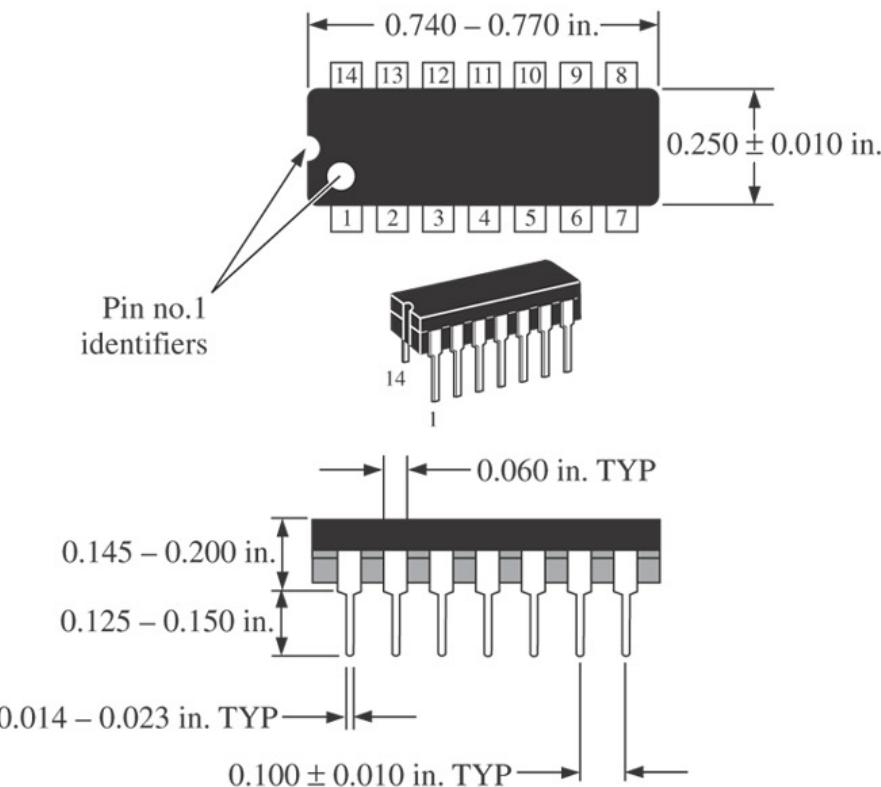


A	B	S
0	0	0
0	1	1
1	0	1
1	1	0

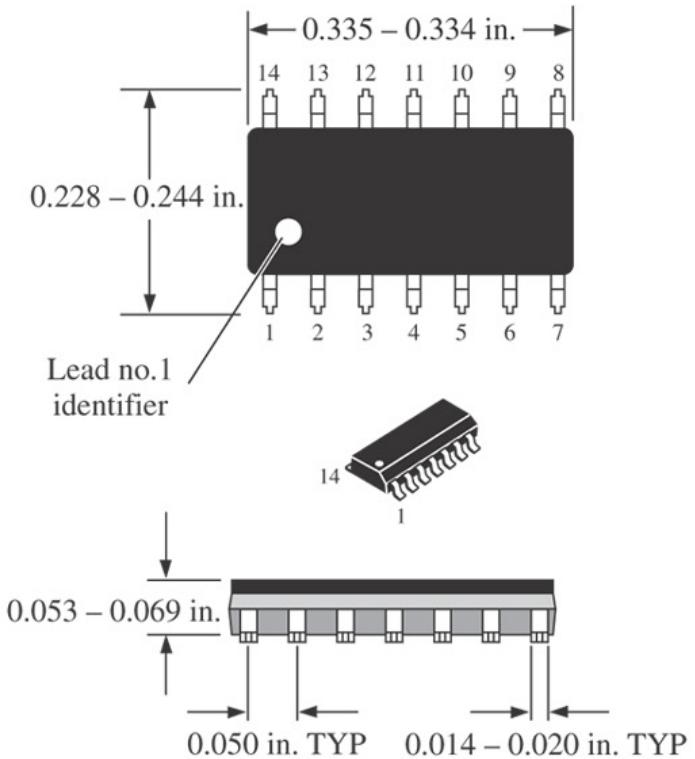


A	B	S
0	0	1
0	1	0
1	0	0
1	1	1

# Circuitos Integrados. Empaquetado.



(a) 14-pin dual in-line package (DIP) for feedthrough mounting



(b) 14-pin small outline package (SOIC) for surface mounting

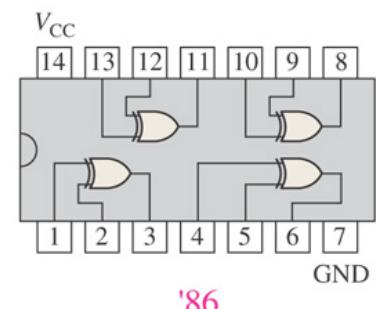
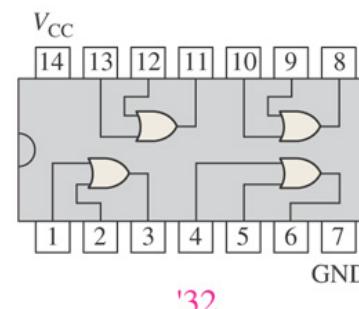
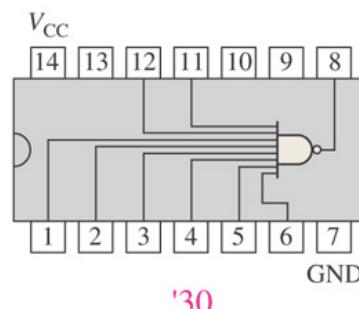
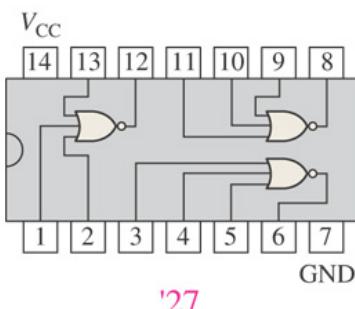
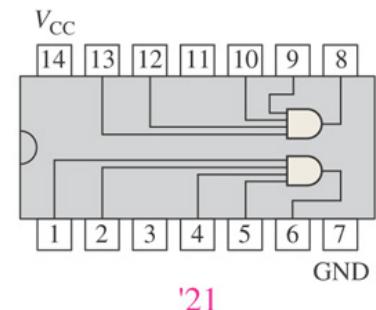
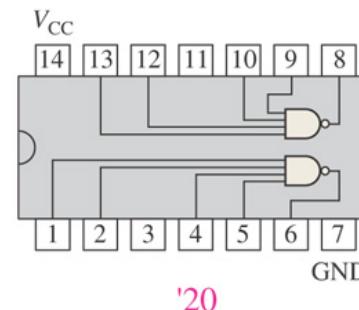
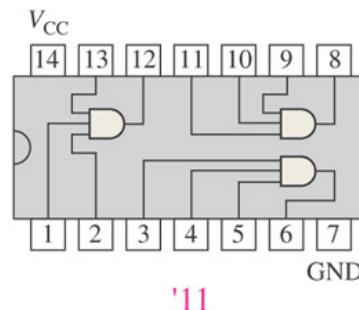
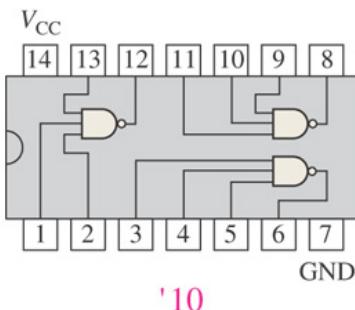
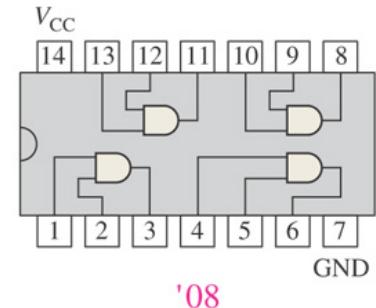
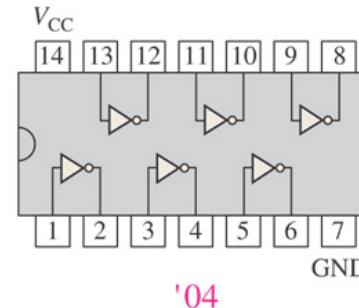
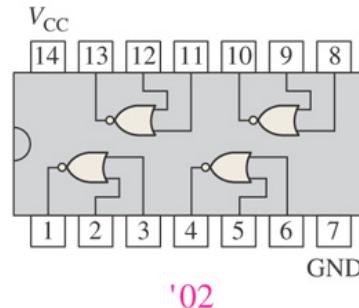
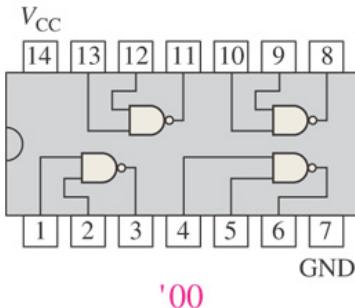
# Circuitos Integrados. Familias lógicas.

- Existen una gran variedad de familias de circuitos integrados dependiendo de sus características tecnológicas de fabricación. Las más utilizadas son:
- **TTL.** Lógica Transistor-Transistor.
- **CMOS.** Semiconductores complementarios de óxido metálico.
- **ECL.** Lógica de emisor acoplado.

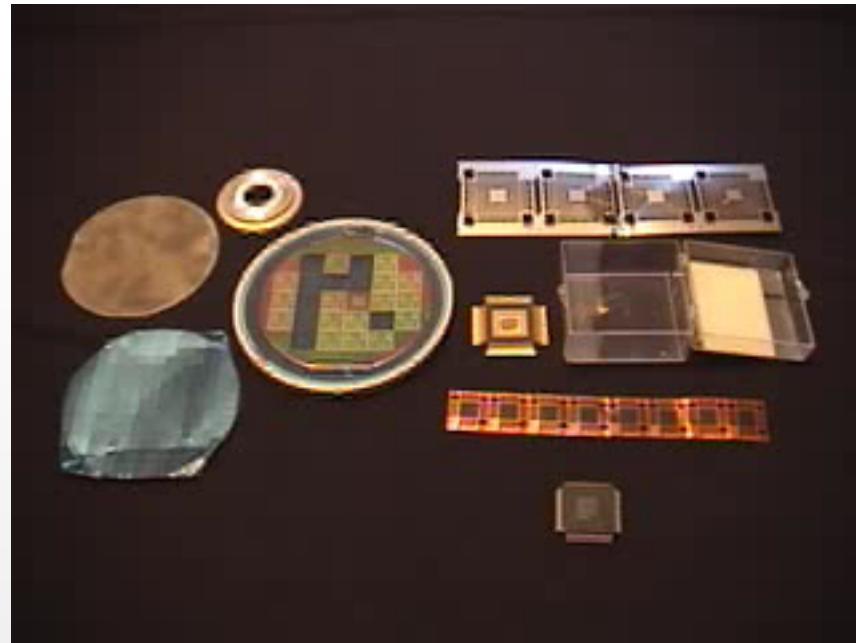
¡Atención! Los niveles lógicos varían dependiendo del tipo de familia.

Un diseñador debe tener cuidado cuando mezcle diferentes tipos de circuitos integrados digitales porque pueden tener características incompatibles (niveles de tensión de entrada-salida, fan-out, corrientes, etc.)

# Circuitos Integrados. Ejemplos



# Circuitos Integrados. Fabricación.



# Álgebra de Boole.

El álgebra de Boole se utiliza para el análisis y síntesis de circuitos digitales.

## Estados posibles.

Lógica positiva:

- 0: Nivel de tensión “bajo” (LOW, L)
- 1: Nivel de tensión “alto” (HIGH, H)

Lógica negativa:

- 1: Nivel de tensión “bajo”
- 0: Nivel de tensión “alto”

## Señales eléctricas y Variables Booleanas.

- Las señales en los sistemas digitales se representan mediante caracteres alfabéticos “A”, “B”, “X” o una combinación de letras y números “A1”, “A2”, “A3”... y se corresponden con variables booleanas.
- Pueden tomar dos valores (0 ó 1).
- Se corresponden con señales de entrada, de salida o intermedias.

# Álgebra de Boole. Axiomas.

Se postulan 5 pares de axiomas:

## Abstracción digital.

$$(A1) \ X=0 \text{ si } X \neq 1 \quad (A1') \ X=1 \text{ si } X \neq 0$$

OJO!!! Se cumple el principio de dualidad

## Función inversora.

$$(A2) \ Si \ X=0 \text{ entonces } X'=1 \quad (A2') \ Si \ X=1 \text{ entonces } X'=0$$

OJO!!! Se cumple el principio de dualidad

## Definición formal de las operaciones básicas.

$$(A3) \ 0.0 = 0 \quad (A3') \ 1+1 = 1$$

$$(A4) \ 1.1 = 1 \quad (A4') \ 0+0 = 0$$

$$(A5) \ 0.1 = 1.0 = 0 \quad (A5') \ 1+0 = 0+1 = 1$$

# Álgebra de Boole. Teoremas (I).

## Teoremas fundamentales.

OR

$$A+0 = A$$

$$A+1 = 1$$

$$A+A = A$$

$$A+A' = 1$$

AND

$$A \cdot 0 = 0$$

$$A \cdot 1 = A$$

$$A \cdot A = A$$

$$A \cdot A' = 0$$

NOT

$$A'' = A$$

## Propiedad Conmutativa.

$$A+B = B+A$$

$$A \cdot B = B \cdot A$$

## Propiedad Asociativa.

$$(A+B)+C = A+(B+C)$$

$$(A \cdot B) \cdot C = A \cdot (B \cdot C)$$

## Propiedad Distributiva.

$$A \cdot (B+C) = (A \cdot B) + (A \cdot C)$$

$$A + (B \cdot C) = (A + B) \cdot (A + C)$$

Los teoremas se pueden demostrar por **Inducción Perfecta** o **Teoría de Conjuntos**

# Álgebra de Boole. Teoremas (II).

## Otros Teoremas.

$$A + (A \cdot B) = A$$

$$A + (A' \cdot B) = A + B$$

$$(A + B) \cdot (A + C) = A + (B \cdot C)$$

$$(A \cdot B) + (B \cdot C) + (B' \cdot C) = (A \cdot B) + C$$

$$A \cdot (A + B) = A$$

$$(A + B) \cdot (A + B') = A$$

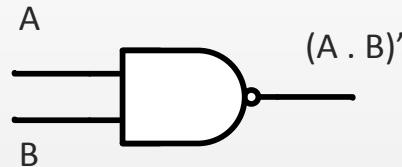
$$A + B + (A \cdot B') = A + B$$

$$(A \cdot B) + (A \cdot C) + (B' \cdot C) = (A \cdot B) + (B' \cdot C)$$

## Teoremas de De Morgan.

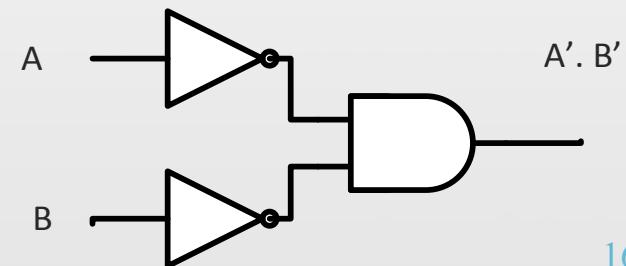
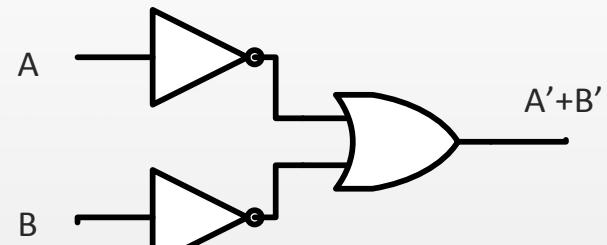
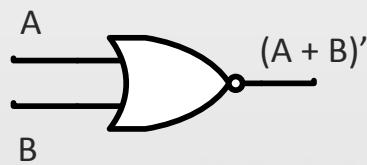
### 1er Teorema

$$(A \cdot B)' = A' + B'$$



### 2º Teorema

$$(A + B)' = A' \cdot B'$$

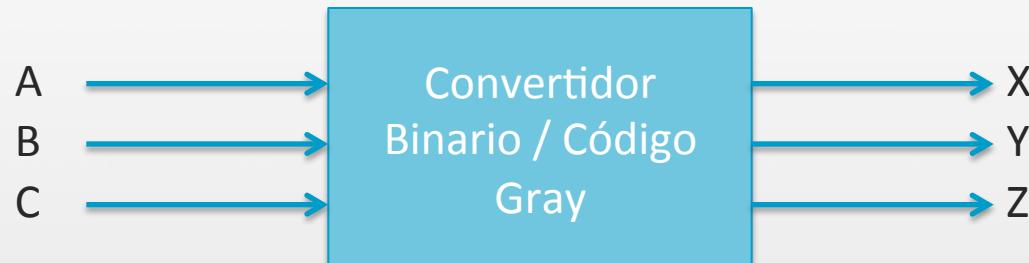


# Diseño de circuitos combinacionales.

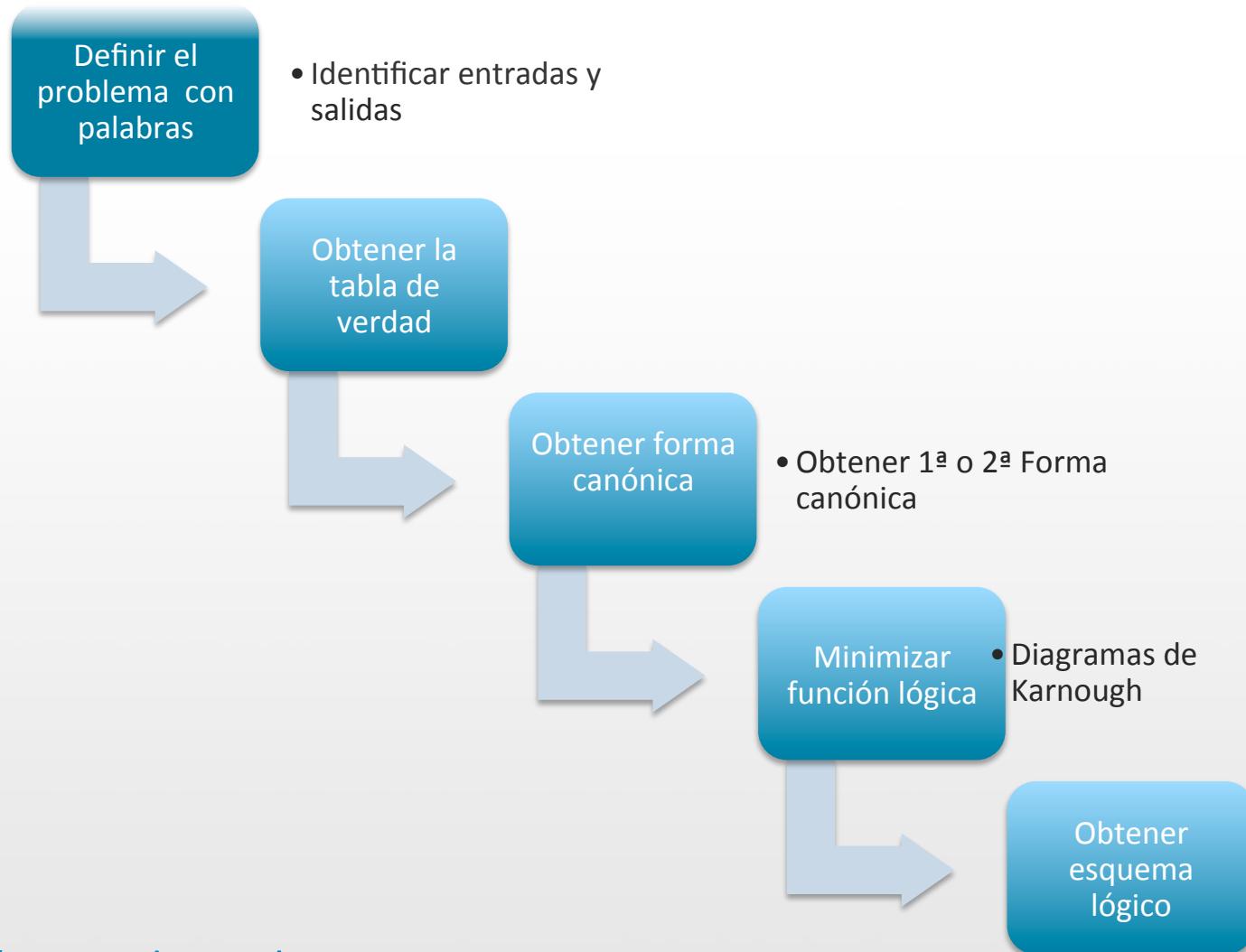
- Los circuitos combinacionales se caracterizan porque las salidas solo dependen del estado actual de las entradas. Son circuitos que no almacenan ningún tipo de información y se pueden construir utilizando exclusivamente puertas lógicas.

