

TEMA 7

SISTEMAS DIGITALES

CUD

FUNDAMENTOS DE
ELECTRÓNICA

CURSO 2020-2021



Centro Universitario
de la Defensa Zaragoza

24 de septiembre de 2020

TEMA 7 – SISTEMAS DIGITALES

- Introducción. IFF
- Sistemas combinacionales
- Biestables
- Sistemas secuenciales

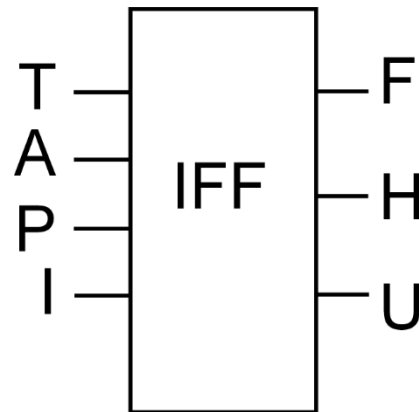
EJEMPLO DE SISTEMA DIGITAL. IFF

- Un sistema IFF es un sistema electrónico cuya misión es identificar aeronaves en vuelo mediante la emisión y recepción de un código.
- El sistema IFF se usa asociado a un sistema radar.
- Debido a las acciones de Inteligencia y de Guerra Electrónica desarrolladas por el enemigo no es un sistema totalmente fiable para ser empleado, por lo que debe complementarse con unos CRITERIOS DE HOSTILIDAD para poder calificar las aeronaves en nuestras inmediaciones como AMIGAS, ENEMIGAS O DESCONOCIDA.

EJEMPLO DE SISTEMA DIGITAL. IFF

- El sistema tendrá 4 entradas que indican si:
 - La aeronave ataca a una unidad propia (entrada: $T=1$)
 - La aeronave respeta la altura de vuelo (entrada: $A=0$)
 - La aeronave respeta los pasillos de vuelo (entrada: $P=0$)
 - La aeronave responde adecuadamente al IFF (entrada: $I=0$)
- El sistema tendrá 3 salidas que indican si la aeronave es:

- AMIGA (F)
- ENEMIGA (H)
- DESCONOCIDA (U)



EJEMPLO DE SISTEMA DIGITAL. IFF

Los criterios de hostilidad establecidos por el Mando para la operación en curso son los siguientes:

- Una aeronave será considerada AMIGA cuando:
 - Respete altura de vuelo, utilice los pasillos de vuelo, responda adecuadamente al IFF **y** por supuesto, no ataque a ninguna unidad propia.
- Una aeronave será considerada ENEMIGA cuando:
 - Ataque a una unidad propia **o**
 - No respete altura de vuelo, ni respete los pasillos de vuelo.
- Una aeronave será considerada DESCONOCIDA en el resto de los casos posibles

EJEMPLO DE SISTEMA DIGITAL. IFF

	T	A	P	I	F	H	U
0	0	0	0	0	1	0	0
1	0	0	0	1	0	0	1
2	0	0	1	0	0	0	1
3	0	0	1	1	0	0	1
4	0	1	0	0	0	0	1
5	0	1	0	1	0	0	1
6	0	1	1	0	0	1	0
7	0	1	1	1	0	1	0
8	1	0	0	0	0	1	0
9	1	0	0	1	0	1	0
10	1	0	1	0	0	1	0
11	1	0	1	1	0	1	0
12	1	1	0	0	0	1	0
13	1	1	0	1	0	1	0
14	1	1	1	0	0	1	0
15	1	1	1	1	0	1	0

EJEMPLO DE SISTEMA DIGITAL. IFF

TA \ PI	00	01	11	10
00	1	0	0	0
01	0	0	0	0
11	0	0	0	0
10	0	0	0	0

$$F = \bar{T}\bar{A}\bar{P}\bar{I}$$

TA \ PI	00	01	11	10
00	0	0	0	0
01	0	0	1	1
11	1	1	1	1
10	1	1	1	1

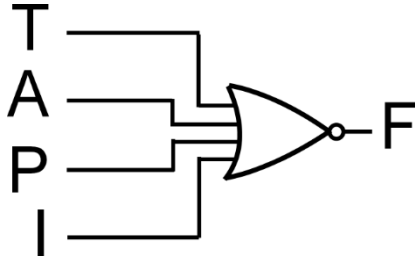
$$H = T + AP$$

TA \ PI	00	01	11	10
00	0	1	1	1
01	1	1	0	0
11	0	0	0	0
10	0	0	0	0

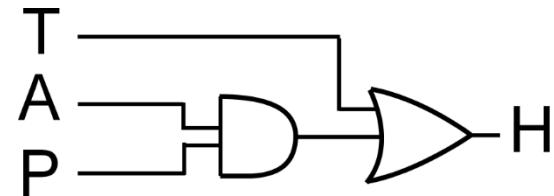
$$U = \bar{T}(\bar{A} + \bar{P})(A + P + I)$$

EJEMPLO DE SISTEMA DIGITAL. IFF

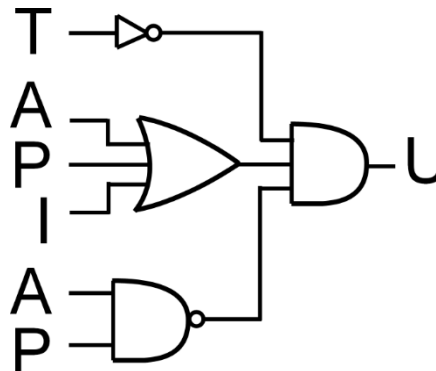
$$F = \bar{T}\bar{A}\bar{P}\bar{I} = \overline{T + A + P + I}$$



$$H = T + AP$$



$$U = \bar{T}(A + P + I)(\bar{A} + \bar{P}) = \bar{T}(A + P + I)\bar{A}\bar{P}$$



[Link simulador Falstad](#)

TEMA 7 – SISTEMAS DIGITALES

- Introducción. IFF
- Sistemas combinacionales
- Biestables
- Sistemas secuenciales

SISTEMAS COMBINACIONALES

- Basados en puertas lógicas
- La salida depende sólo del valor de las entradas
- Se definen mediante una tabla de verdad
- Se clasifican según su funcionalidad:
 - Comparadores
 - Sumadores (restadores)
 - Codificadores
 - Decodificadores
 - Conversores de Código
 - Multiplexores
 - Demultiplexores
 - Memorias ROM

COMPARADORES

- Es un circuito que permite determinar si dos datos son iguales, o si uno es mayor que otro.
- Para hacer la comparación más sencilla (dos bits) no basta con una salida. Se suelen usar tres, de las cuales solo una está activa.



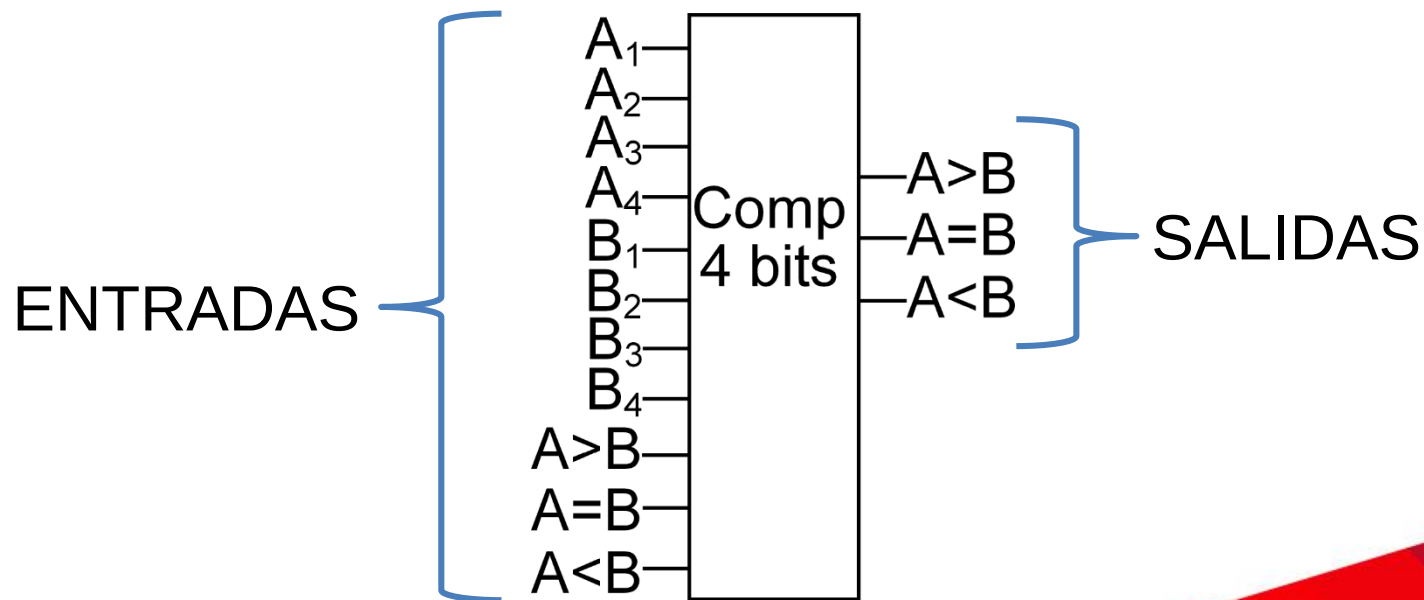
[Link simulador Falstad](#)

a	b	a>b	a=b	a<b
0	0	0	1	0
0	1	0	0	1
1	0	1	0	0
1	1	0	1	0

COMPARADORES

- El integrado comercial es el 7485. Compara dos números de 4 bits y permite conexión en cascada.

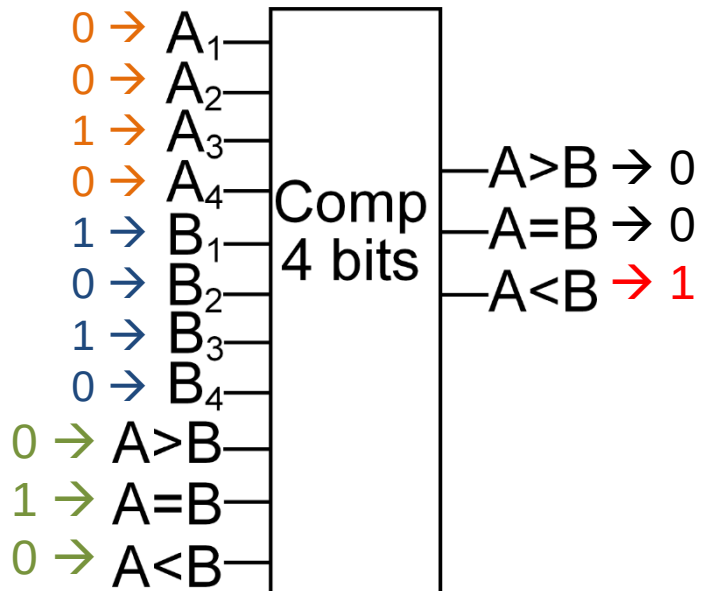
Números que se comparan $\left\{ \begin{array}{l} A \rightarrow A_4A_3A_2A_1 \\ B \rightarrow B_4B_3B_2B_1 \end{array} \right.$



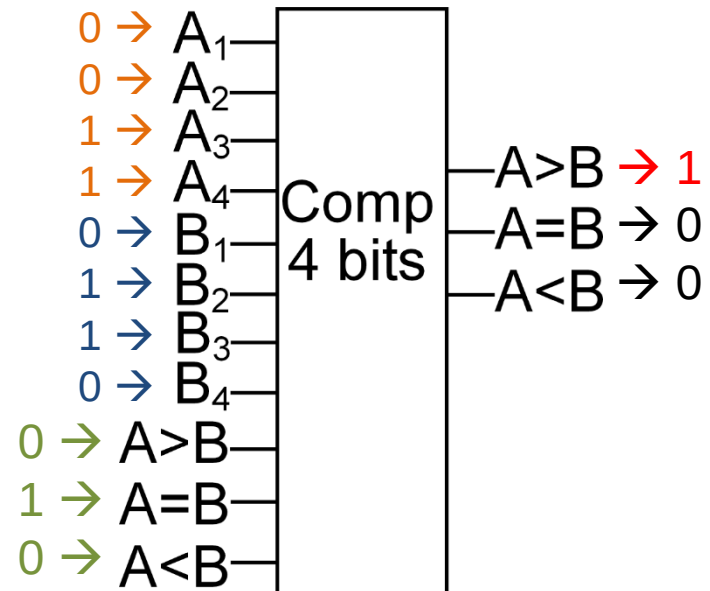
COMPARADORES

➤ Ejemplos: $A \rightarrow A_4A_3A_2A_1$
 $B \rightarrow B_4B_3B_2B_1$

$A \rightarrow 0100$
 $B \rightarrow 0101$



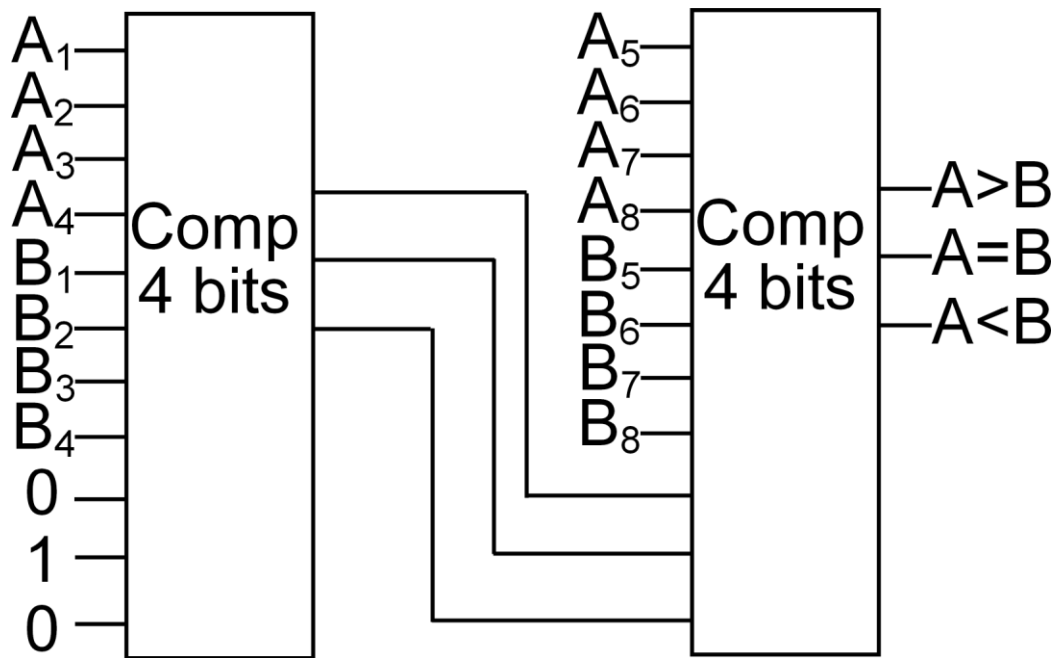
$A \rightarrow 1100$
 $B \rightarrow 0110$