## Ejemplo

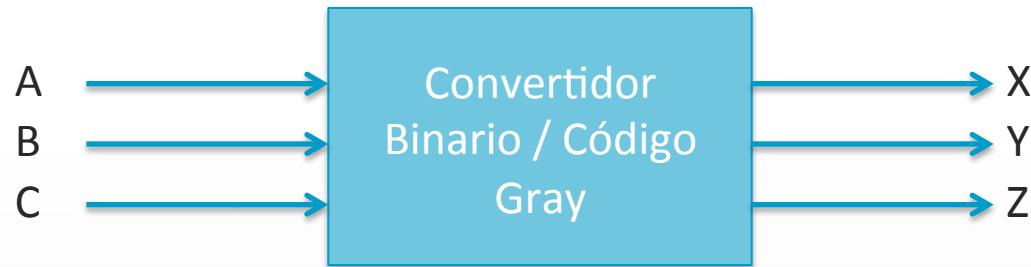
Circuito para convertir números binarios de 3 bits a código Gray



# Diseño. Flujo de trabajo



# Diseño. Tabla de verdad

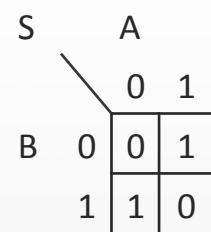


A	B	C	X	Y	Z
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	1
0	1	1	0	1	0
1	0	0	1	1	0
1	0	1	1	1	1
1	1	0	1	0	1
1	1	1	1	0	0

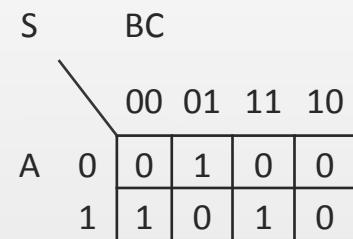
# Diseño. Mapas de Karnaugh (I)

El **método de Karnaugh** es un procedimiento gráfico de **minimización de circuitos lógicos combinacionales**. Se basa en representar la información de la tabla de verdad sobre los mapas de Karnaugh y luego aplicar un conjunto de reglas de agrupamiento.

A	B	S
0	0	0
0	1	1
1	0	1
1	1	0



A	B	C	S
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1



A	B	C	D	S
0	0	0	0	0
0	0	0	1	1
0	0	1	0	0
0	0	1	1	0
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	1
1	0	1	1	0
1	1	0	0	0
1	1	0	1	1
1	1	1	0	0
1	1	1	1	1

S	CD	00	01	11	10
AB	00	0	1	0	0
01	1	0	1	0	
11	0	1	1	0	
10	0	0	0	0	

# Diseño. Mapas de Karnaugh (II)

Fundamento del Método.

A	B	C	S
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

A	B	C	S
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

A	B	C	S
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

S	BC				
	00	01	11	10	
A	0	0	1	1	0
	1	1	1	1	0

$$S = A'B'C + A'BC = A'C(B' + B) = A'C$$

S	BC				
	00	01	11	10	
A	0	0	1	1	0
	1	1	1	1	0

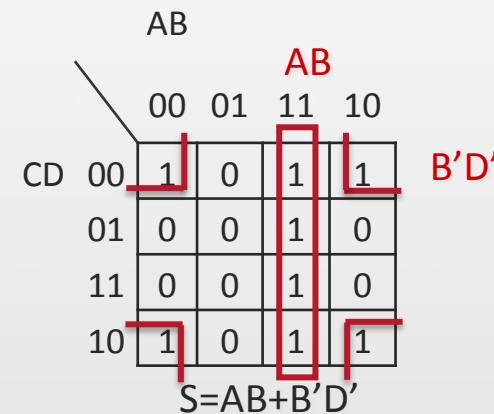
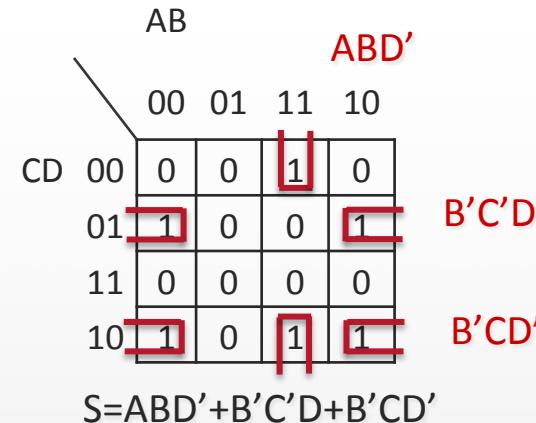
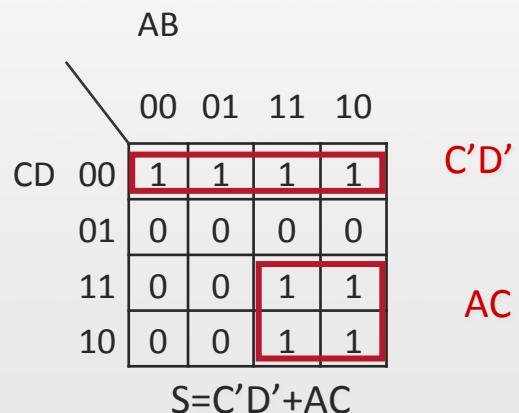
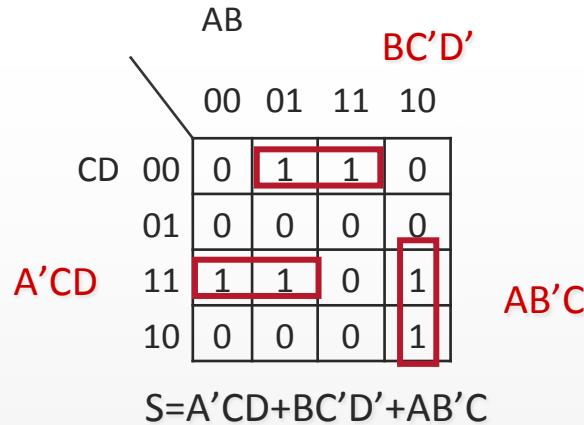
$$S = A'BC + ABC = BC(A' + A) = BC$$

S	BC				
	00	01	11	10	
A	0	0	1	1	0
	1	1	1	1	0

$$S = C$$

# Diseño. Mapas de Karnaugh (III)

Agrupamientos permitidos.



# Diseño. Mapas de Karnaugh (IV)

Agrupamientos permitidos.

S	AB	00	01	11	10
CD	00	0	0	0	0
01	1	1	1	1	
11	1	1	1	1	
10	0	0	0	0	

S=D

D

S	AB	00	01	11	10
CD	00	1	0	0	1
01	1	1	0	0	1
11	1	0	0	1	
10	1	0	0	1	

S=B'

B'

S	AB	00	01	11	10
CD	00	0	1	1	0
01	0	0	1	1	
11	0	0	1	1	
10	1	0	0	0	

S=BC'D'+AD+A'B'CD'

BC'D'  
AD

A'B'CD'

S	AB	00	01	11	10
CD	00	1	1	1	1
01	1	1	1	1	
11	1	1	1	1	
10	1	1	1	1	

S=1

AB

B'D'

# Diseño. Mapas de Karnaugh (V)

Agrupamientos no permitidos.

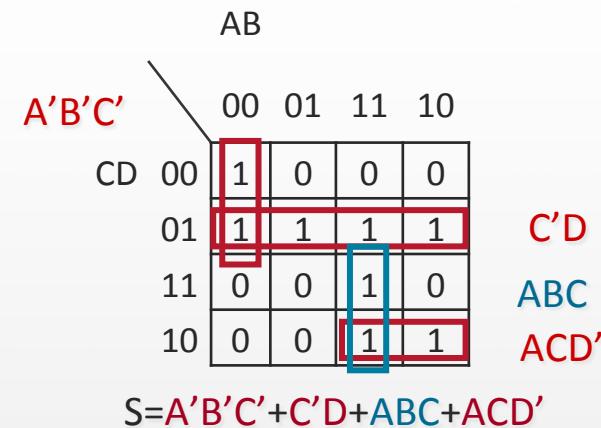
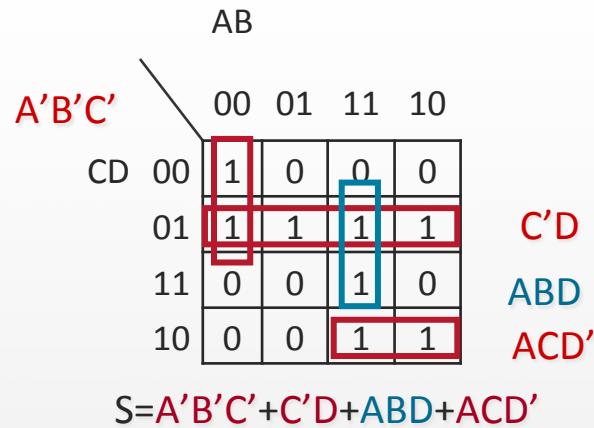
S	AB	00	01	11	10
CD	00	1	0	1	0
	01	1	0	1	0
	11	0	1	0	0
	10	0	1	1	1

S	AB	00	01	11	10
CD	00	1	0	0	0
	01	0	1	0	0
	11	0	0	1	0
	10	0	0	0	1

No se permiten agrupamientos en L, ni diagonales

# Diseño. Mapas de Karnaugh (VI)

Agrupamientos alternativos.



Para un mismo mapa, la solución no tiene por qué ser única.

# Diseño. Mapas de Karnaugh (VII)

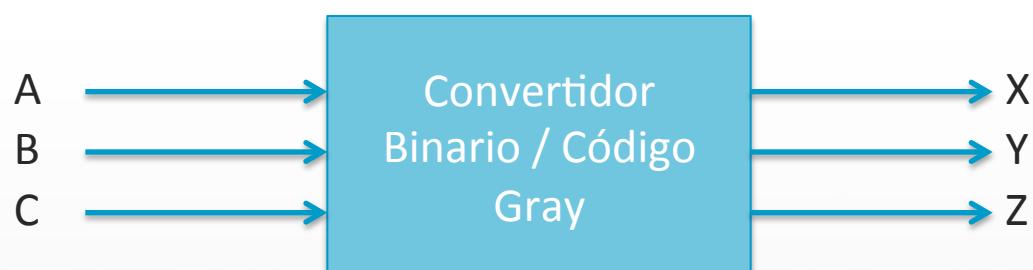
Sistematización del método.

1º) Representar en un mapa de Karnaugh la función lógica o tabla de verdad que se deseé minimizar.

2º) Se agruparan los “1” siguiendo las reglas que a continuación se citan:

- Deberán construirse primero los grupos de celdas más grandes; cada uno deberá contener  $2^n$  elementos.
- Deberán agregarse grupos cada vez más pequeños, hasta que cada celda que contenga un “1” se haya incluido por lo menos una vez.
- Deberán eliminarse los grupos redundantes (aun cuando se trate de grupos grandes) para evitar la duplicación.

## Diseño. Ejemplo



A	B	C	X	Y	Z
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	1
0	1	1	0	1	0
1	0	0	1	1	0
1	0	1	1	1	1
1	1	0	1	0	1
1	1	1	1	0	0

X	AB	00	01	11	10
C	0	0	0	1	1
C	1	0	0	1	1

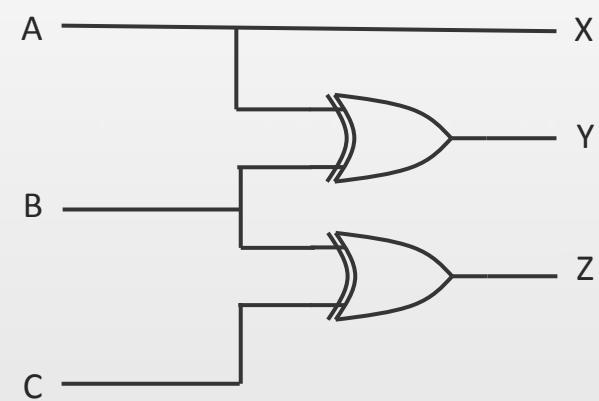
$X = A$

Y	AB	00	01	11	10
C	0	0	1	0	1
C	1	0	1	0	1

$X = A'B + AB' = A \oplus B$

Z	AB	00	01	11	10
C	0	0	1	1	0
C	1	1	0	0	1

$X = BC' + B'C = B \oplus C$



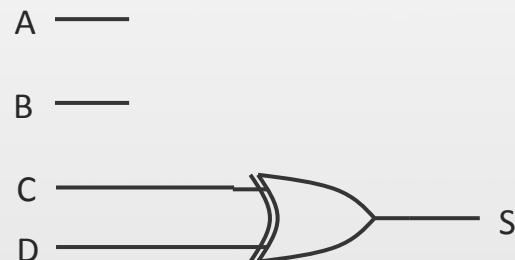
# Diseño. Condiciones indiferentes

Diseñar un circuito que tome un número BCD y produzca una sola salida Y que esté activa si la entrada es: 1, 2, 5, 6 ó 9

Decimal	A	B	C	D	S
0	0	0	0	0	0
1	0	0	0	1	1
2	0	0	1	0	1
3	0	0	1	1	0
4	0	1	0	0	0
5	0	1	0	1	1
6	0	1	1	0	1
7	0	1	1	1	0
8	1	0	0	0	0
9	1	0	0	1	1
10	1	0	1	0	X
11	1	0	1	1	X
12	1	1	0	0	X
13	1	1	0	1	X
14	1	1	1	0	X
15	1	1	1	1	X

S	AB	00	01	11	10
CD	00	0	0	X	0
	01	1	1	X	1
	11	0	0	X	X
	10	1	1	X	X

$$S = C'D + CD' = C \oplus D$$

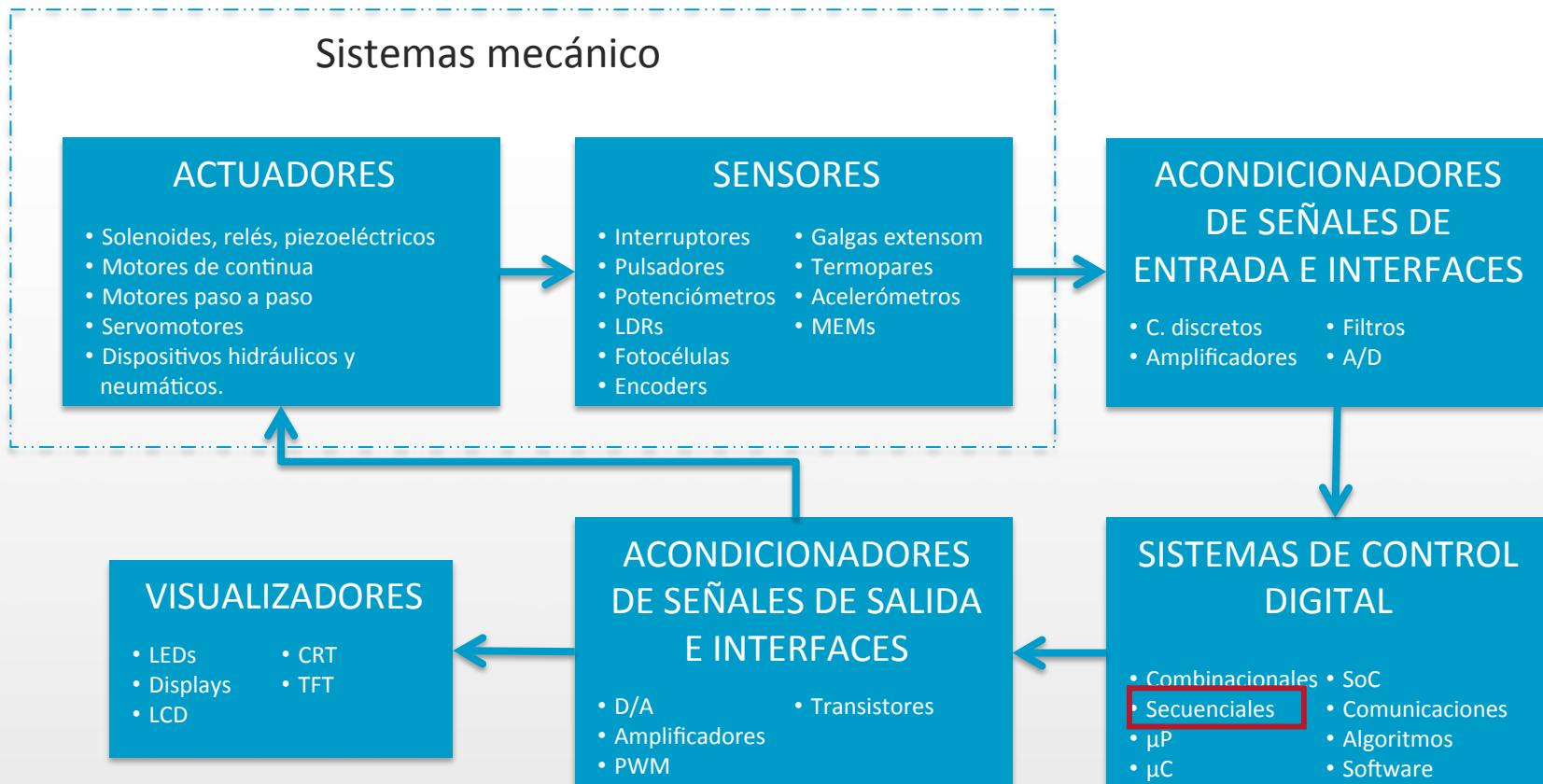


## Diseño propuesto

En un barco el piloto automático controla la navegación e indica mediante cuatro señales N, S, E y O que rumbo lleva. Diseñar el menor circuito que decodifique el rumbo sobre un display de 7-segmentos, según el siguiente criterio:

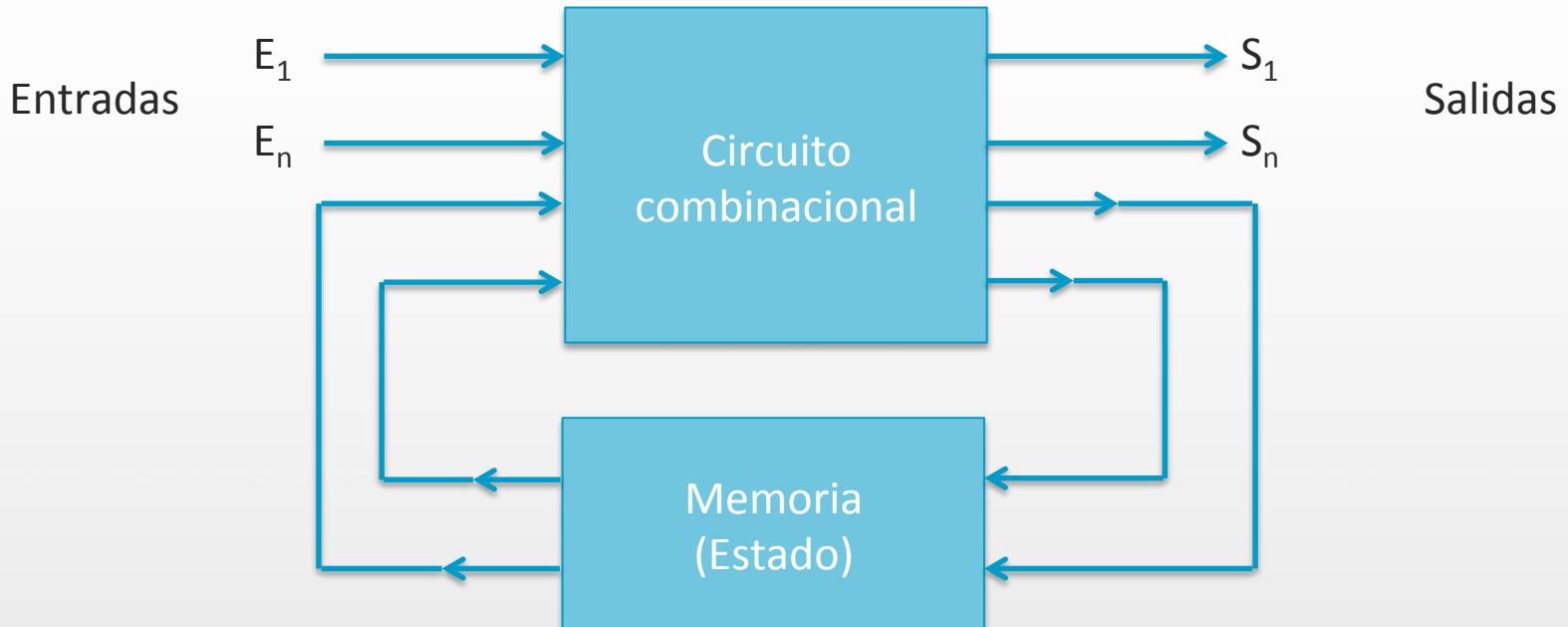
- Si sigue rumbo norte, se activa el segmento a; si sur, el d.
- Si sigue rumbo este, se activan los segmentos b y c; si oeste, se activan e y f.
- Si sigue rumbo noreste se activan a y b; si noreste, a y f.
- Si sigue rumbo sureste se activan c y d; si suroeste, d y e.

# Circuitos digitales Secuenciales



# Circuitos digitales secuenciales.

Los **circuitos secuenciales** se caracterizan porque las salidas están determinadas no sólo por las entradas existentes sino también por la secuencia de entradas que condujeron al estado existente. ==> **El circuito tiene MEMORIA**.

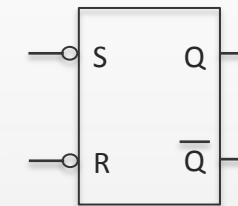
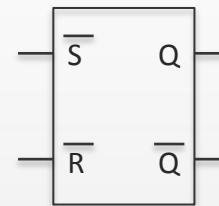
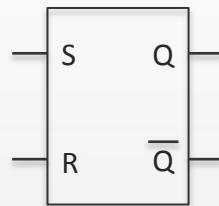


Se pueden clasificar en SÍNCRONOS y ASÍNCRONOS

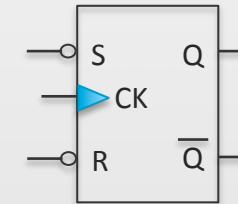
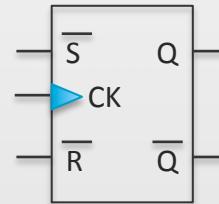
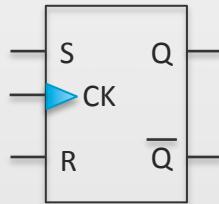
# Flip-Flops.

- Los flip-flops **son los elementos más sencillos para almacenar información**.
- Se caracterizan porque tienen **dos y sólo dos estados posibles** de salida. Al ser dichos estados “ESTABLES” se conocen también como **biestables**.
- Dentro de los diferentes tipos de flip-flops existentes, los más usados son los **Latches** y los **Flip-Flops activados por flanco**

## Latches



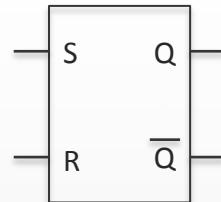
## Flip-Flops activados por flanco



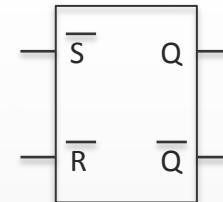
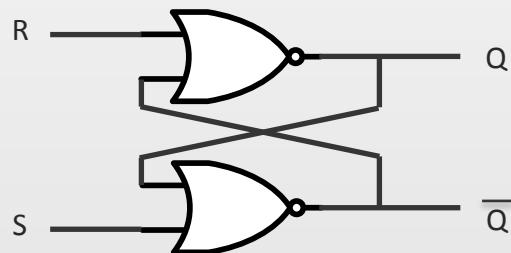
# Latches.

Los **latches** (cerros) son los biestables más sencillos ya que no precisan señal de reloj para su activación. Se utilizan poco (circuitos asíncronos).

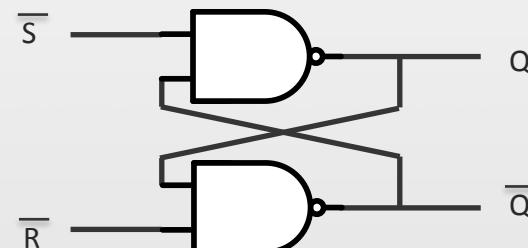
**Latch SR**



S	R	$Q_n$	$Q'_n$	Estado
0	0	$Q_{n-1}$	$Q'_{n-1}$	Sin cambio
0	1	0	1	RESET
1	0	1	0	SET
1	1	0	0	Estado Ambiguo



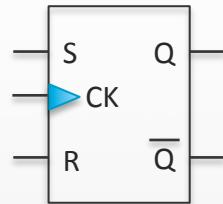
$S'$	$R'$	$Q_n$	$Q'_n$	Estado
0	0	1	1	Estado Ambiguo
0	1	1	0	SET
1	0	0	1	RESET
1	1	$Q_{n-1}$	$Q'_{n-1}$	Sin Cambio



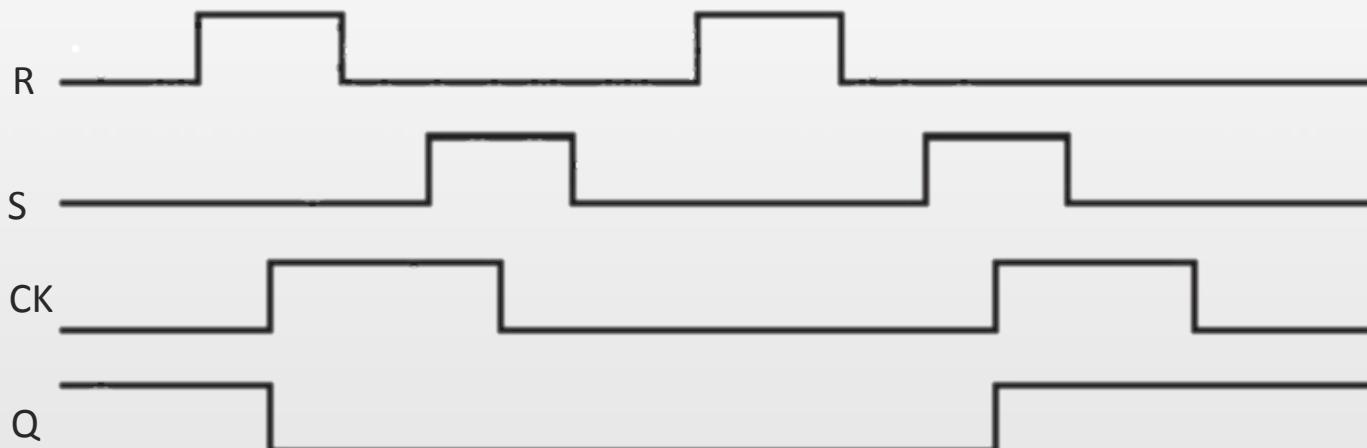
# Flip-Flops activados por flanco (I).

En los flip-flops la señal de reloj (CK) sincroniza todos los cambios de los estados de salida del dispositivo. Esto permite el diseño de circuitos complejos ( $\mu$ P y  $\mu$ C), donde todos los cambios se disparan o activan con una señal de reloj común.

## Flip-Flops SR

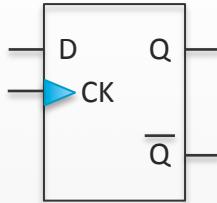


S	R	CK	$Q_n$	$Q'_{n-1}$
0	0	↑	$Q_{n-1}$	$Q'_{n-1}$
0	1	↑	0	1
1	0	↑	1	0
1	1	↑		NA
X	X	0, 1, ↓	$Q_{n-1}$	$Q'_{n-1}$

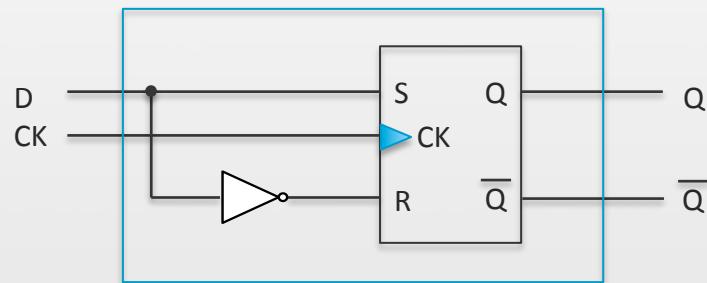
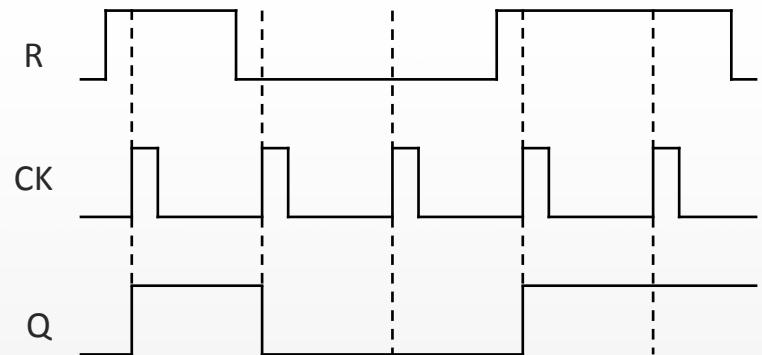


# Flip-Flops activados por flanco (II).

## Flip-Flops tipo D

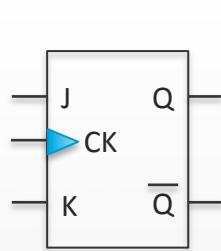


D	CK	$Q_n$	$Q'_n$
0	↑	0	1
1	↑	1	0
X	0, 1, ↓	$Q_{n-1}$	$Q'_{n-1}$

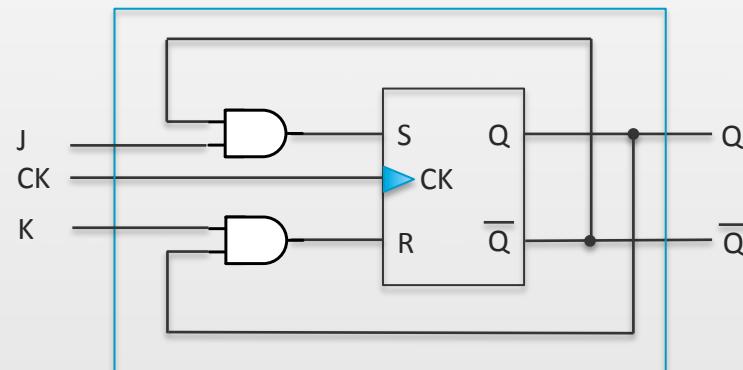
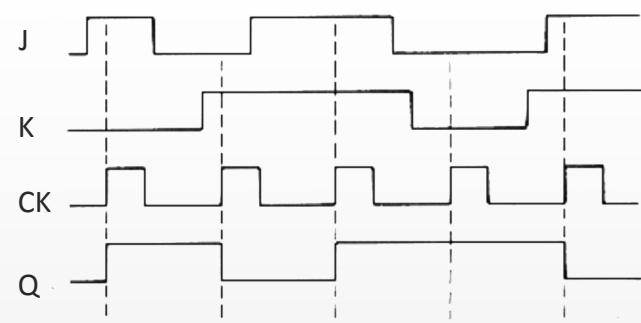


# Flip-Flops activados por flanco (III).

## Flip-Flops tipo JK

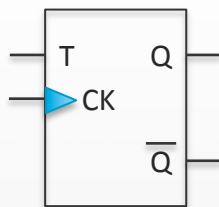


J	K	CK	$Q_n$	$Q'_n$
0	0	↑	$Q_{n-1}$	$Q'_{n-1}$
0	1	↑	0	1
1	0	↑	1	0
1	1	↑	$Q'_{n-1}$	$Q_{n-1}$
X	X	0, 1, ↓	$Q_{n-1}$	$Q'_{n-1}$

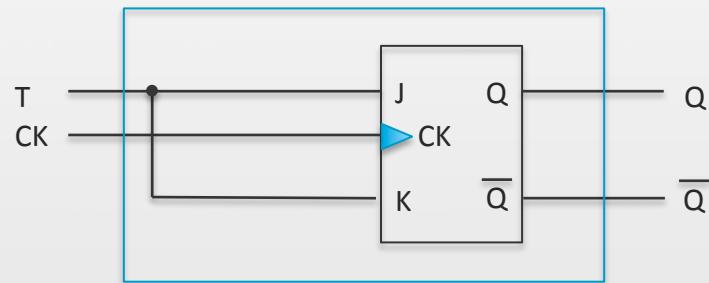
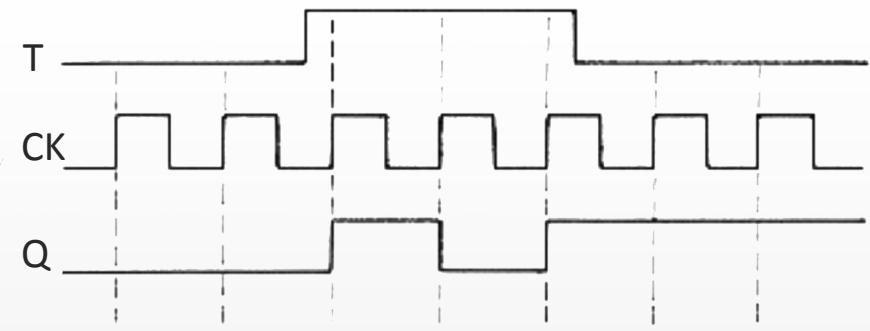


# Flip-Flops activados por flanco (IV).

## Flip-Flops tipo T



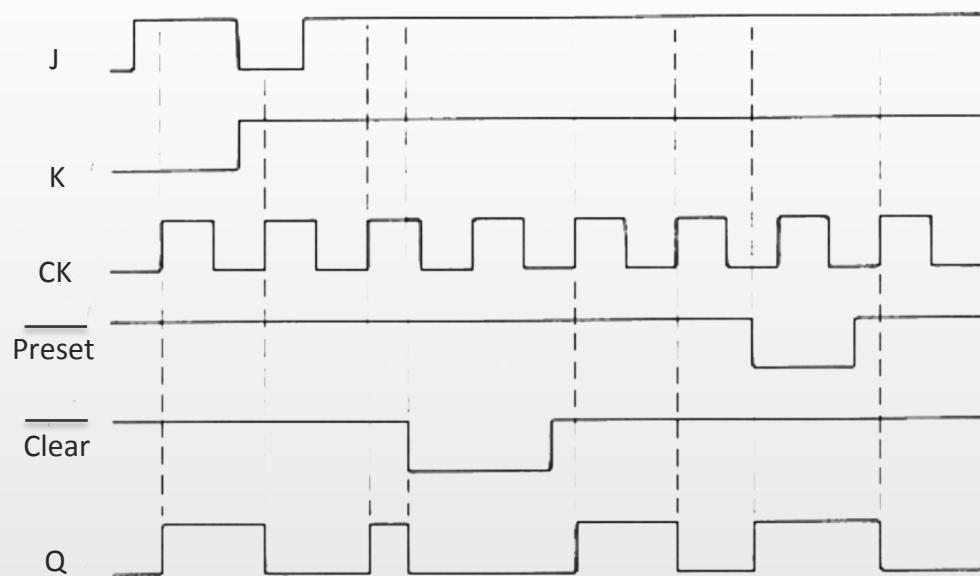
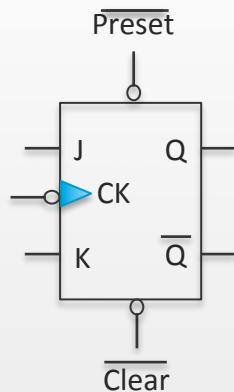
T	CK	$Q_n$	$Q'_n$
0	↑	$Q_{n-1}$	$Q'_{n-1}$
1	↑	$Q'_{n-1}$	$Q_{n-1}$
X	0, 1, ↓	$Q_{n-1}$	$Q'_{n-1}$



# Flip-Flops activados por flanco (V).

## Entradas de control asíncronas

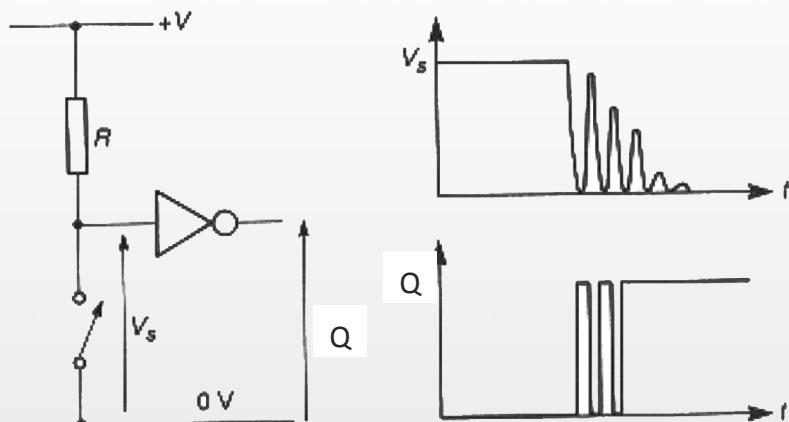
Se utilizan para poner la salida de un flip-flop a “0” o a “1” en cualquier momento. Reciben diferentes denominaciones (PRESET, CLEAR, DC SET, DC CLEAR, SET, RESET, DIRECT SET, DIRECT CLEAR).