CONEXIÓN EN CASCADA

- Comparador de 8 bits = 2 comparadores de 4 bits en cascada

$$\begin{array}{cccccccc} A_8 & A_7 & A_6 & A_5 & A_4 & A_3 & A_2 & A_1 \\ B_8 & B_7 & B_6 & B_5 & B_4 & B_3 & B_2 & B_1 \end{array}$$

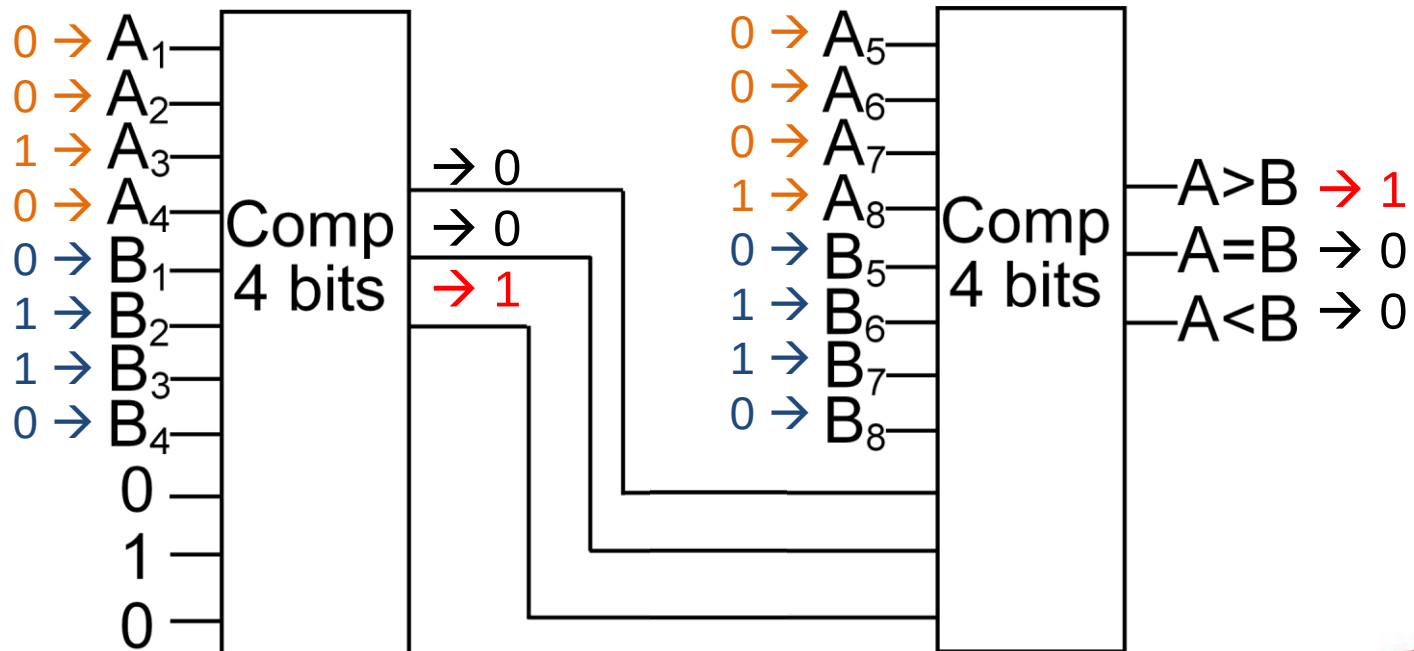


CONEXIÓN EN CASCADA

➤ Ejemplo:

$A \rightarrow A_8 A_7 A_6 A_5 A_4 A_3 A_2 A_1 \rightarrow 10000100$

$B \rightarrow B_8 B_7 B_6 B_5 B_4 B_3 B_2 B_1 \rightarrow 01100110$

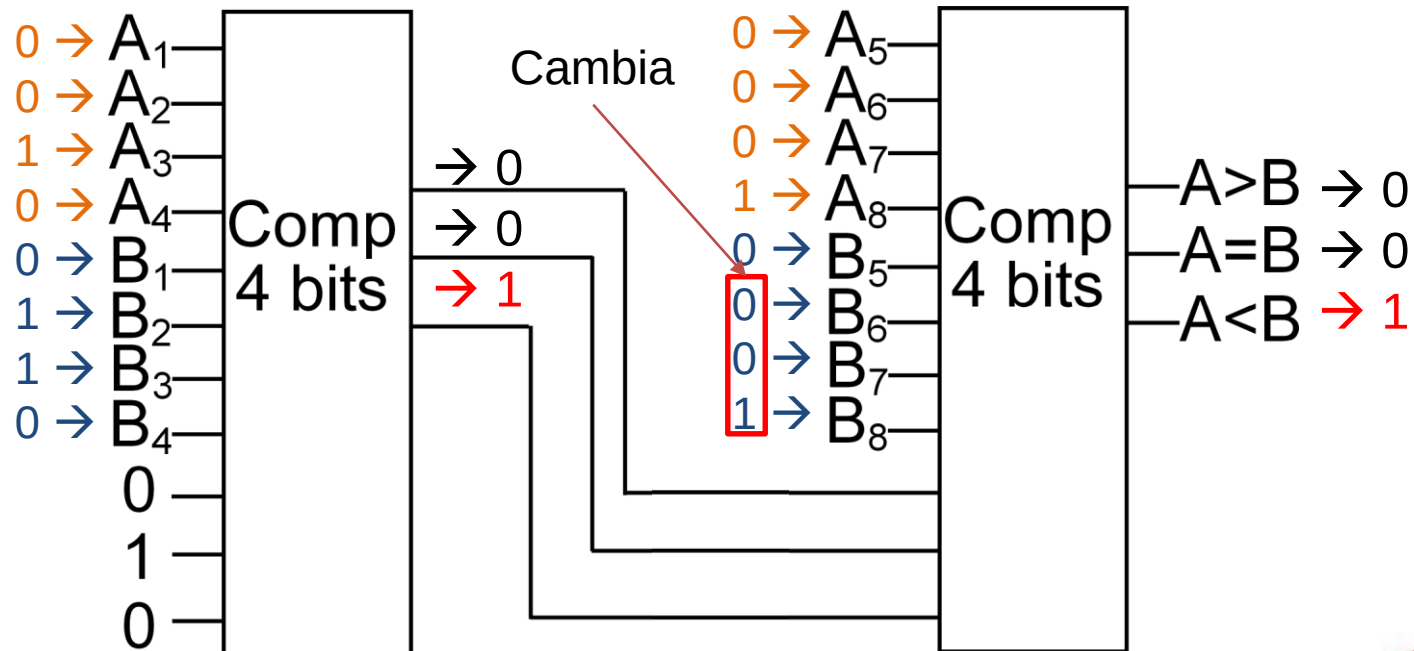


CONEXIÓN EN CASCADA

➤ Ejemplo:

$A \rightarrow A_8 A_7 A_6 A_5 A_4 A_3 A_2 A_1 \rightarrow 10000100$

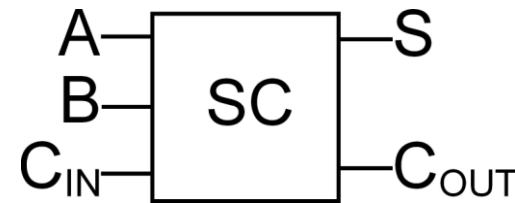
$B \rightarrow B_8 B_7 B_6 B_5 B_4 B_3 B_2 B_1 \rightarrow 10000110$



SUMADOR COMPLETO (full adder)

- El sumador completo tiene en cuenta el posible acarreo (C_{IN})

A	B	C_{IN}	S	C_{OUT}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

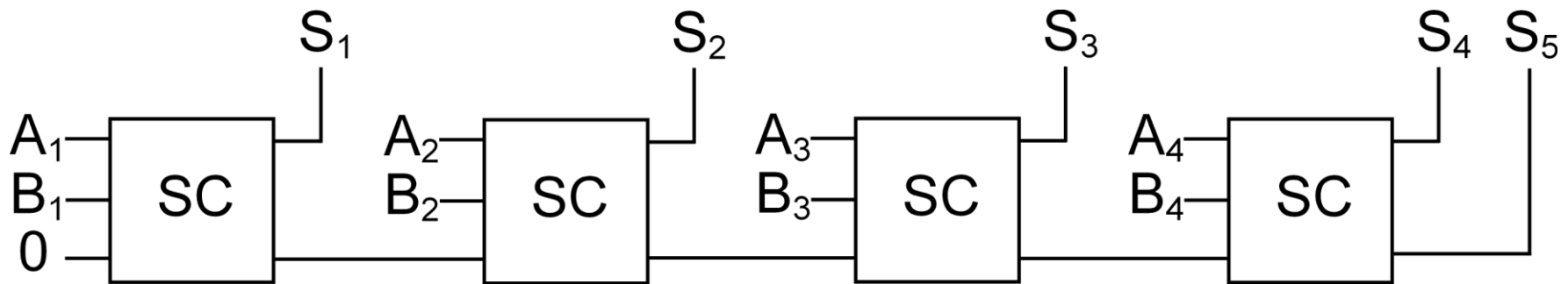


[Link simulador Falstad](#)

SUMADOR DE N BITS

- Para formar un sumador de n bits, unimos varios sumadores completos en cascada. Cada uno tiene en cuenta el acarreo del anterior.
- Ejemplo: 4 bits

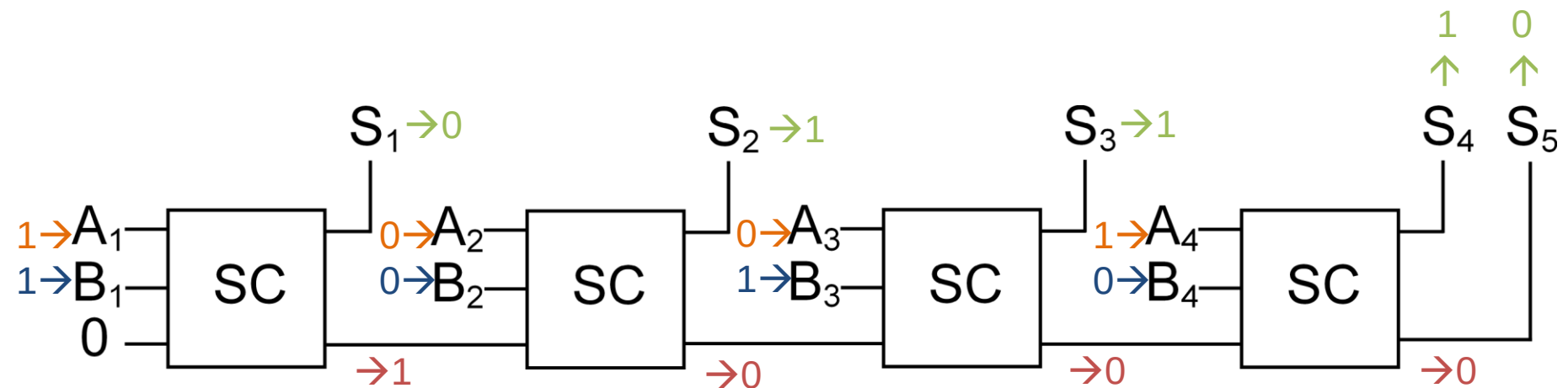
$$\begin{array}{r} A_4 A_3 A_2 A_1 \\ + B_4 B_3 B_2 B_1 \\ \hline S_5 S_4 S_3 S_2 S_1 \end{array}$$



SUMADOR DE N BITS

➤ Ejemplo:

$$\begin{array}{r} A_4 A_3 A_2 A_1 \\ + B_4 B_3 B_2 B_1 \\ \hline S_5 S_4 S_3 S_2 S_1 \end{array} \quad \begin{array}{r} 1 \\ 1001 \\ + 0101 \\ \hline 01110 \end{array}$$



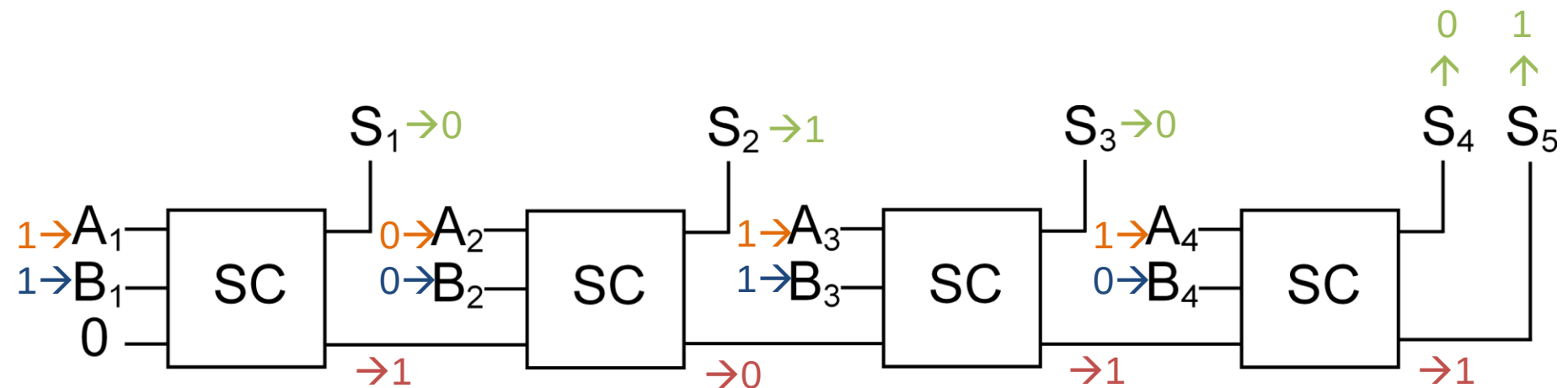
SUMADOR DE N BITS

➤ Ejemplo:

$$\begin{array}{r}
 A_4 A_3 A_2 A_1 \\
 + B_4 B_3 B_2 B_1 \\
 \hline
 S_5 S_4 S_3 S_2 S_1
 \end{array}$$