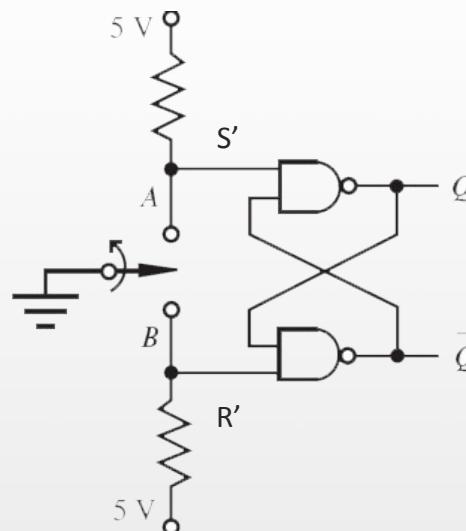
# Aplicaciones de los Flip-Flops (I).

## Eliminación de rebotes (debouncing) en interruptores

Se utilizan para poner la salida de un flip-flop a “0” o a “1” en cualquier momento. Reciben diferentes denominaciones (PRESET, CLEAR, DC SET, DC CLEAR, SET, RESET, DIRECT SET, DIRECT CLEAR).



Interruptor SPST  
(1 polo, 1 dirección)



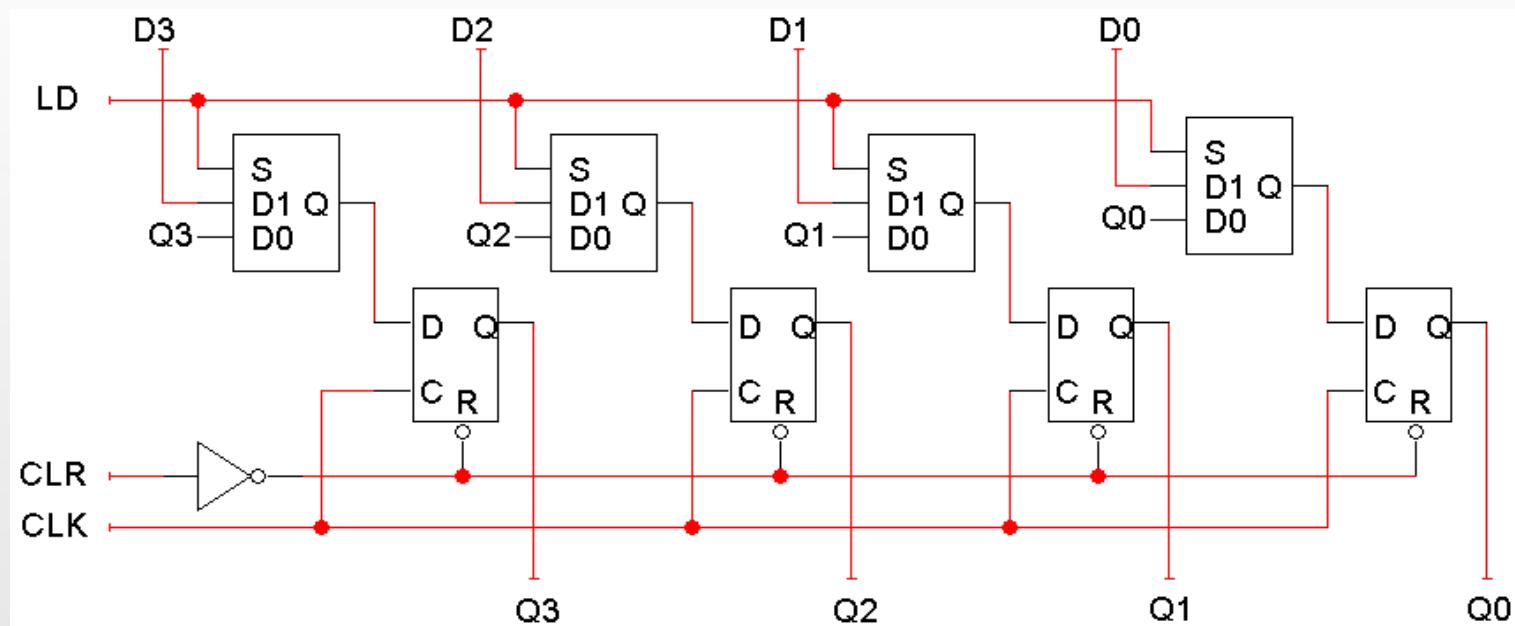
Interruptor SPDT  
(1 polo, 2 direcciones)

# Aplicaciones de los Flip-Flops (II).

## Registros de datos

Los FF se pueden poner en cascada para construir registros de datos de tantos bits como se quiera. Configuraciones típicas son de 4, 8, 16, 32, 64 o 128 bits. Se suelen utilizar en los microprocesadores para retener datos para cálculos aritméticos.

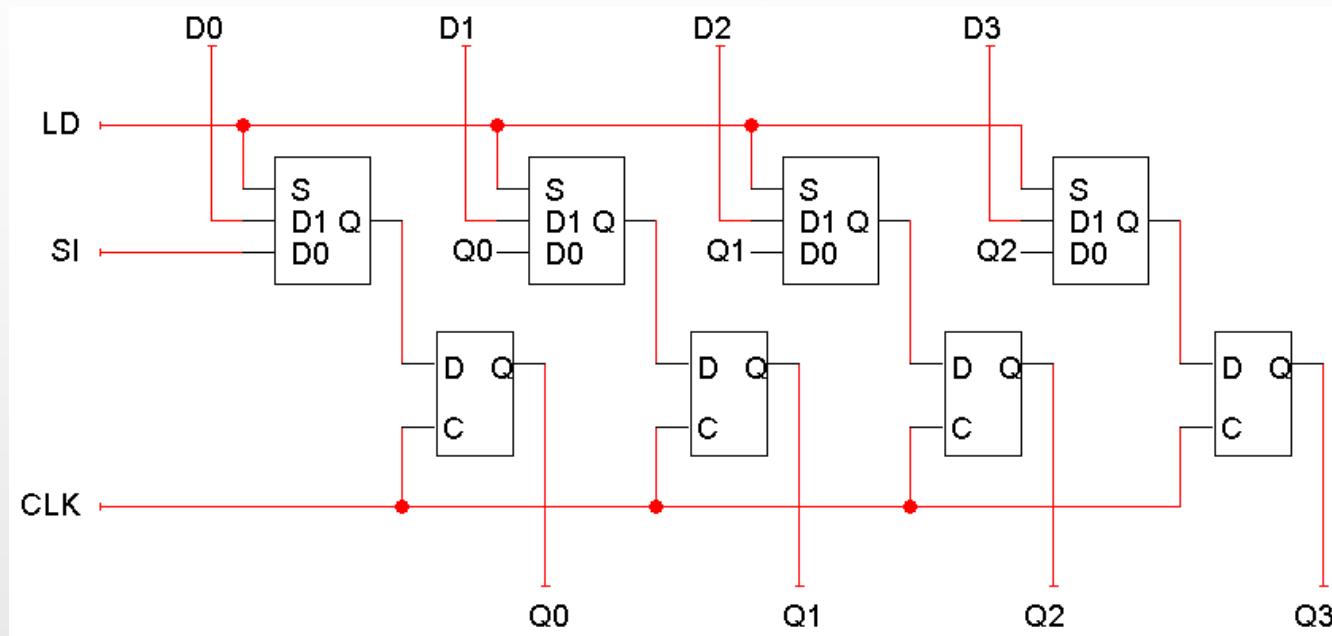
Los Registros son más caros que los dispositivos de memoria de las CPU pero mucho más rápidos.



# Aplicaciones de los Flip-Flops (III).

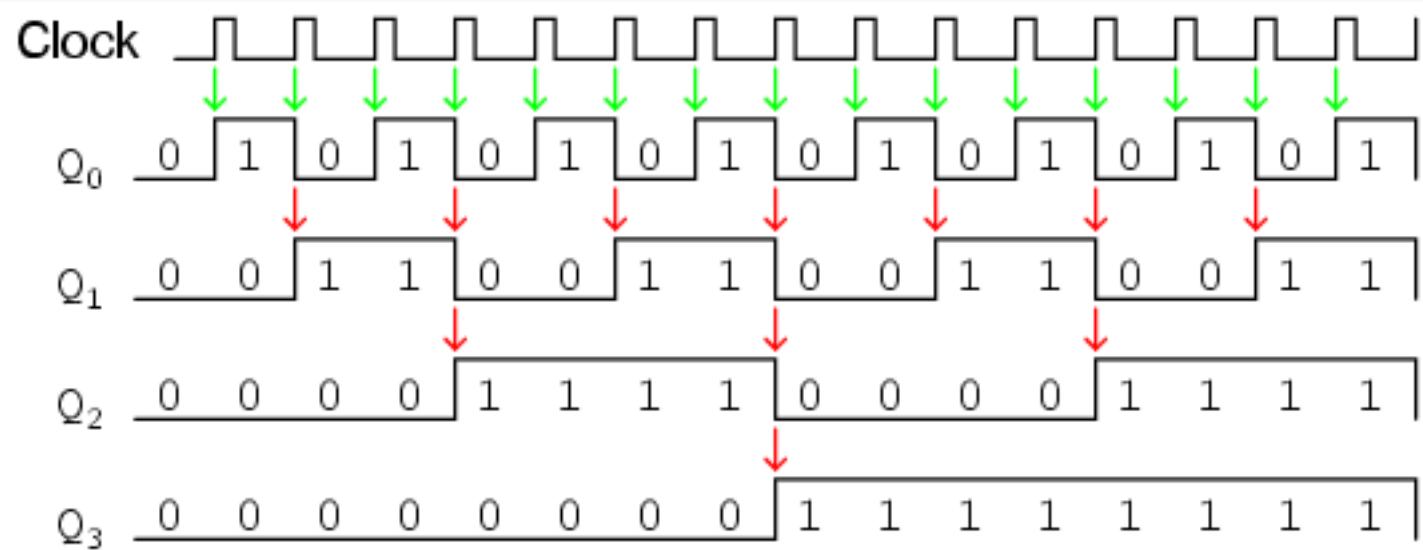
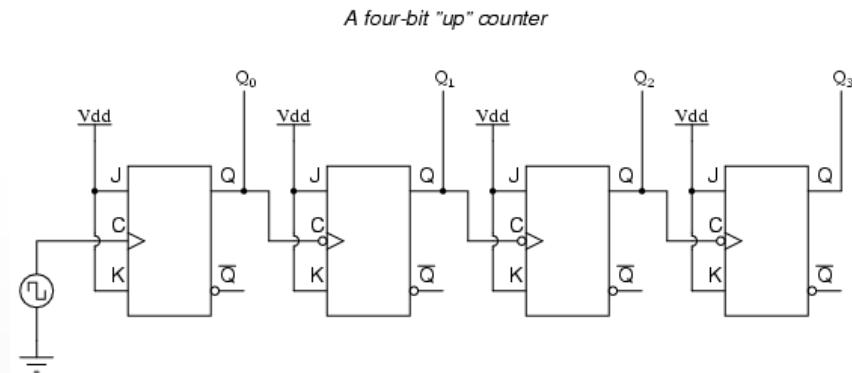
## Registros de desplazamiento

Los registros de desplazamiento se suelen utilizar para implementar convertidores paralelo serie-serie paralelo con objeto de comunicar dispositivos (USB, Fireware, HD Serial ATA, etc.).



# Aplicaciones de los Flip-Flops (IV).

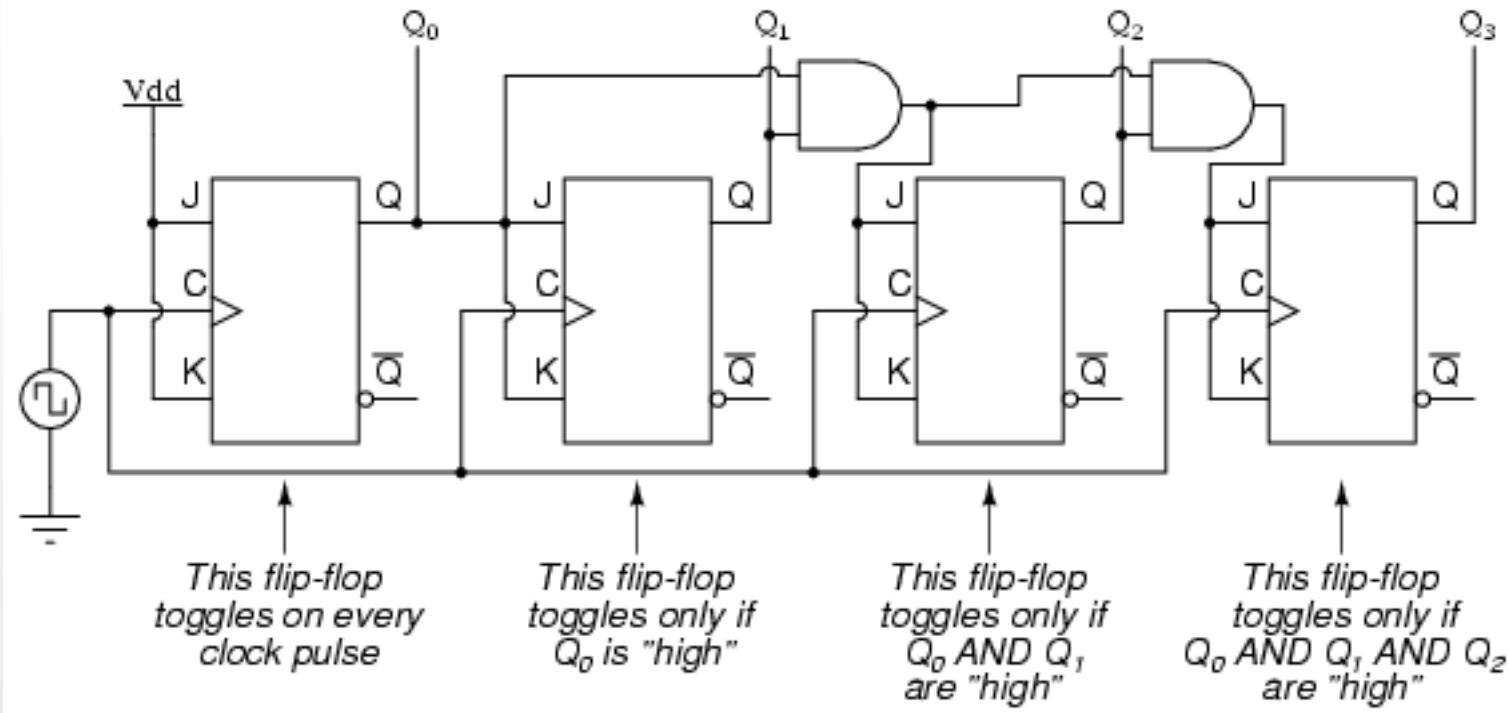
## Contadores asíncronos



# Aplicaciones de los Flip-Flops (V).

## Contadores síncronos

*A four-bit synchronous "up" counter*



# Familias Lógicas

- Una **familia lógica** es una colección de Circuitos Integrados que tienen características eléctricas similares en cuanto a sus entradas, salidas y circuitería interna. Las familias más usuales son la CMOS (Complementary Metal-Oxide-Semiconductor Field Effect Transistor), TTL (Transistor Transistor Logic) y ECL (Emitter Coupled Logic).
- El diseño lógico de un circuito digital es independiente de la tecnología usada, sin embargo para la realización física de los circuitos si se debe tener en cuenta. En concreto, hay que considerar los siguientes factores:
  - Márgenes de ruido
  - Fan-out
  - Velocidad
  - Consumo
  - Alimentación disponible
  - Entorno de trabajo del circuito
  - Necesidad de:
    - Salidas en colector abierto
    - Salidas Tri-State

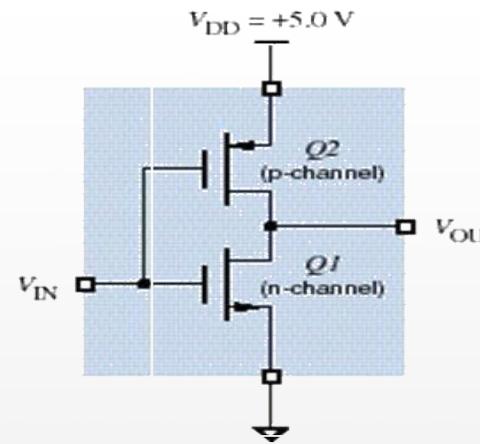
Los chips de una misma familia lógica se pueden interconectar directamente

Los chips de familias lógicas diferentes no tienen porque ser compatibles

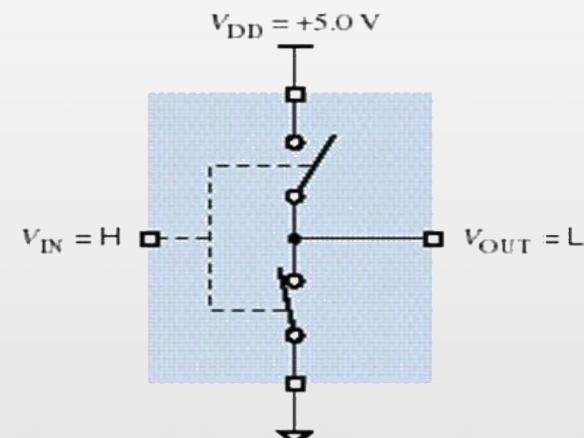
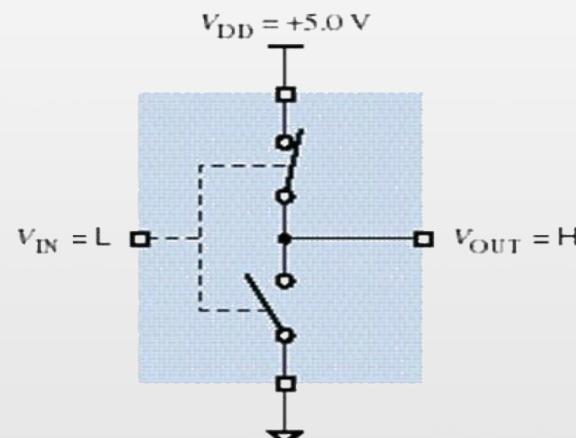
# Familias Lógicas. Tecnología CMOS

La lógica CMOS utiliza pares complementarios MOSFET (NMOS y PMOS) como elemento básico.