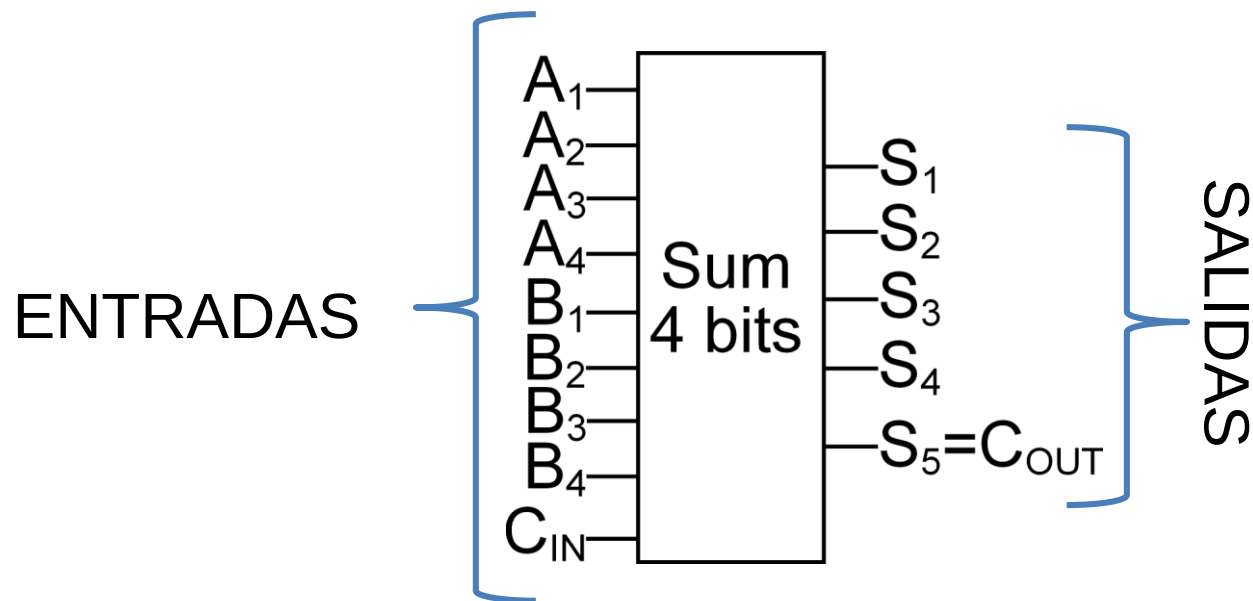
Cambia

$$\begin{array}{r}
 1\ 1\ 1 \\
 1\ 1\ 0\ 1 \\
 + 0\ 1\ 0\ 1 \\
 \hline
 1\ 0\ 0\ 1\ 0
 \end{array}$$



SUMADOR DE 4 BITS

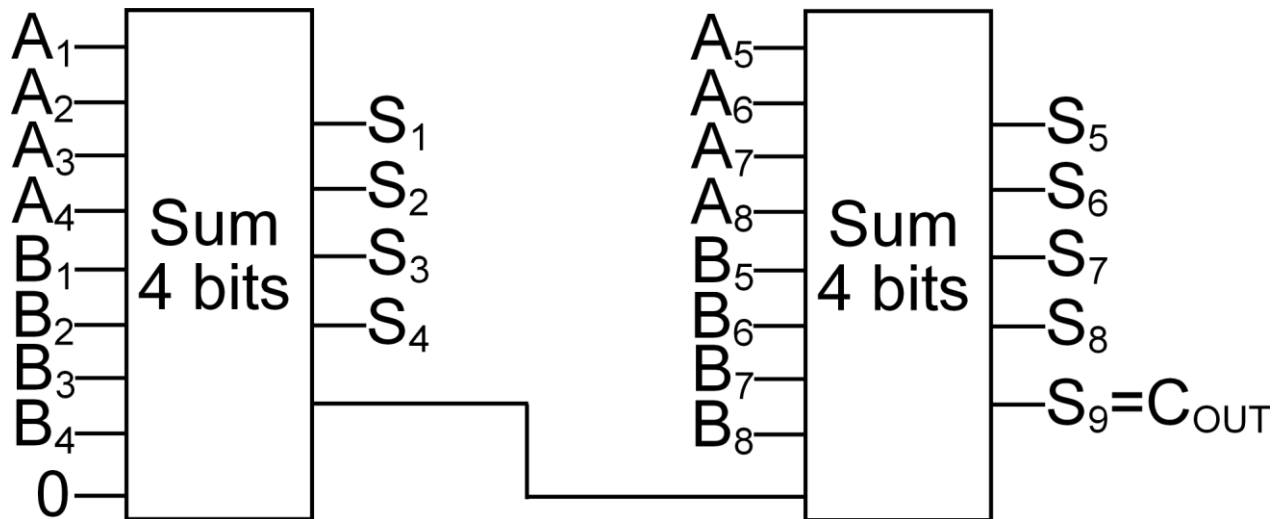
- El integrado comercial 74283 es un sumador completo de 4 bits, permitiendo conexión en cascada:



CONEXIÓN EN CASCADA

- 2 sumadores de 4 bit permiten implementar un sumador de 8 bits

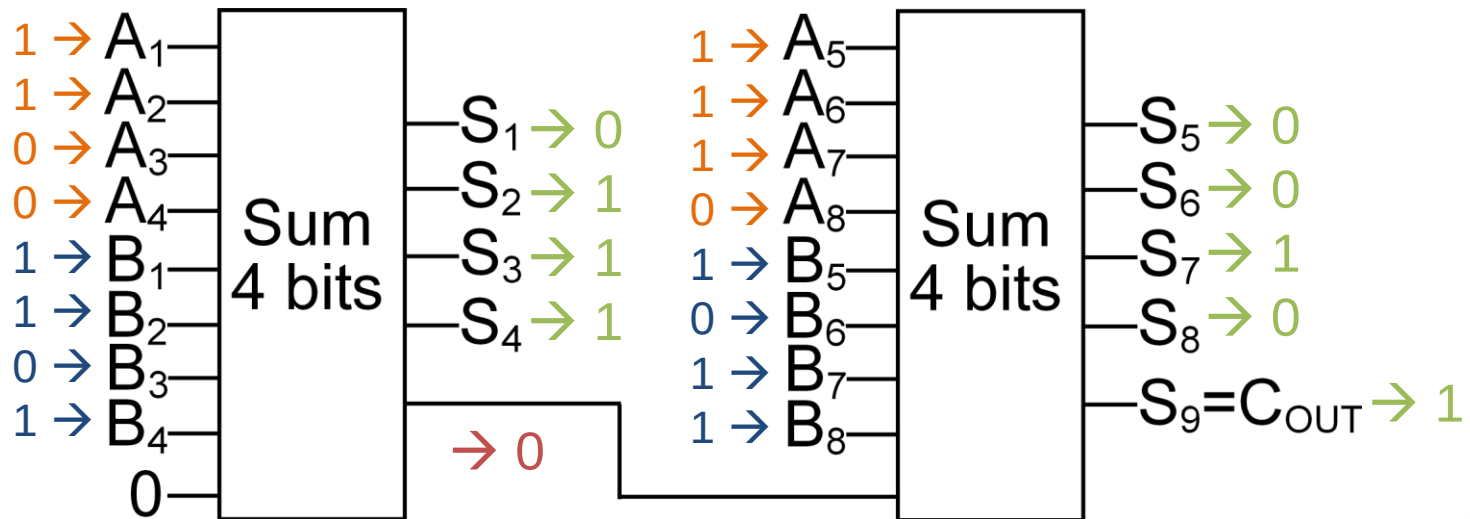
$$\begin{array}{r} A_8 A_7 A_6 A_5 A_4 A_3 A_2 A_1 \\ + B_8 B_7 B_6 B_5 B_4 B_3 B_2 B_1 \\ \hline S_9 S_8 S_7 S_6 S_5 S_4 S_3 S_2 S_1 \end{array}$$



CONEXIÓN EN CASCADA

➤ Ejemplo

$$\begin{array}{r} A_8 A_7 A_6 A_5 A_4 A_3 A_2 A_1 \\ + B_8 B_7 B_6 B_5 B_4 B_3 B_2 B_1 \\ \hline S_9 S_8 S_7 S_6 S_5 S_4 S_3 S_2 S_1 \end{array} \quad + \quad \begin{array}{r} 01110011 \\ 11011011 \\ \hline 101001110 \end{array}$$



CONEXIÓN EN CASCADA

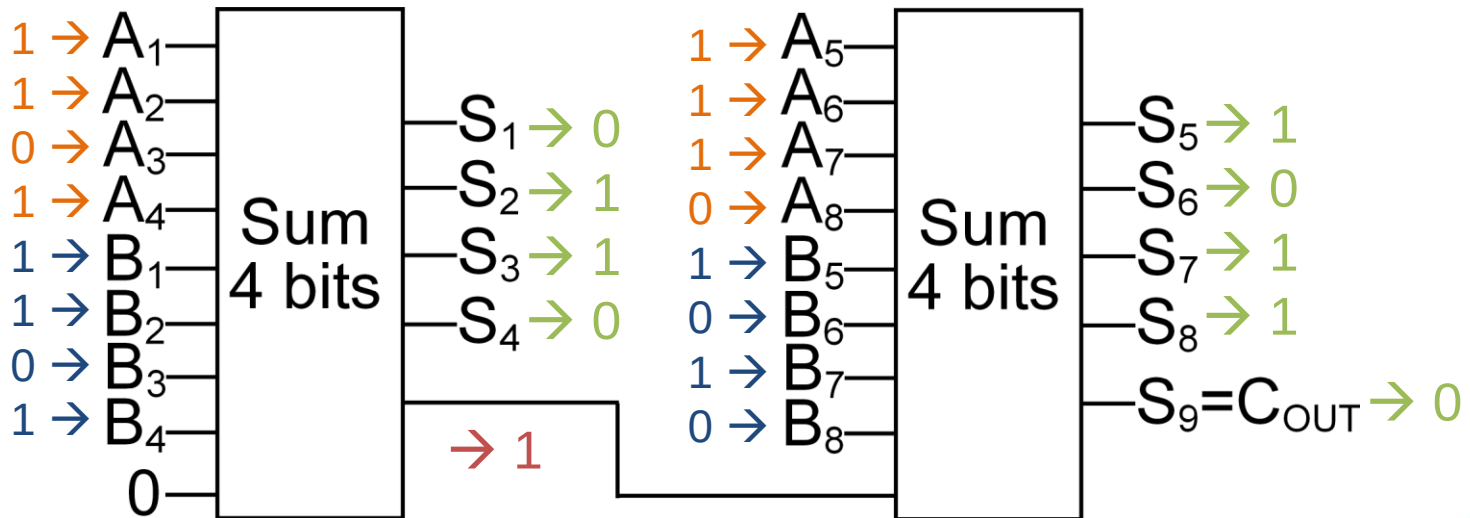
➤ Ejemplo

$$\begin{array}{r}
 A_8 A_7 A_6 A_5 A_4 A_3 A_2 A_1 \\
 + B_8 B_7 B_6 B_5 B_4 B_3 B_2 B_1 \\
 \hline
 S_9 S_8 S_7 S_6 S_5 S_4 S_3 S_2 S_1
 \end{array}$$

Cambia

$$\begin{array}{r}
 01111011 \\
 + 01011011 \\
 \hline
 011010110
 \end{array}$$

Cambia



SUMADOR/RESTADOR

- Para sumar y restar, trabajamos en complemento a 2. Restar dos números es equivalente a sumar el número cambiado de signo.

- $A - B = A + \text{Ca2}(B)$

- Recordemos que cambiar el signo en Ca2 consiste en complementar todos los bits y sumarle 1 al resultado. Según esto podemos diseñar un sumador/restador de 4 bits basándonos en un sumador de 4 bits y puertas XOR, ya que:

- Si $a = 0$:

$$y = a \oplus b = b$$

- Si $a = 1$:

$$y = a \oplus b = \bar{b}$$

Tabla de verdad
de la función XOR

a	b	y
0	0	0
0	1	1
1	0	1
1	1	0

Complementamos el valor de la variable b (o no) en función de la variable a

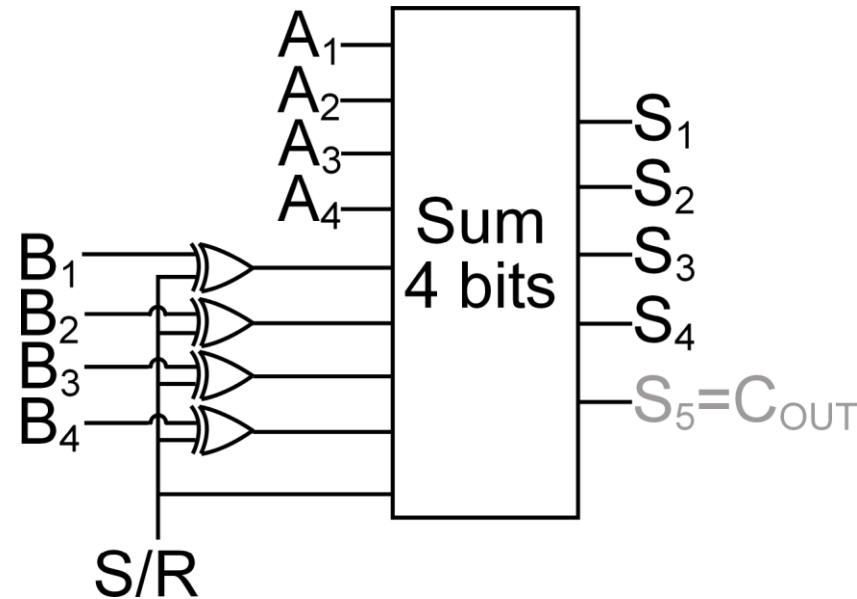
SUMADOR/RESTADOR

➤ Si $S/R = 0$

- Los valores de las entradas B_1 - B_4 no se complementan en las puertas XOR
- El C_{IN} es 0
- Por tanto, se realiza una suma