## Inversor CMOS



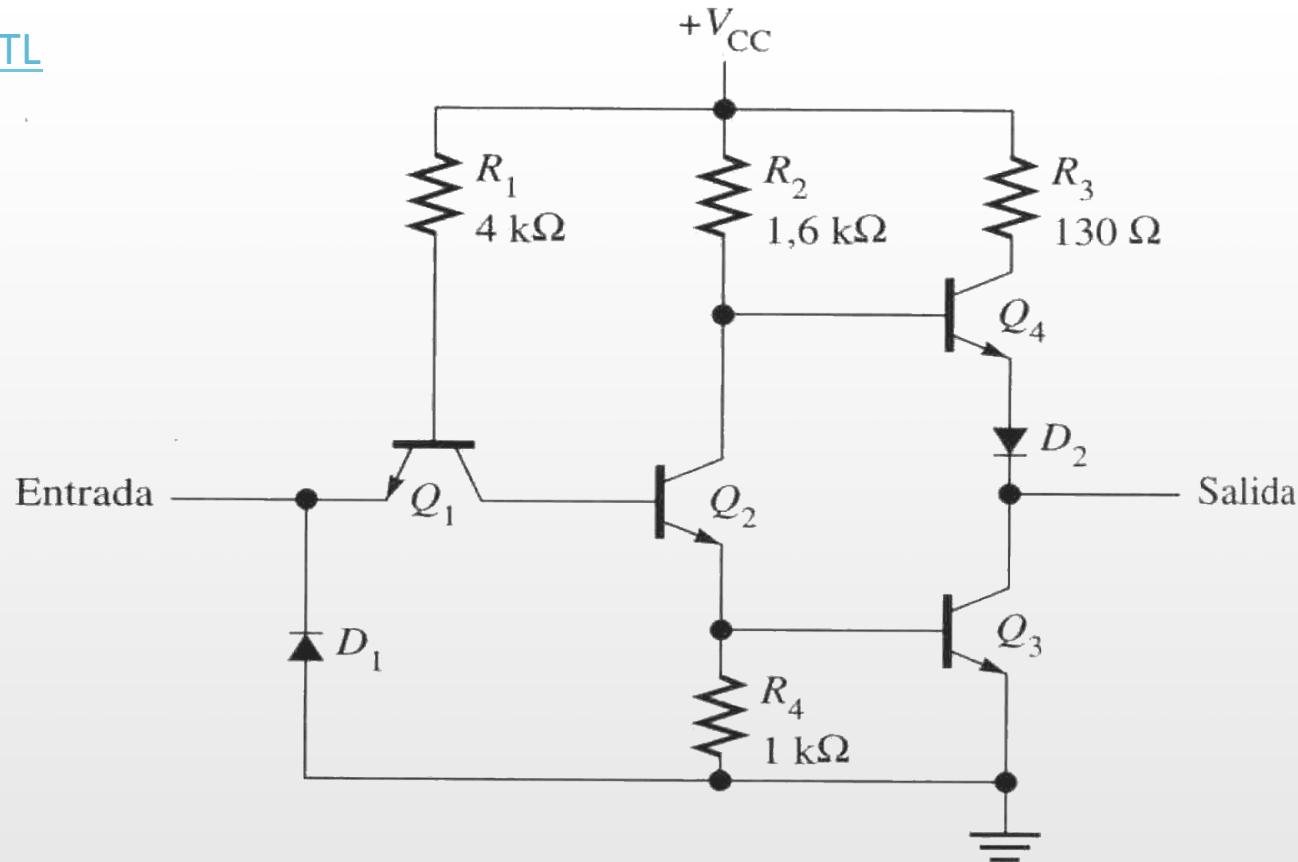
$V_{IN}$	$Q1$	$Q2$	$V_{OUT}$
0.0 (L)	off	on	5.0 (H)
5.0 (H)	on	off	0.0 (L)



# Familias Lógicas. Tecnología TTL

La lógica TTL utiliza transistores bipolares como elementos básicos

## Inversor TTL

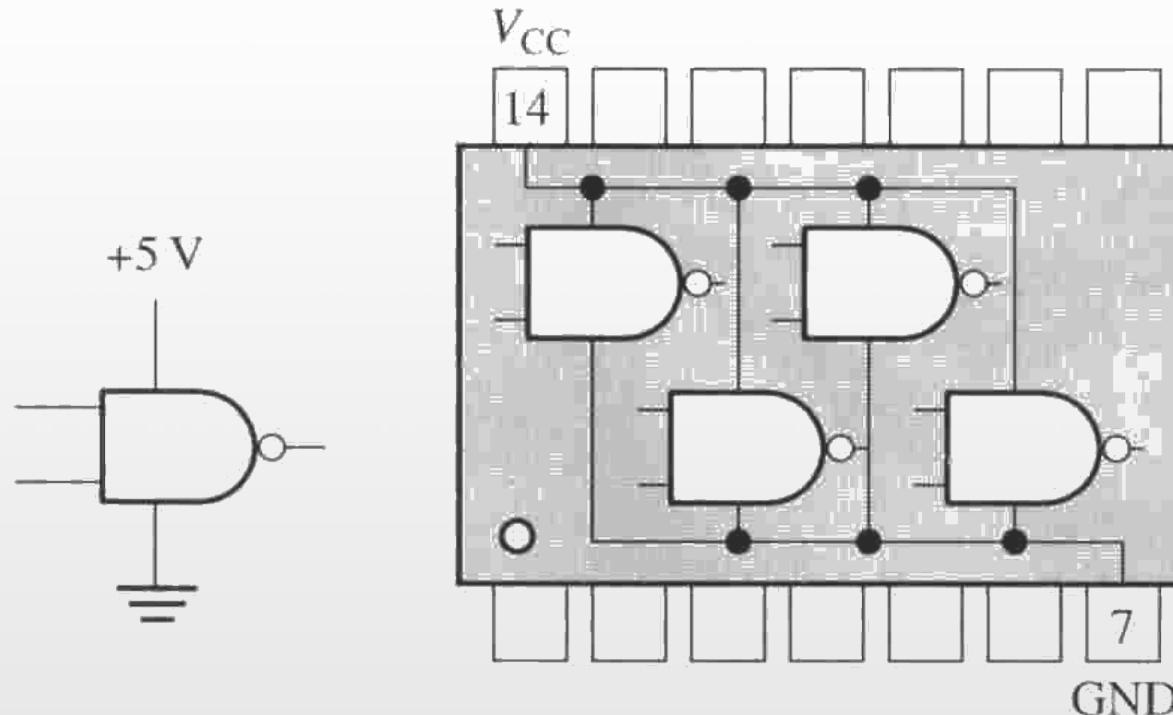


# Parámetros característicos (I)

## Tensión de alimentación

TTL => 5V

CMOS => 5V, 3.3V y 2.7V



## Parámetros característicos (II)

### Potencia consumida



Si el ciclo de trabajo es el 50% =>  $I_{CC} = (I_{CCH} + I_{CCL})/2 \Rightarrow P_D = V_{CC} \cdot I_{CC}$

La disipación de potencia en un circuito TTL se puede considerar constante dentro de su rango de frecuencias de operación.

En CMOS la disipación de potencia depende de la frecuencia. En condiciones estáticas es extremadamente baja y aumenta cuando aumenta la frecuencia.

**Los CMOS presentan baja disipación estática y una significativa disipación dinámica.**

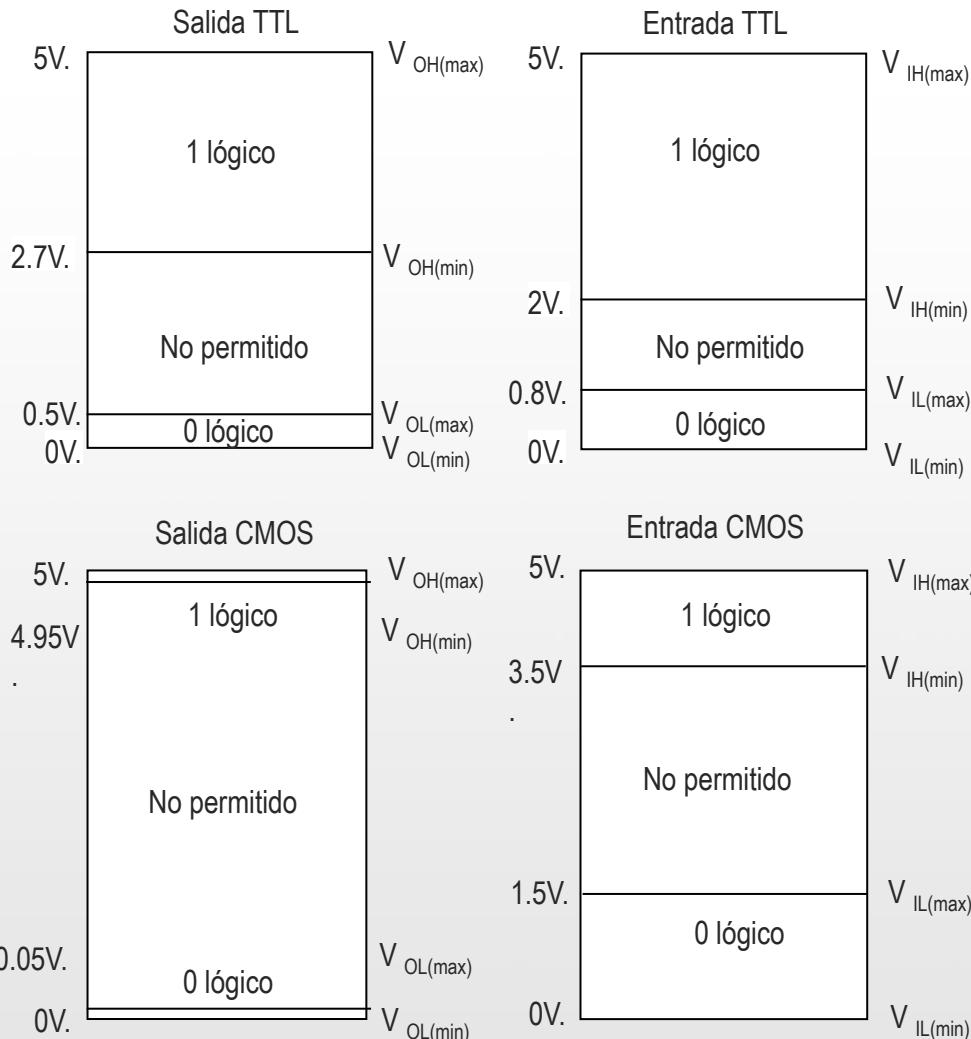
Por ejemplo:

TTL Schottkly => 2,2 mW (siempre).

HCMOS => 2,75  $\mu$ W (DC) y 170  $\mu$ W (100 KHz).

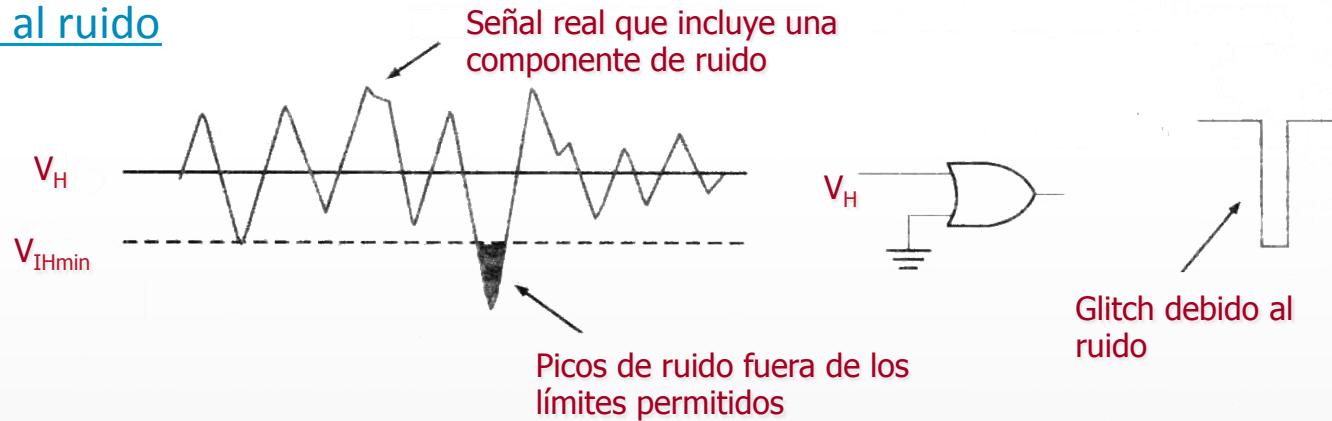
# Parámetros característicos (III)

## Niveles de entrada y salida CMOS y TTL



# Parámetros característicos (IV)

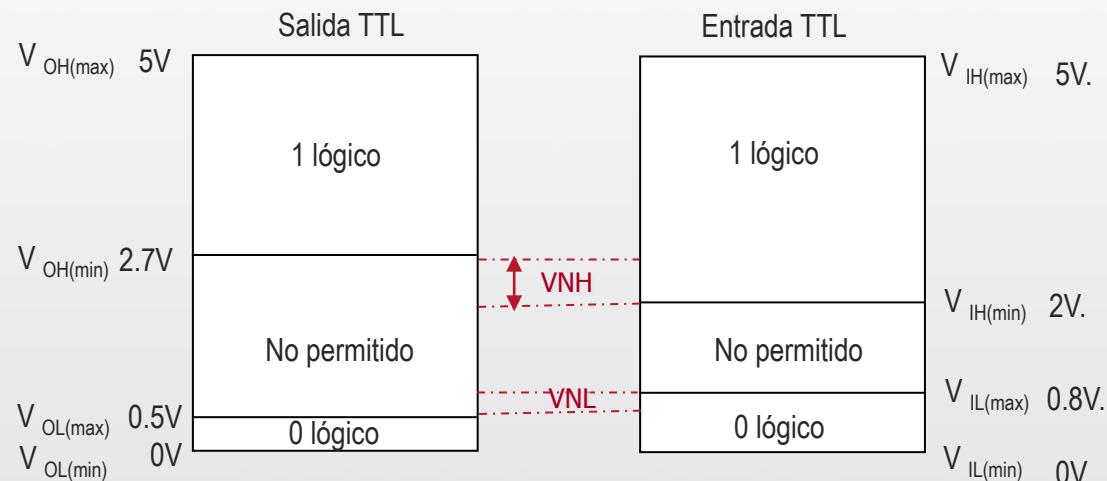
## Inmunidad al ruido



## Márgenes de ruido

$$V_{NH} = V_{OH(\min)} - V_{IH(\min)}$$

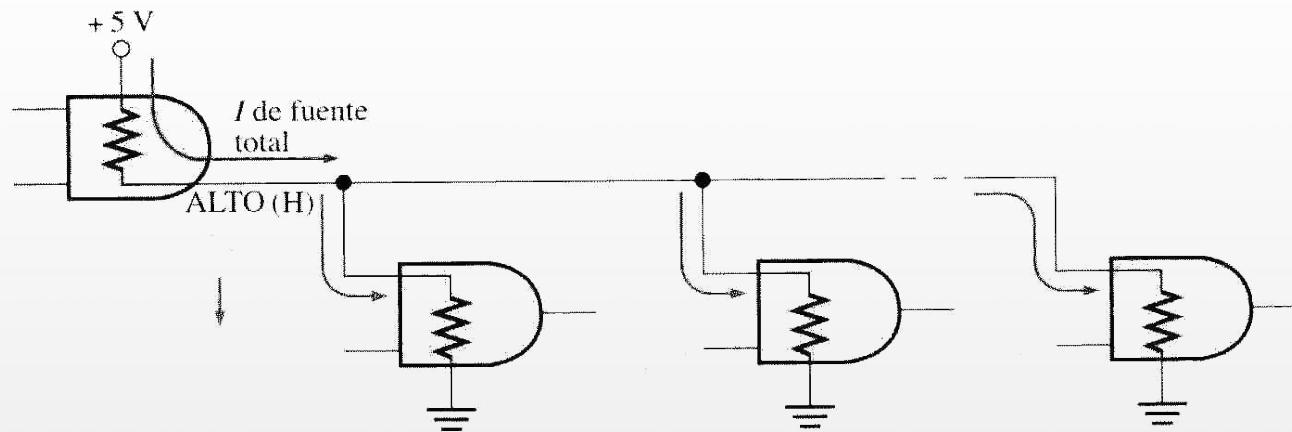
$$V_{NL} = V_{IL(\max)} - V_{OL(\max)}$$



# Parámetros característicos (V)

## Capacidad de carga de una puerta TTL cuando la salida está nivel ALTO

Una puerta excitadora TTL (fuente) entrega corriente a las entradas de las puertas de carga en el estado ALTO ( $I_{IH}$ ) y absorbe corriente de las puertas de carga en el estado BAJO ( $I_{IL}$ ).

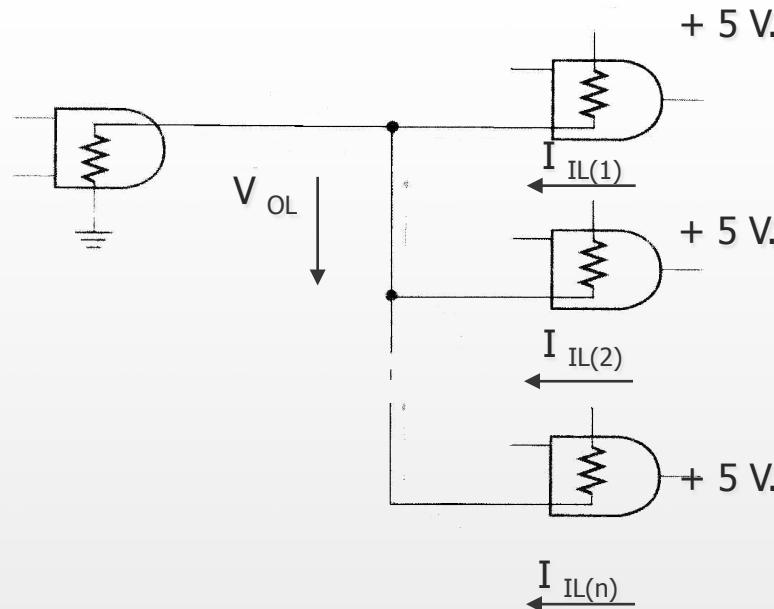


Cuantas más puertas de carga se conectan a la fuente, mayor es la carga de la misma. La corriente que suministra la fuente aumenta con cada puerta de carga que se añade. Al aumentar esta corriente, la caída de tensión interna de la puerta excitadora aumenta., haciendo que la tensión de salida  $V_{OH}$  disminuya. Si se conecta un número excesivo de puertas de carga, la tensión  $V_{OH}$  cae por debajo de su valor mínimo  $V_{OH(min)}$

# Parámetros característicos (VI)

## Capacidad de caga de una puerta TTL cuando su salida está a nivel BAJO

Una puerta excitadora TTL (sumidero) absorbe corriente de las puertas de carga en el estado BAJO ( $I_{IL}$ ).



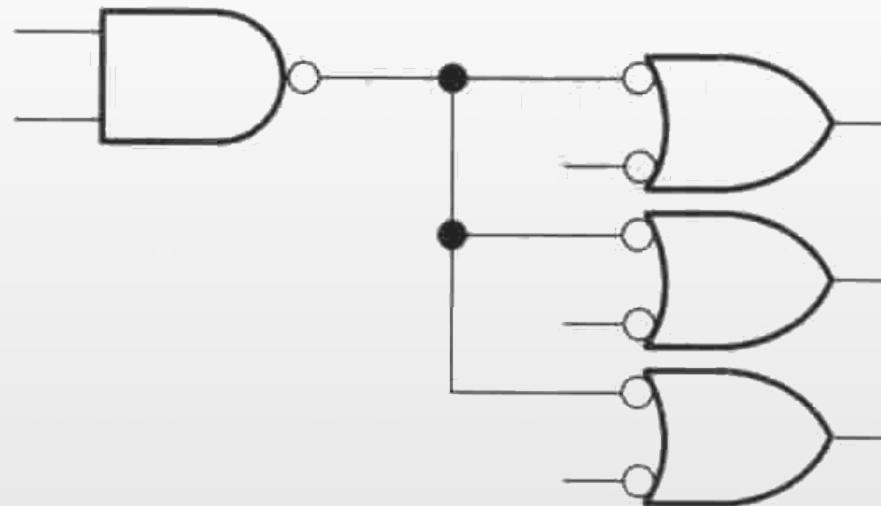
La corriente de sumidero también aumenta con cada entrada de carga que se añade. Al aumentar esta corriente, la caída de tensión interna de la puerta excitadora aumenta, haciendo que  $V_{OL}$  aumente. Si se conecta un número excesivo de puertas de carga, la tensión  $V_{OL}$  se hará mayor que  $V_{OL(max)}$

# Parámetros Característicos (VII)

## Fan-out

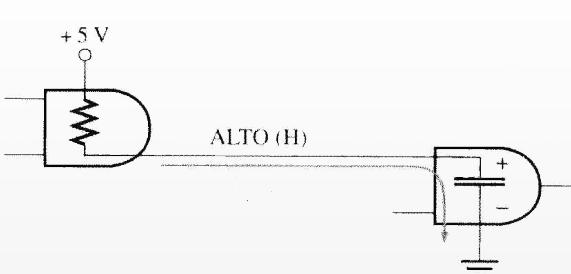
Existe un límite para el número de entradas de carga que una puerta puede excitar. Este límite se denomina **fan-out** de la puerta.

El fan-out para las puertas TTL standard es de 10 entradas y hasta 40 para TTL Schottky.

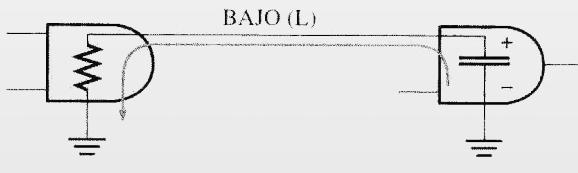
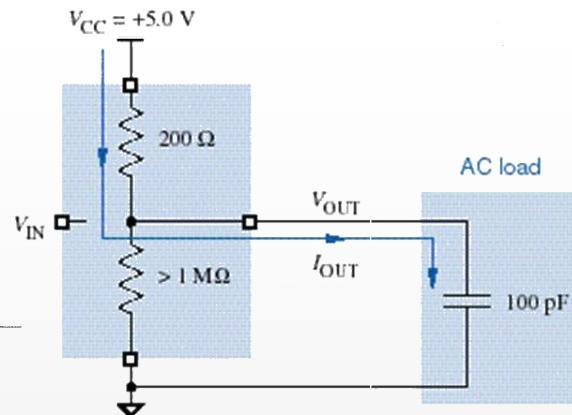


# Parámetros característicos (VIII)

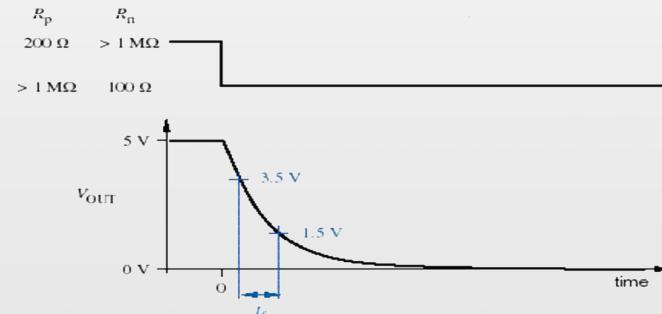
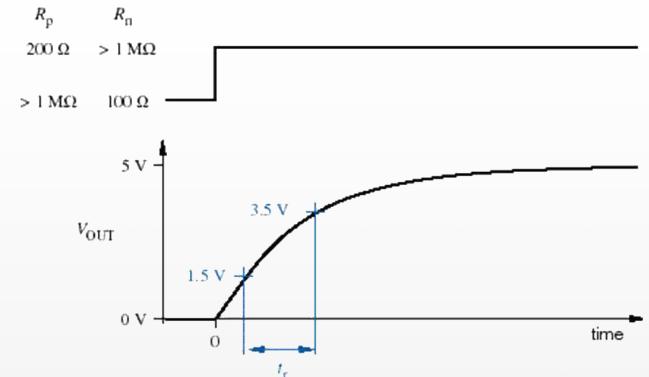
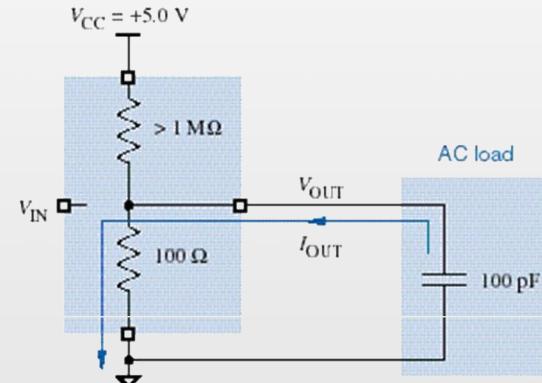
## Comportamiento dinámico de las puertas CMOS



Carga



Descarga



# Hojas de datos de los fabricantes de CI

## Familia TTL

Familia	Características
74	Es la más antigua, fue introducida por Silvana en 1963 y popularizada por Texas Instruments.
74H 74L	High Speed TTL Low Power TTL Tienen la misma estructura pero cambian los valores de los resistores
	El desarrollo de los transistores Schottky y su introducción en los años 70 en la familia TTL hizo obsoletas las familias 74, 74H, 74L
74S	Schottky TTL Es la primera familia que utiliza transistores Schottky Mejora mucho la velocidad de la serie 74 pero con mucho más consumo.
74LS	Low power Schottky TTL Es la TTL más utilizada y la menos costosa Iguala la velocidad de la serie 74 TTL pero consume una quinta parte.
74AS	Advanced Shottky TTL Ofrece el doble de velocidad que la 74S con la mitad de consumo
74ALS	Advanced Low Power Schottky TTL Ofrece velocidades y consumos mejores que la LS. Rivaliza con la LS
74F	Fast TTL Esta posicionada entre la AS y la ALS

# Hojas de datos de los fabricantes de CI

## Familia CMOS

Familia	Características
4000	Es la más antigua, ha sido sustituida por el resto de familias. Eran lentas, presentaban un bajo consumo frente a las TTL de la época. Pero se conectaban mal con las TTL
	74 FAM nn 74HC30, 74HCT30, 74AC30, 74HCT30, 74AHC30 Puertas NAND de 8 entradas
HC HCT	High Speed CMOS High Speed CMOS, TTL Compatible Tienen mayor velocidad y mejor capacidad de consumo y de suministro de corriente que la 4000
VHC VHCT	Very High Speed CMOS Very High Speed CMOS, TTL Compatible Son el doble de rápidas que las HC y HCT, siendo compatibles eléctricamente.

# Hojas de datos de los fabricantes de CI

**SN5400, SN54LS00, SN54S00  
SN7400, SN74LS00, SN74S00**  
**QUADRUPLE 2-INPUT POSITIVE-NAND GATES**  
SDLS025 - DECEMBER 1983 - REVISED MARCH 1988

- Package Options Include Plastic "Small Outline" Packages, Ceramic Chip Carriers and Flat Packages, and Plastic and Ceramic DIPs
- Dependable Texas Instruments Quality and Reliability

**description**  
These devices contain four independent 2-input-NAND gates.

The SN5400, SN54LS00, and SN54S00 are characterized for operation over the full military temperature range of  $-55^{\circ}\text{C}$  to  $125^{\circ}\text{C}$ . The SN7400, SN74LS00, and SN74S00 are characterized for operation from  $0^{\circ}\text{C}$  to  $70^{\circ}\text{C}$ .

**FUNCTION TABLE (each gate)**

INPUTS	OUTPUT	
A	B	Y
H	H	L
L	X	H
X	L	H

**logic symbol<sup>†</sup>**

**SN5400 . . . J PACKAGE  
(TOP VIEW)**

1A	1	14	VCC
1B	2	13	4B
1Y	3	12	4A
2A	4	11	4Y
2B	5	10	3B
2Y	6	9	3A
GND	7	8	3Y

**SN5400 . . . W PACKAGE  
(TOP VIEW)**

1A	1	14	4Y
1B	2	13	4B
1Y	3	12	4A
V <sub>CC</sub>	4	11	GND
2Y	5	10	3B
2A	6	9	3A
2B	7	8	3Y

**SN54LS00, SN54S00 . . . FK PACKAGE  
(TOP VIEW)**

NC - No internal connection

**logic diagram (positive logic)**

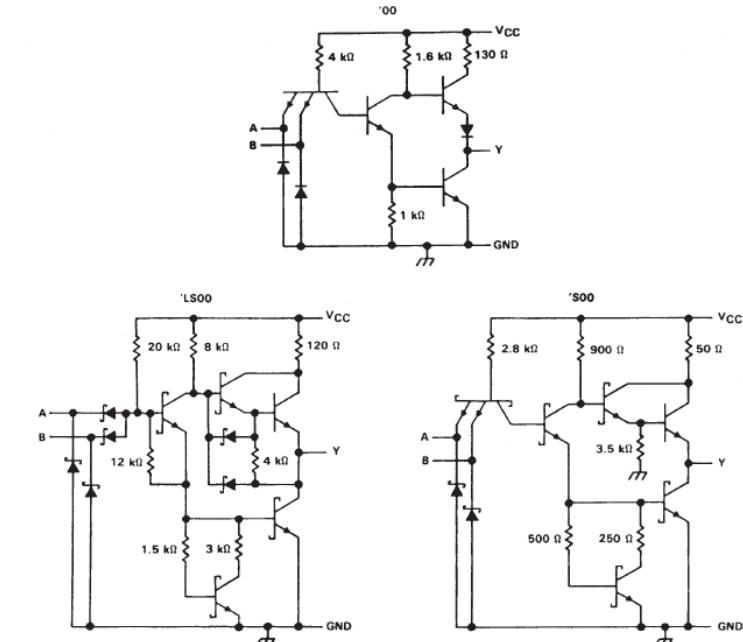
**PRODUCTION DATA Information is current as of publication date. Production processing does not necessarily include testing of all parameters.**

**TEXAS INSTRUMENTS**  
POST OFFICE BOX 655303 • DALLAS, TEXAS 75265

Copyright © 1988, Texas Instruments Incorporated

**SN5400, SN54LS00, SN54S00  
SN7400, SN74LS00, SN74S00**  
**QUADRUPLE 2-INPUT POSITIVE-NAND GATES**  
SDLS025 - DECEMBER 1983 - REVISED MARCH 1988

schematics (each gate)



Resistor values shown are nominal.

absolute maximum ratings over operating free-air temperature range (unless otherwise noted)

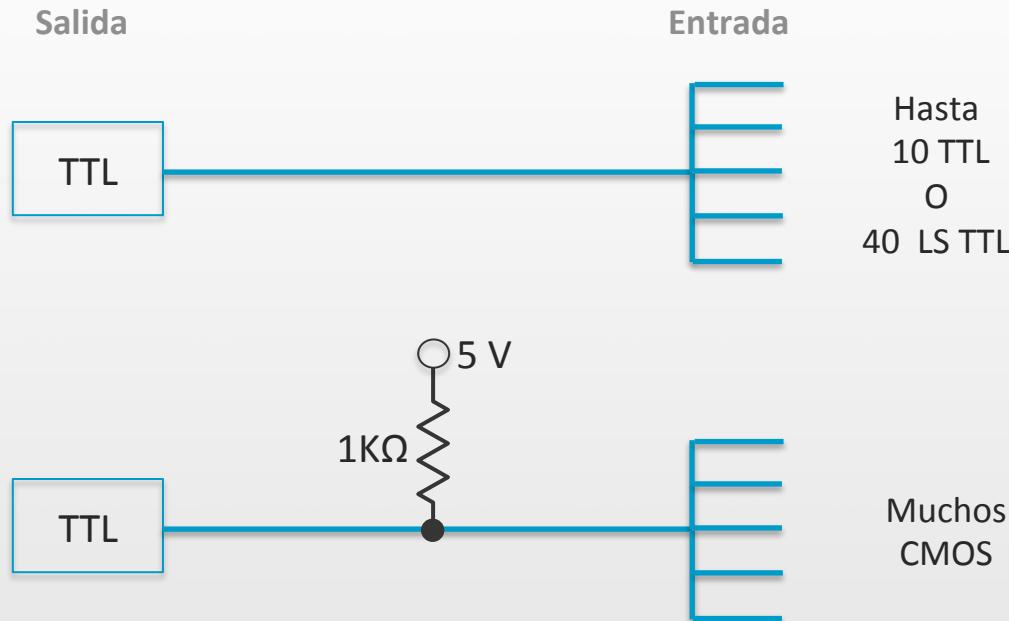
Supply voltage, V <sub>CC</sub> (see Note 1) . . . . .	7 V
Input voltage: '00, 'S00 . . . . .	5.5 V
'LS00 . . . . .	7 V
Operating free-air temperature range: SN54' . . . . .	$-55^{\circ}\text{C}$ to $125^{\circ}\text{C}$
SN74' . . . . .	$0^{\circ}\text{C}$ to $70^{\circ}\text{C}$
Storage temperature range . . . . .	$-65^{\circ}\text{C}$ to $150^{\circ}\text{C}$

NOTE 1: Voltage values are with respect to network ground terminal.

# Conectividad entre CI de diferentes familias

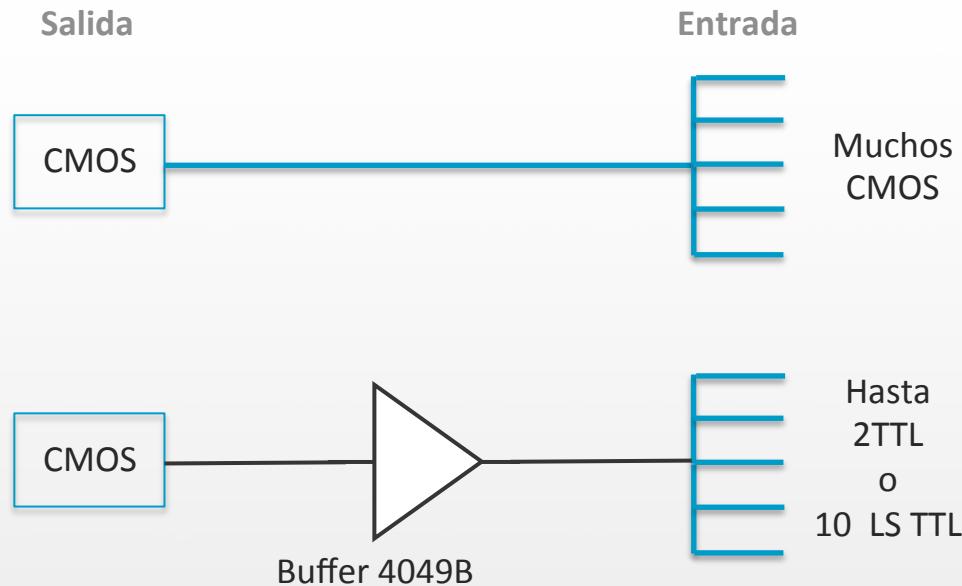
- Cuando se diseñan sistemas digitales se recomienda usar solo una familia de dispositivos (TTL o CMOS) pero a veces puede ser necesario conectar dispositivos de distintas familias, para ello es necesario que sean **compatibles en tensión e intensidad**.

## Conectividad desde TTL a otros dispositivos digitales



# Conectividad entre CI de diferentes familias

## Conectividad desde CMOS a otros dispositivos digitales

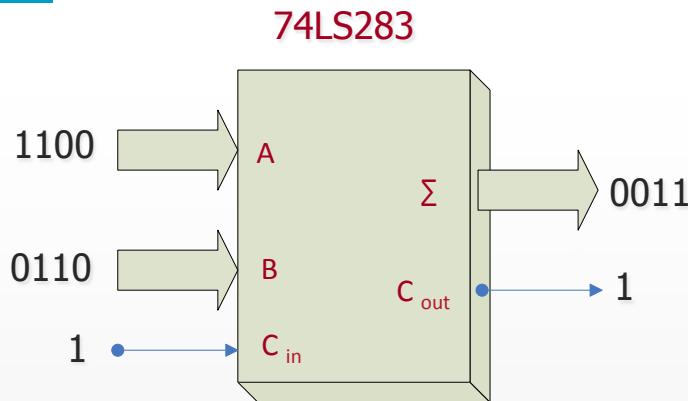


# Niveles de integración de los CI digitales

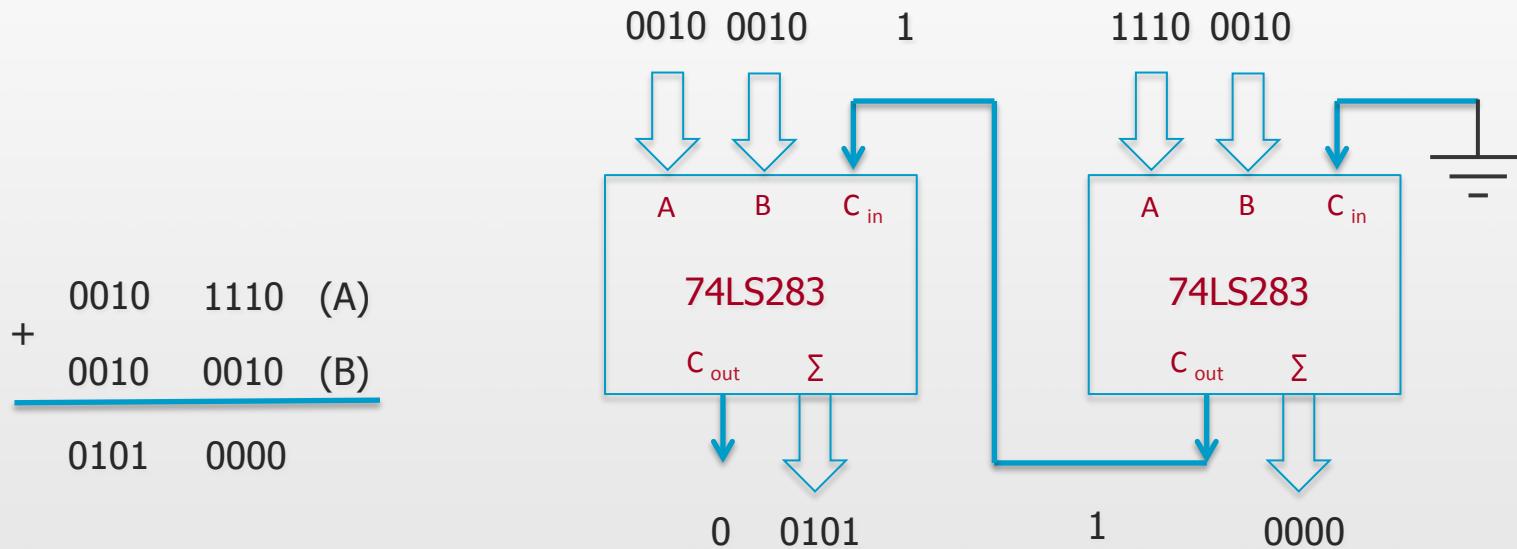
Nivel de integración	Nº de transistores	Nº de puertas	Fecha	Tipo
SSI (Small Scale Integration)	10 a 100	1 a 10	1.960	Puertas Flip-Flops
MSI (Medium Scale Integration)	100 a 1.000	10 a 100	1.965	Contadores Multiplexores Decodificadores Etc.
LSI (Large Scale Integration)	1.000 a 10.000	100 a 1.000	1.970	Microprocesadores Memorias
VLSI (Very Large Scale Integration)	10.000 a 100.000	1.000 a 10.000	1.978	Dispositivos programables ASIC
ULSI (Ultra Large Scale Integration)	100.000 a 1.000.000	10.000 a 100.000	1.985	
GLSI (Giga Large Scale Integration)	> 1.000.000	> 100.000	1.995	