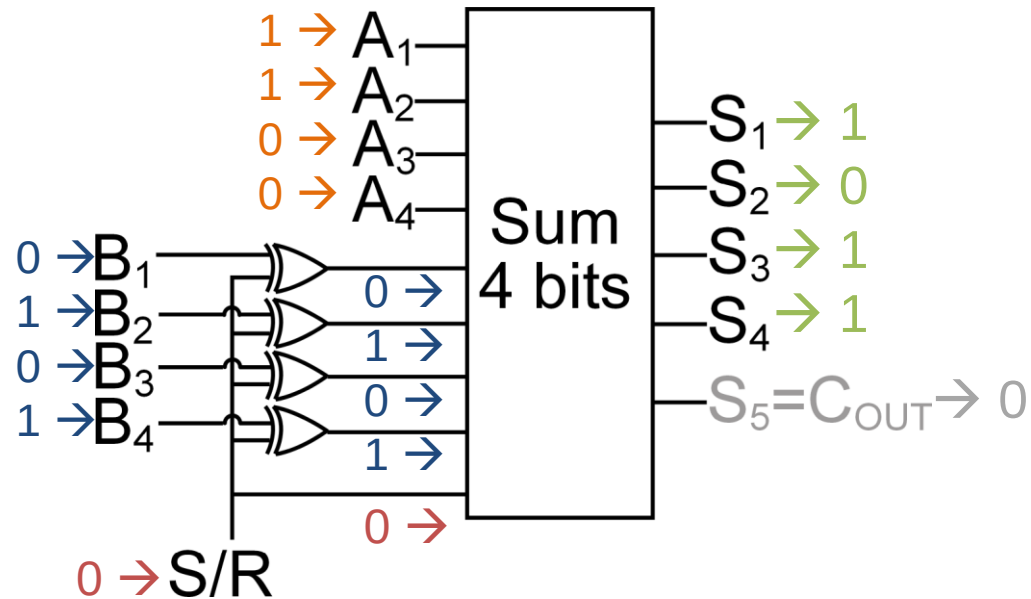
➤ Si $S/R = 1$

- Los valores de las entradas B_1 - B_4 sí se complementan en las puertas XOR
- El C_{IN} es 1
- Por tanto, se realiza una resta usando el $Ca2$



SUMADOR/RESTADOR

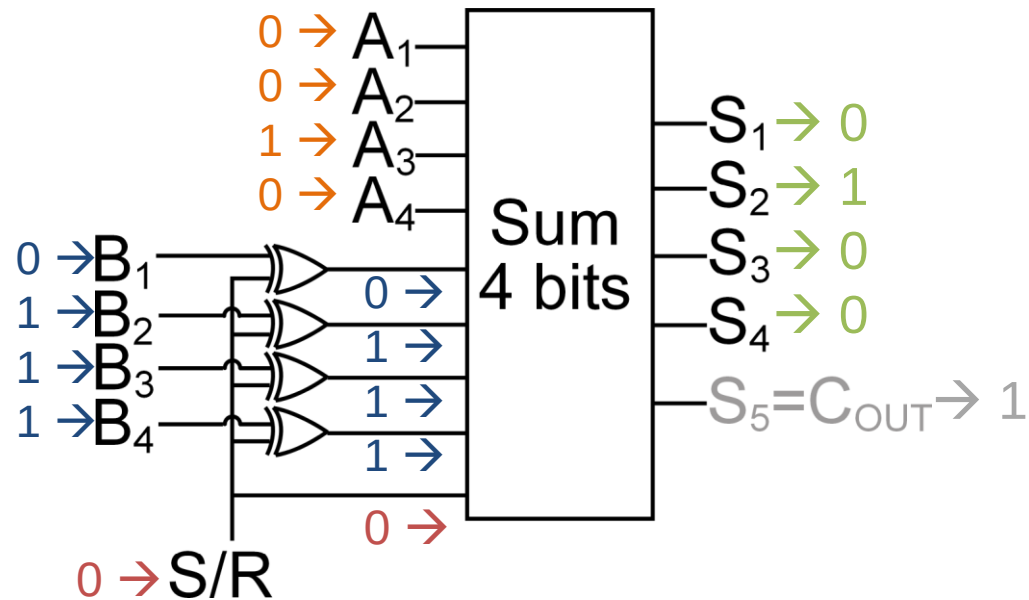
➤ Suma



$$\begin{array}{r}
 A_4 A_3 A_2 A_1 \\
 + \quad B_4 B_3 B_2 B_1 \\
 \hline
 S_5 S_4 S_3 S_2 S_1
 \end{array}
 \quad
 \begin{array}{r}
 0011 \\
 + 1010 \\
 \hline
 01101
 \end{array}$$

SUMADOR/RESTADOR

➤ Suma

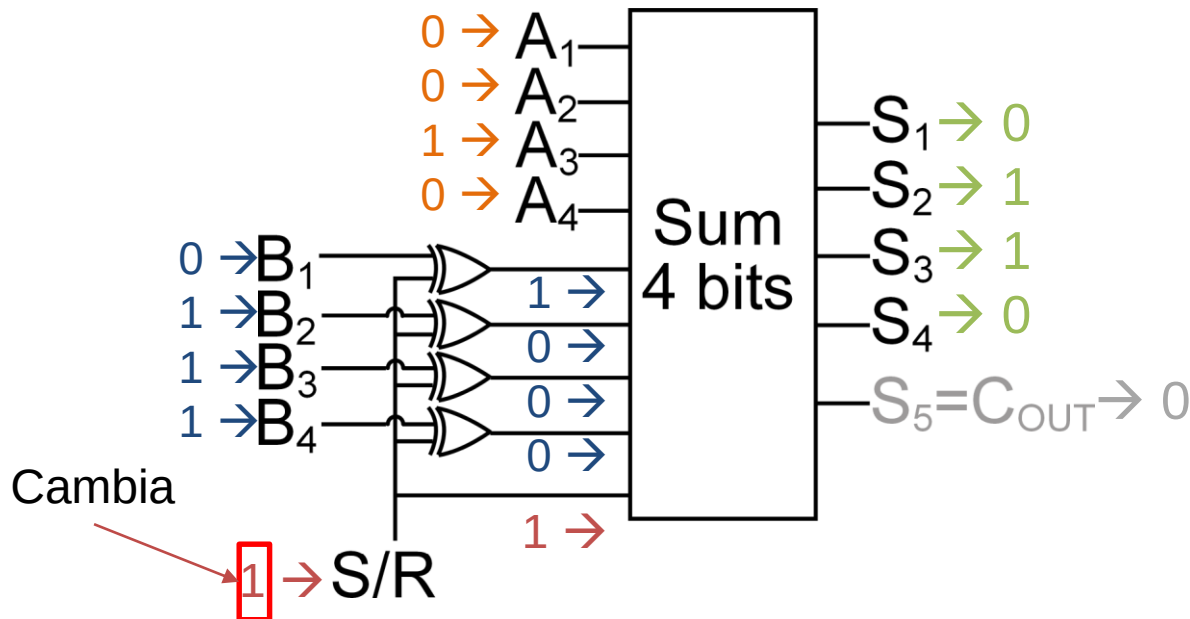


$$\begin{array}{r}
 A_4 A_3 A_2 A_1 \\
 + B_4 B_3 B_2 B_1 \\
 \hline
 S_5 S_4 S_3 S_2 S_1
 \end{array}
 \quad
 \begin{array}{r}
 0100 \\
 + 1110 \\
 \hline
 10010
 \end{array}$$