# Operaciones aritméticas

## Sumadores

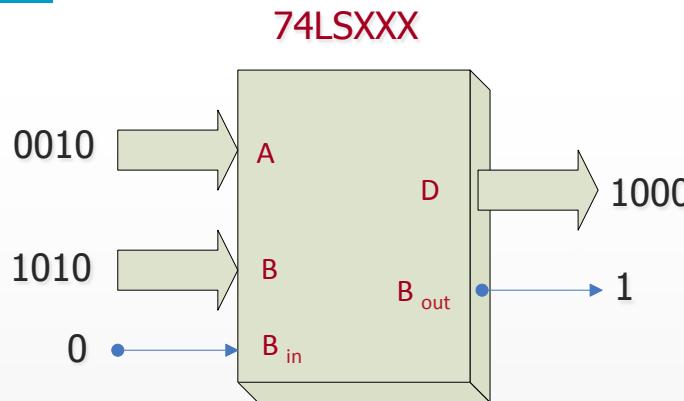


A	B	C <sub>in</sub>	C <sub>out</sub>	Σ
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

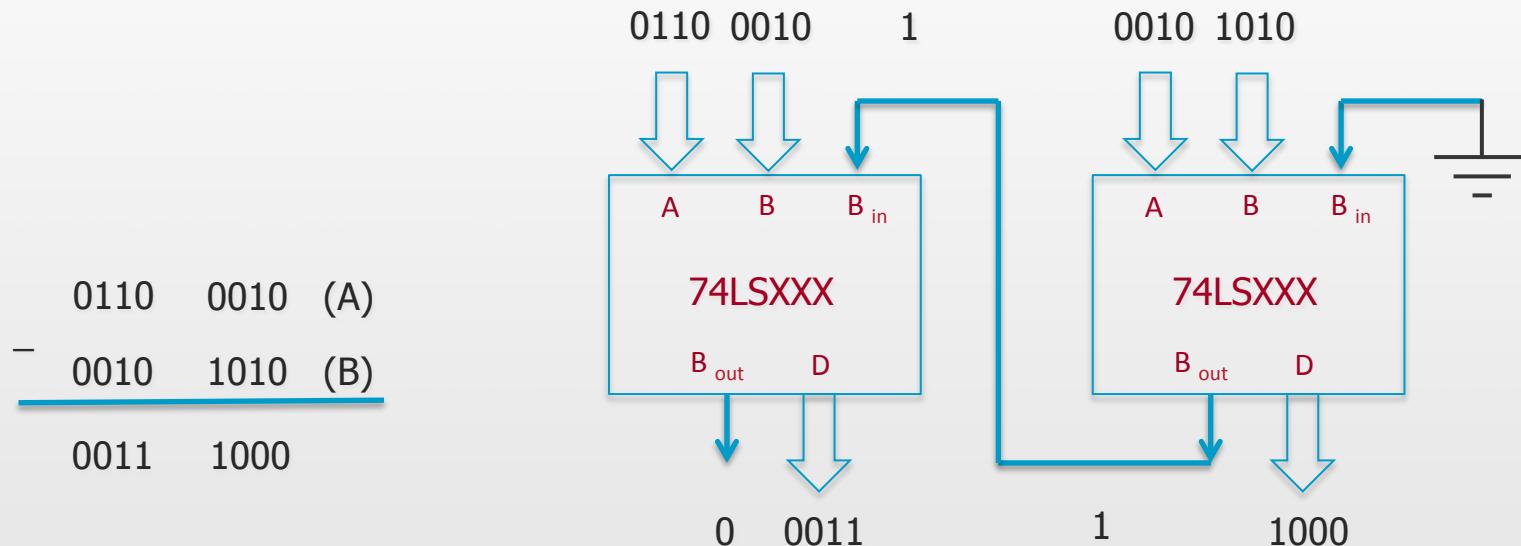


# Operaciones aritméticas

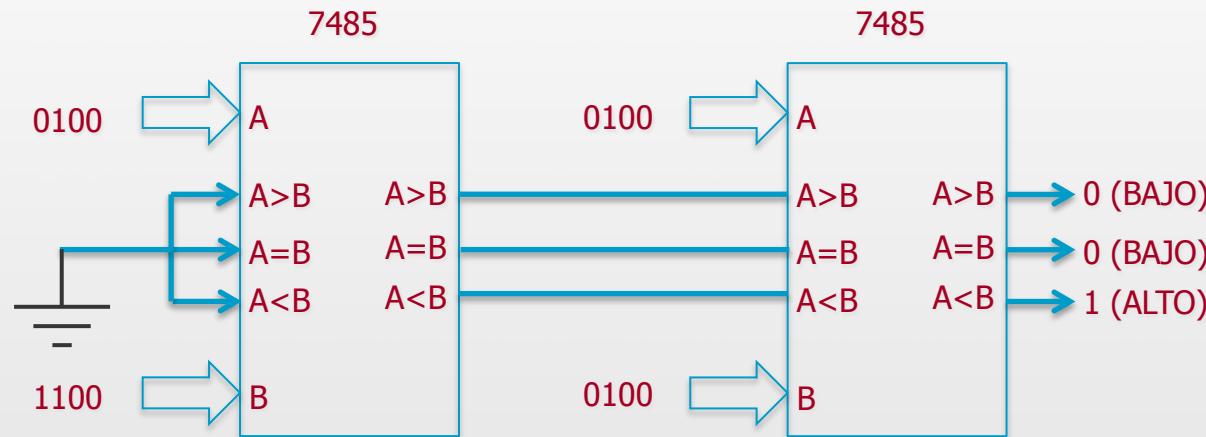
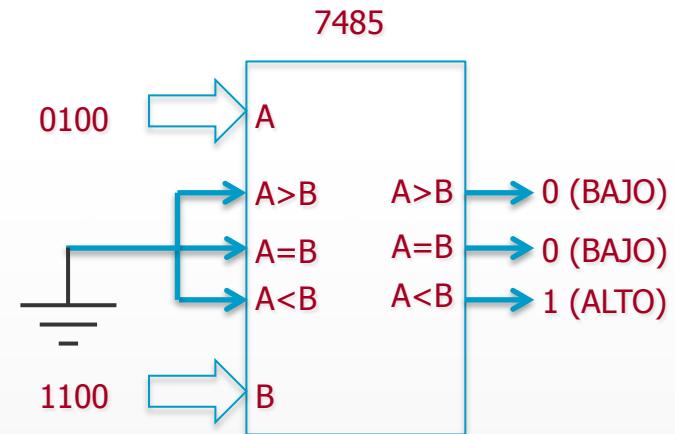
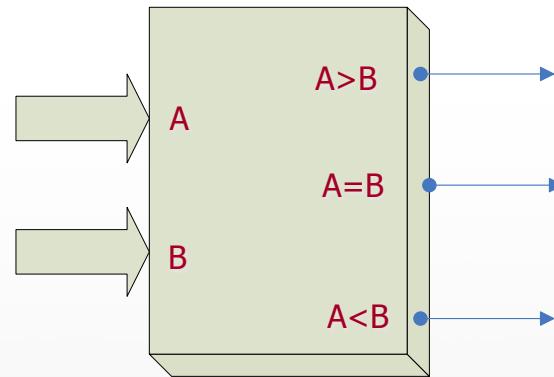
## Restadores



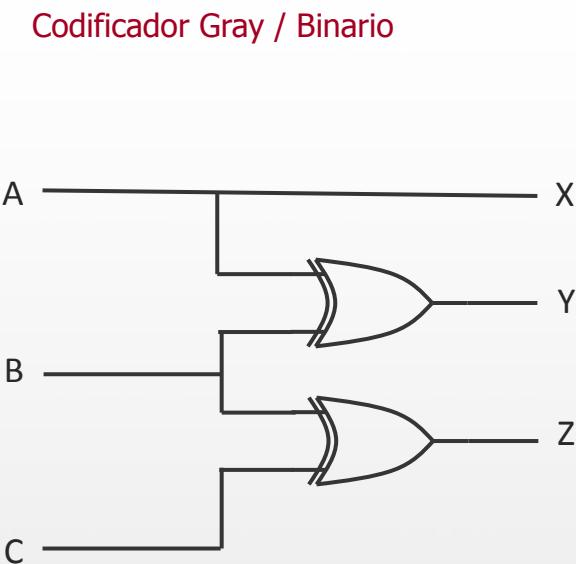
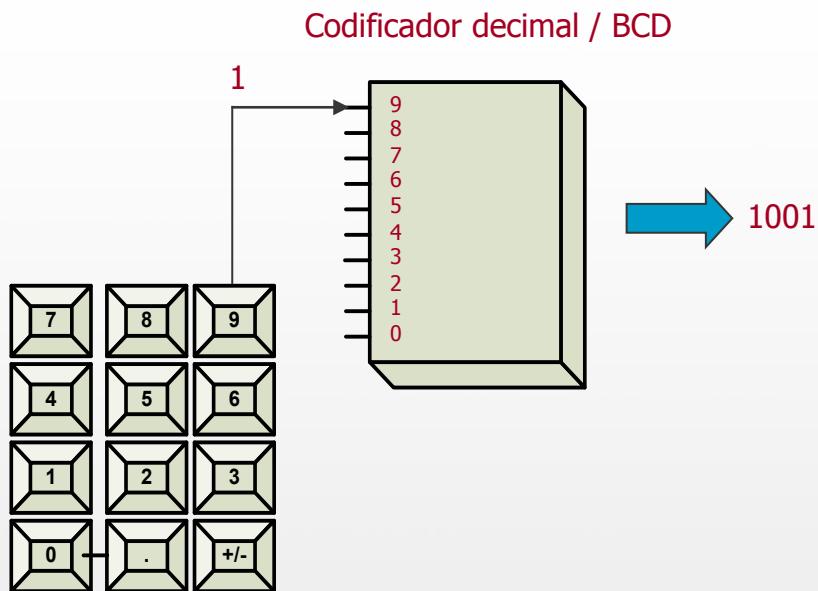
A	B	B <sub>in</sub>	B <sub>out</sub>	D
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	1	0
1	0	0	0	1
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1



# Comparadores

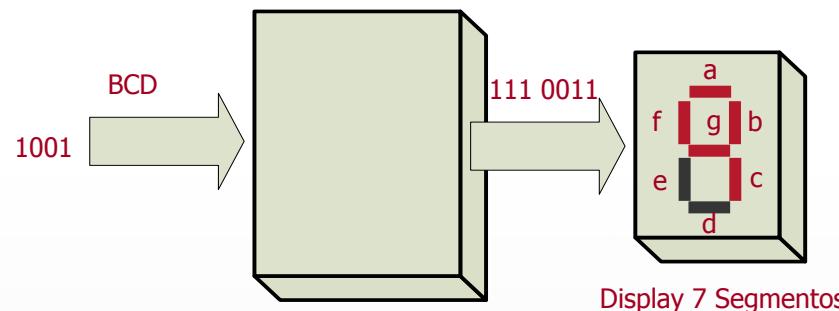


# Codificadores

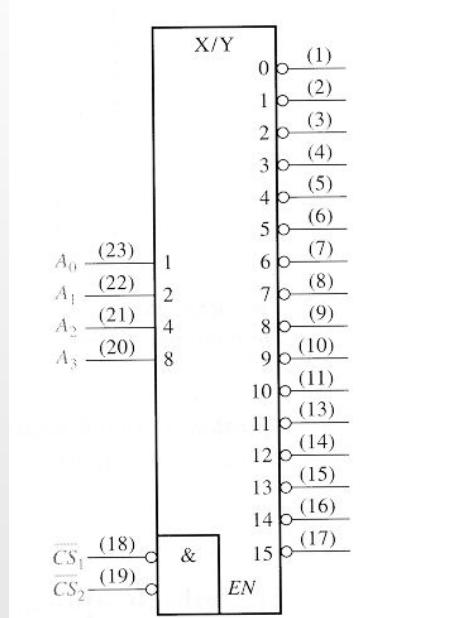


# Decodificadores

## Decodificador BCD a 7 Segmentos



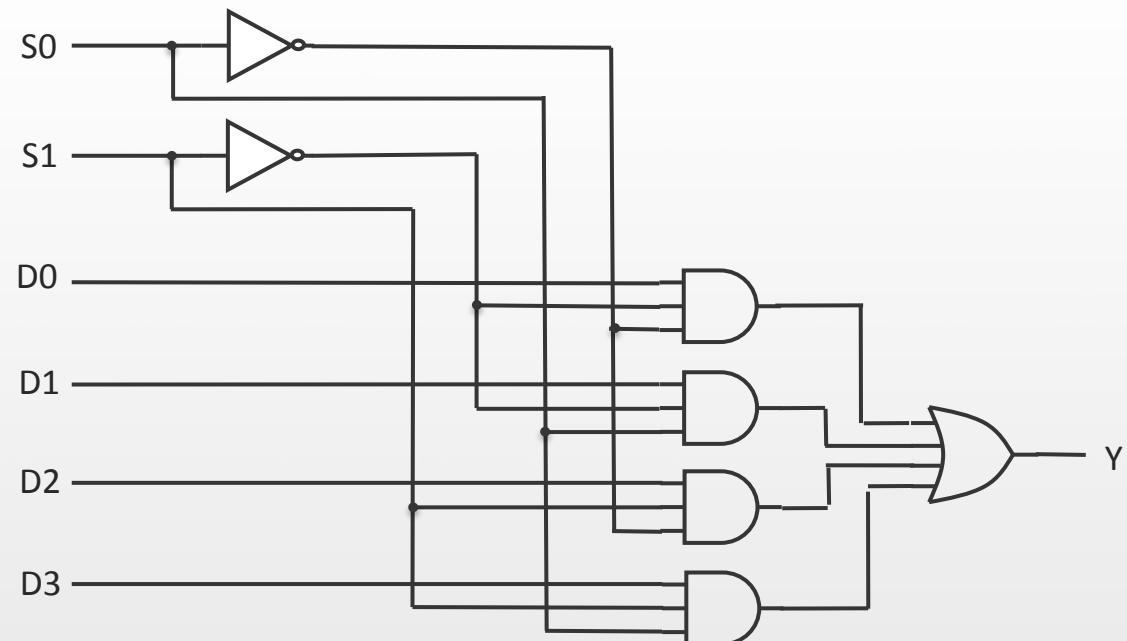
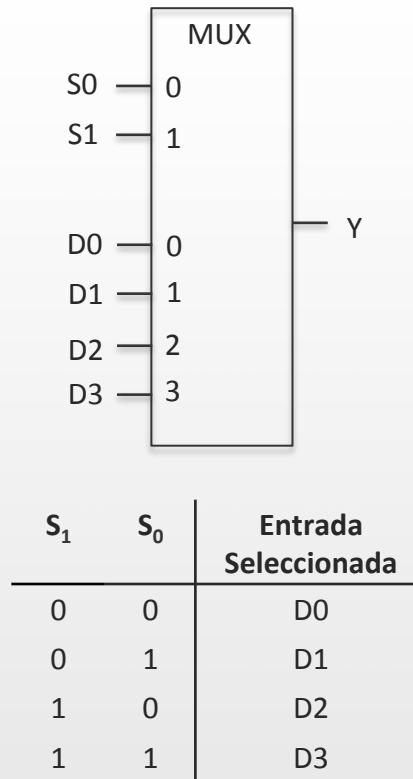
## Decodificador de cuatro bits (74154)



Dígito decimal	Entradas binarias				Función de decodificación	Salidas														
	$A_3$	$A_2$	$A_1$	$A_0$		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
0	0	0	0	0	$\bar{A}_3\bar{A}_2\bar{A}_1\bar{A}_0$	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	0	0	0	1	$\bar{A}_3\bar{A}_2\bar{A}_1A_0$	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1
2	0	0	1	0	$\bar{A}_3\bar{A}_2A_1\bar{A}_0$	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1
3	0	0	1	1	$\bar{A}_3\bar{A}_2A_1A_0$	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1
4	0	1	0	0	$\bar{A}_3A_2\bar{A}_1\bar{A}_0$	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1
5	0	1	0	1	$\bar{A}_3A_2\bar{A}_1A_0$	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1
6	0	1	1	0	$\bar{A}_3A_2A_1\bar{A}_0$	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1
7	0	1	1	1	$\bar{A}_3A_2A_1A_0$	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1
8	1	0	0	0	$A_3\bar{A}_2\bar{A}_1\bar{A}_0$	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1
9	1	0	0	1	$A_3\bar{A}_2\bar{A}_1A_0$	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1
10	1	0	1	0	$A_3\bar{A}_2A_1\bar{A}_0$	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1
11	1	0	1	1	$A_3\bar{A}_2A_1A_0$	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1
12	1	1	0	0	$A_3A_2\bar{A}_1\bar{A}_0$	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1
13	1	1	0	1	$A_3A_2\bar{A}_1A_0$	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1
14	1	1	1	0	$A_3A_2A_1\bar{A}_0$	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1
15	1	1	1	1	$A_3A_2A_1A_0$	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0

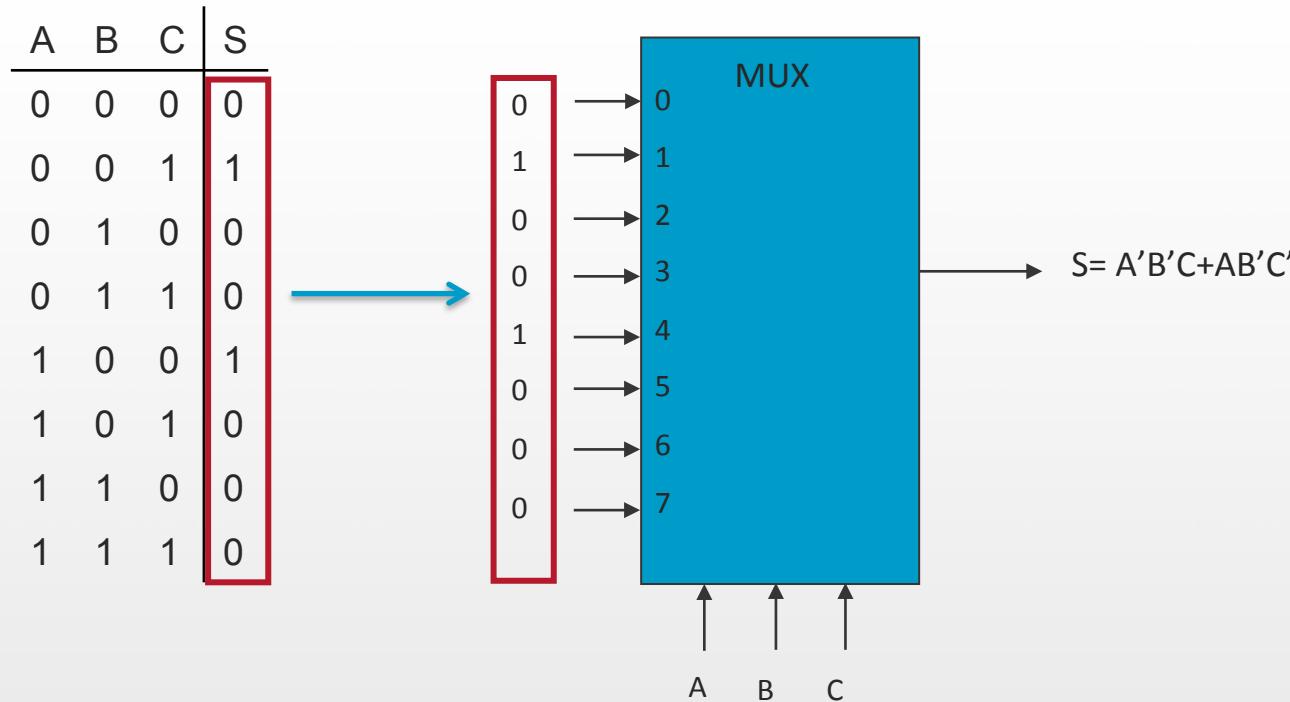
# Multiplexores (I)

La función básica de un **multiplexor** es dirigir la información digital procedente de diversas fuentes a una única línea para ser transmitida a través de dicha línea a un destino común.



## Multiplexores (II)

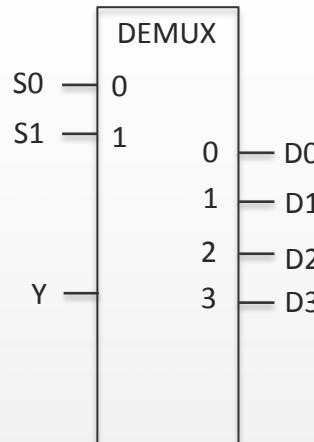
Los **multiplexores** nos permiten construir cualquier función combinacional. Para ello, tan solo hay que fijar las entradas del multiplexor a los niveles lógicos de la tabla de verdad que se quiere reproducir.



Una función combinacional de **n variables** requiere un MUX de  $2^n$  canales de entrada

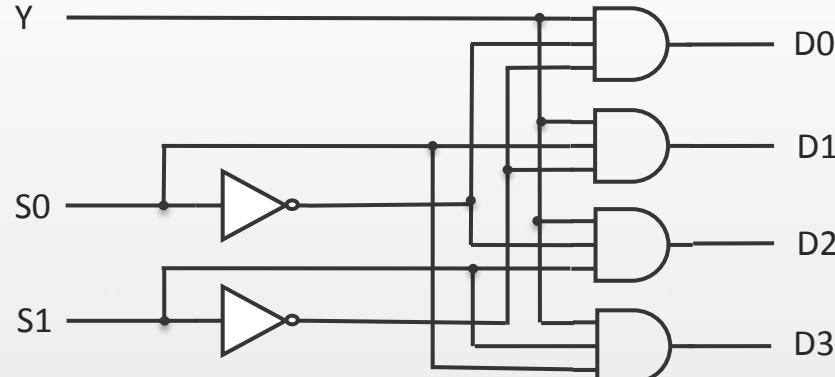
# Demultiplexores

La función básica de un **demultiplexor** es realizar la función contraria al multiplexor. Toma datos de una línea y los distribuye a un determinado número de líneas de salida.



$S_1$	$S_0$	Salida Seleccionada
0	0	D0
0	1	D1
1	0	D2
1	1	D3

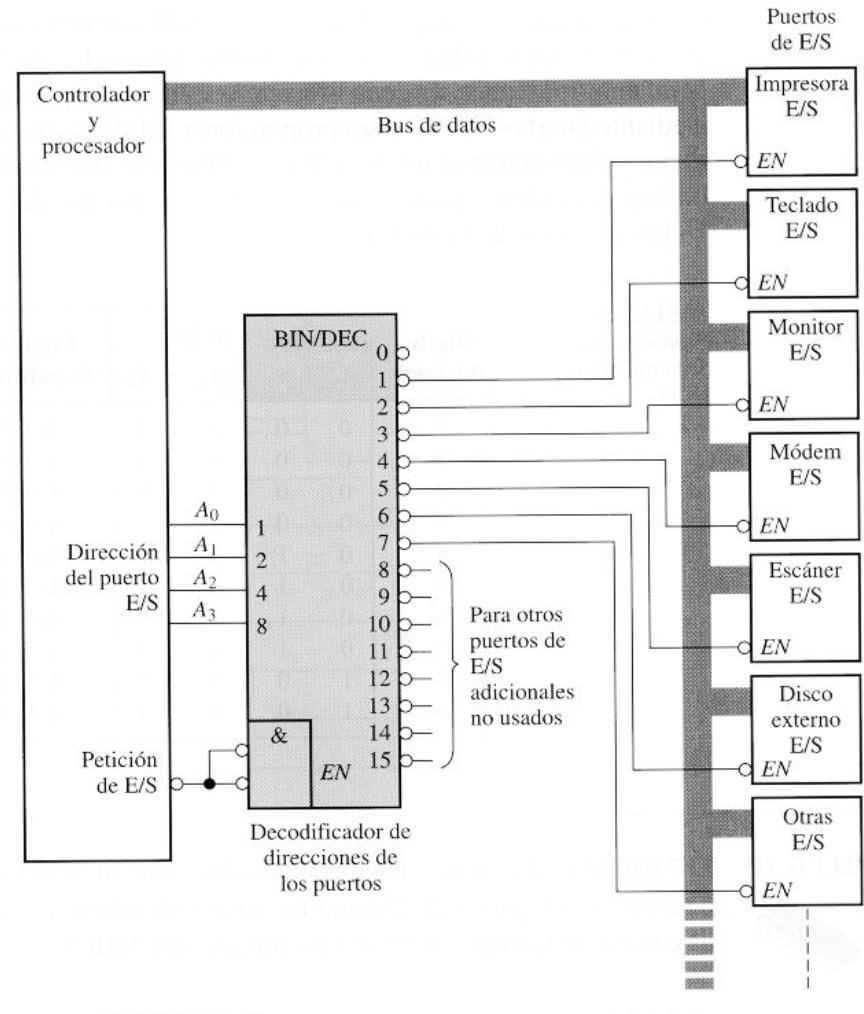
Entrada de Datos



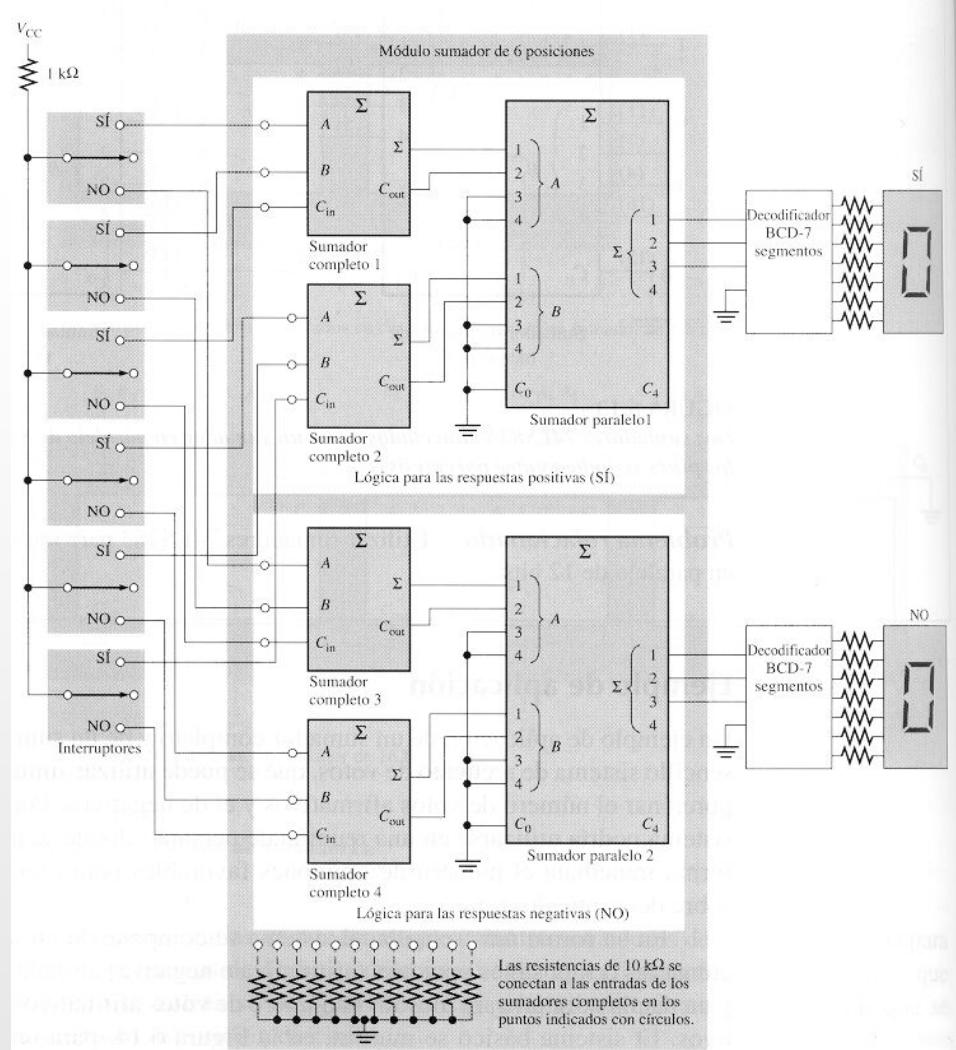
## Diseño de Sistemas Digitales con CI

- Los circuitos integrados se pueden usar como bloques para crear la funcionalidad deseada simplemente interconectándolos entre ellos.
- No existe una metodología predefinida, los diseños se realizan en base a la experiencia y las notas de aplicación que proporcionan los fabricantes.
- Es importante ver diseños ya existentes para aprender a diseñar nuevos.

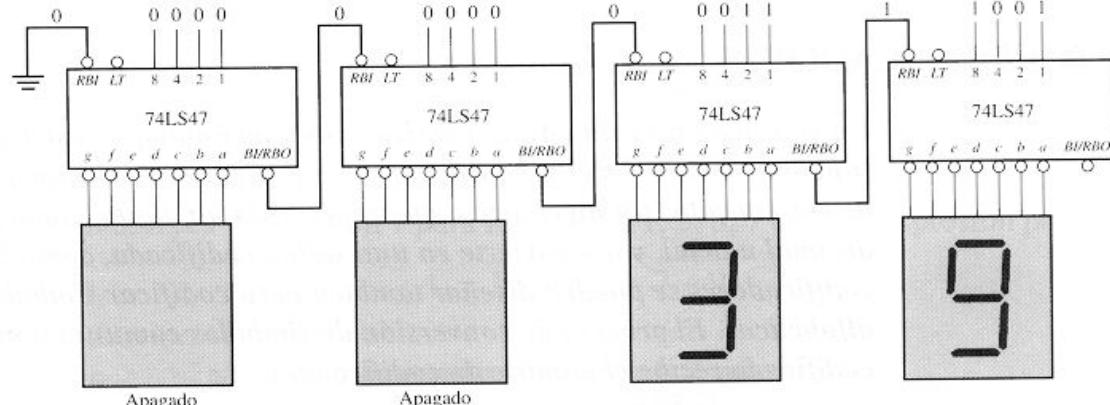
# Habilitación de puertos de E/S



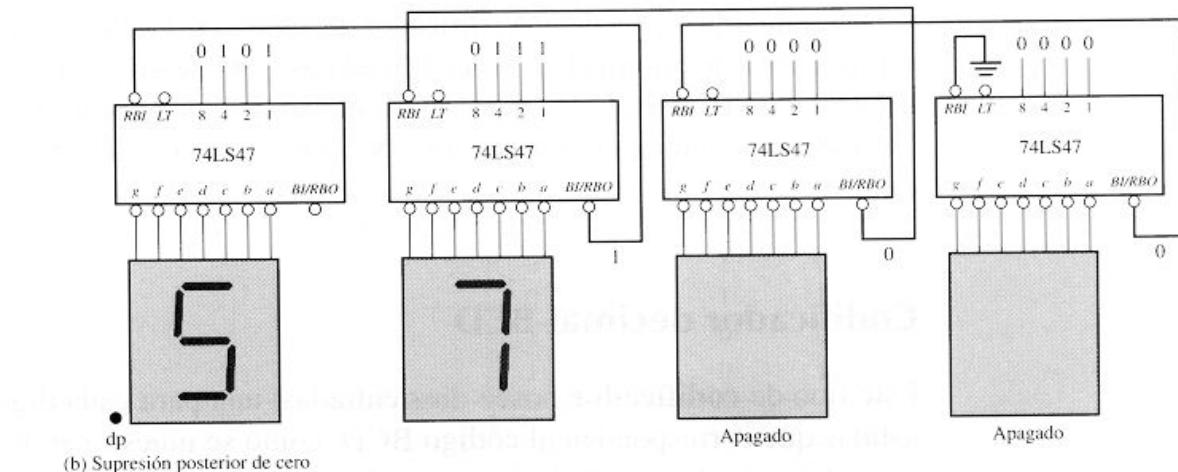
# Sistema de recuento de votos



# Display digital

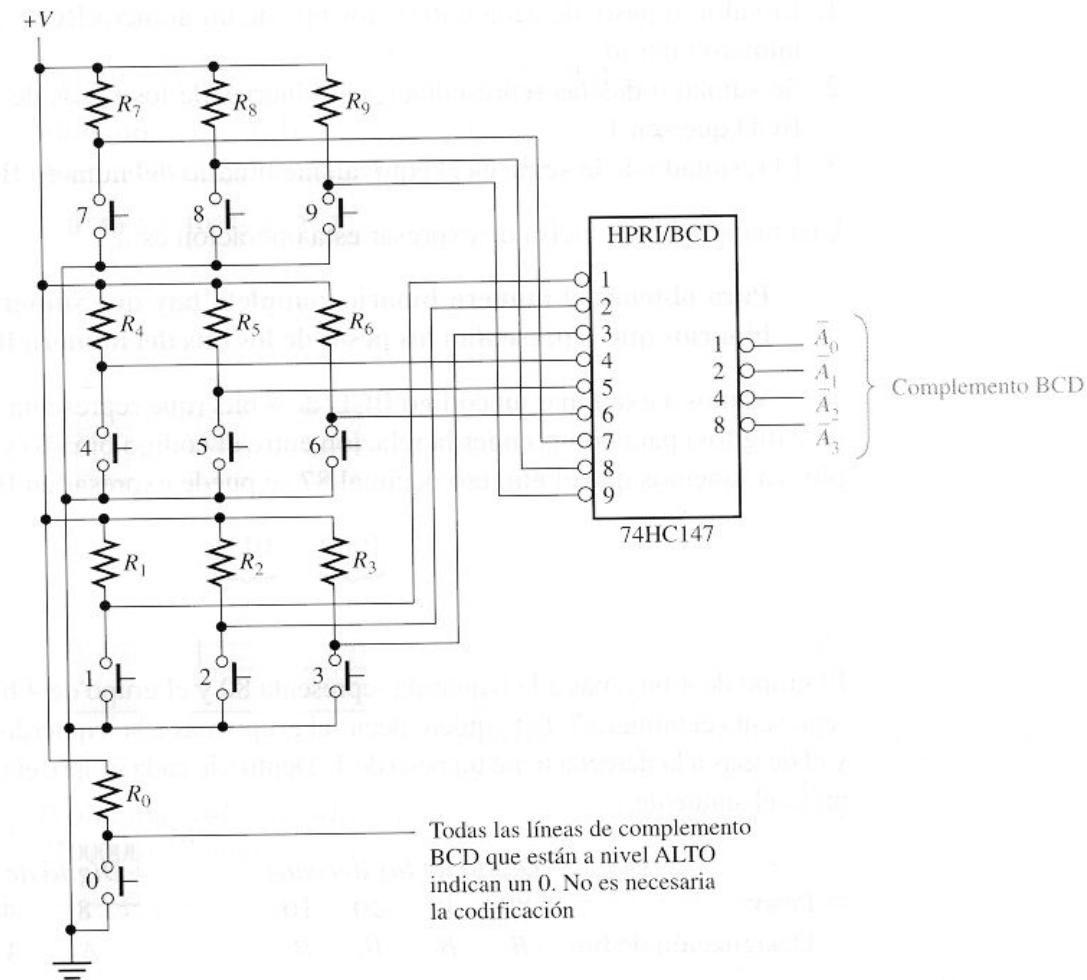


(a) Supresión anterior de cero

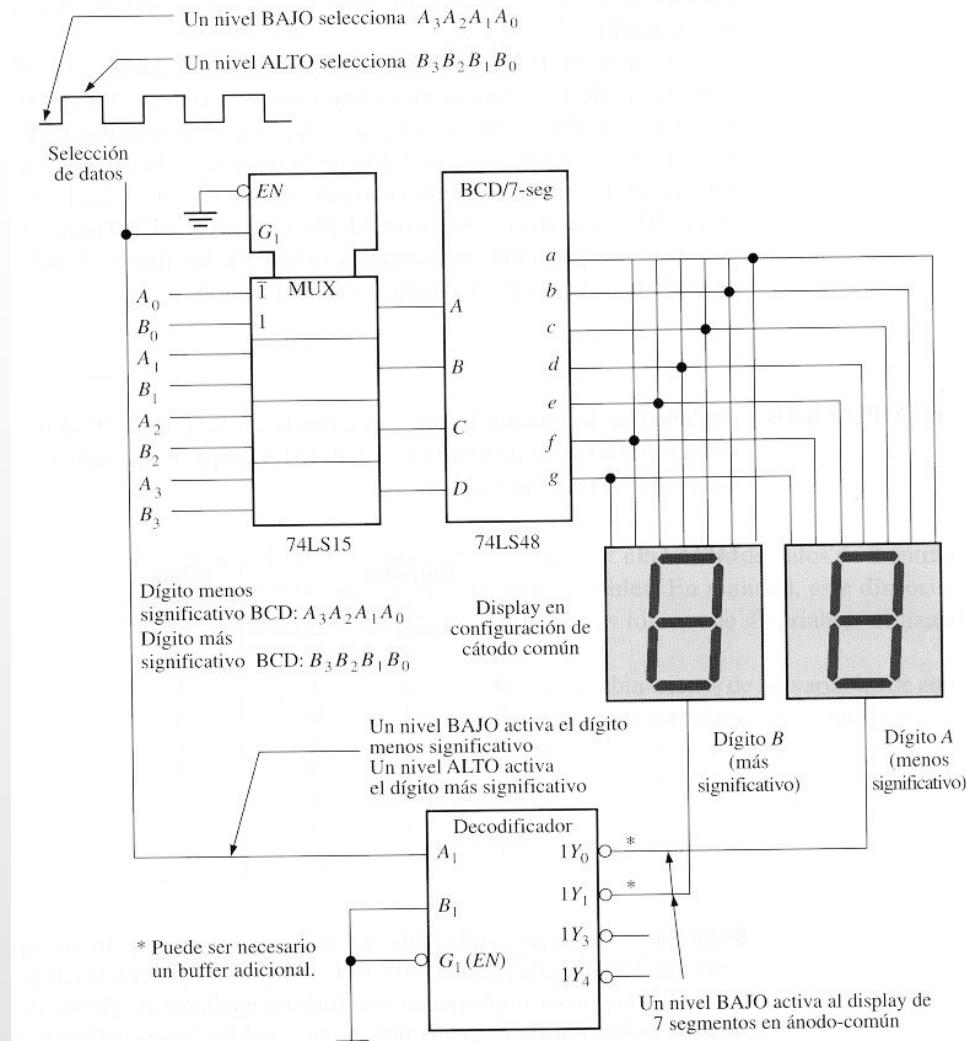


(b) Supresión posterior de cero

# Teclado digital



# Display multiplexado





Universidad  
Politécnica  
de Cartagena

Dr. Andrés Iborra  
Universidad Politécnica de Cartagena  
Campus Muralla del Mar, s/n  
30202 Cartagena

Tel.

+34 968 32 56 54

Fax.

+34 968 32 53 45

E-mail

[andres.iborra@upct.es](mailto:andres.iborra@upct.es)

Twitter

[@aiborra](https://twitter.com/aiborra)

Www

[www.aiborra.com](http://www.aiborra.com)



CLOUD  
INCUBATOR HUB