SUMADOR/RESTADOR

➤ Resta

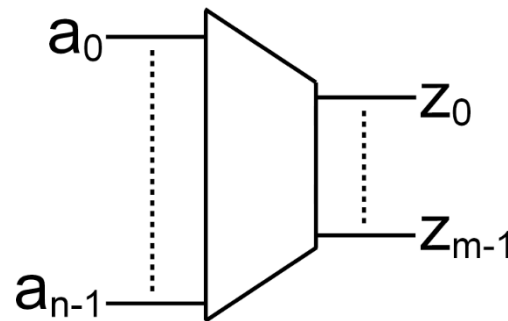
$$\begin{array}{r}
 A_4 A_3 A_2 A_1 \\
 - B_4 B_3 B_2 B_1 \\
 \hline
 S_5 S_4 S_3 S_2 S_1
 \end{array}
 \quad \rightarrow \quad
 \begin{array}{r}
 A_4 A_3 A_2 A_1 \\
 + \overline{B_4 B_3 B_2 B_1} + S/R \\
 \hline
 S_5 S_4 S_3 S_2 S_1
 \end{array}$$



$$\begin{array}{r}
 0100 \\
 - 1110 \\
 \hline
 00110
 \end{array}
 +
 \begin{array}{r}
 0100 \\
 0001 \\
 1 \\
 \hline
 00110
 \end{array}$$

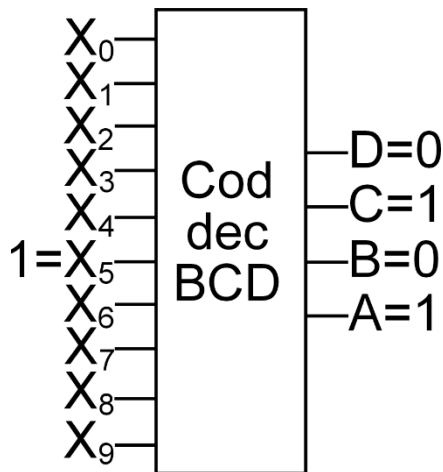
CODIFICADORES

- Es un circuito que permite transformar un nivel activo en una de sus entradas en un valor codificado
- De manera general, tiene n entradas y m salidas.

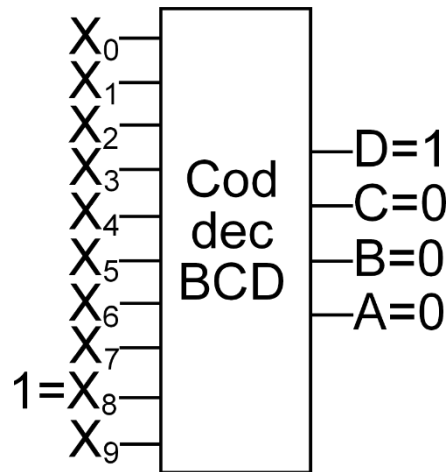


CODIFICADORES

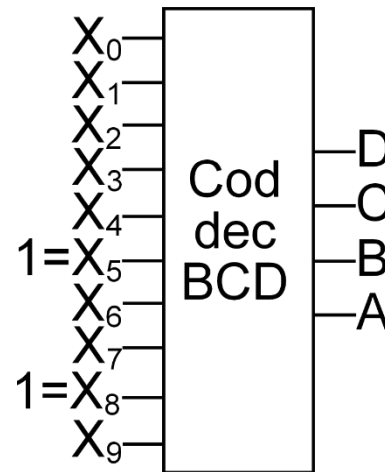
- Ejemplo: 74147 Decimal a BCD. 10 entradas y 4 salidas.
Usado en los teclados numéricos



$DCBA \rightarrow 0101$



$DCBA \rightarrow 1000$



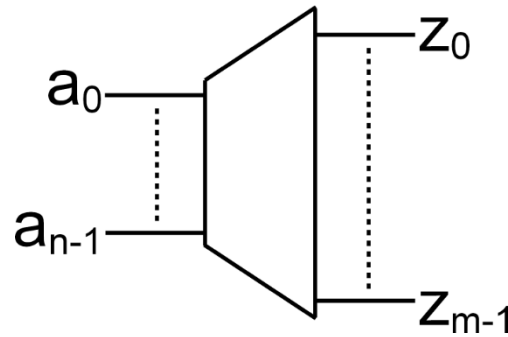
$DCBA \rightarrow \begin{cases} 0101 \\ 1000 \\ 1111 \end{cases}$

Prioridad: menor
Prioridad: mayor
Error

Si se activan varias entradas a la vez, se puede dar prioridad a una de ellas, o alertar del error

DECODIFICADORES

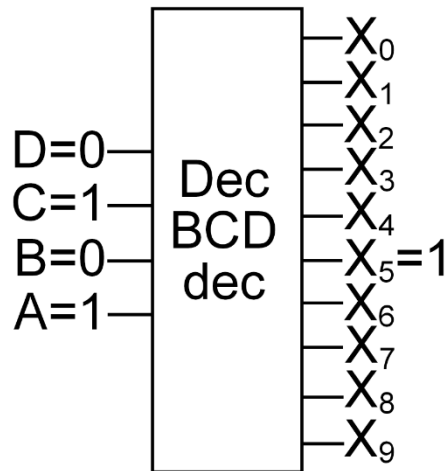
- Transforma un valor codificado en:
 - La activación de una única salida correspondiente a dicho valor. Realizan la función inversa a los codificadores.
 - Un valor codificado en otro código (convertidores de código)



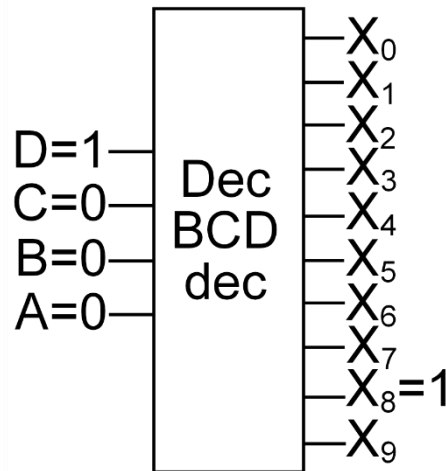
Símbolo genérico del decodificador con n entradas y m salidas

DECODIFICADOR

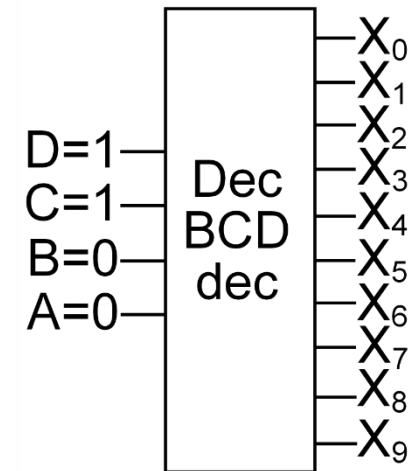
➤ Ejemplo: 7442 BCD a decimal



$DCBA \rightarrow 0101$



$DCBA \rightarrow 1000$

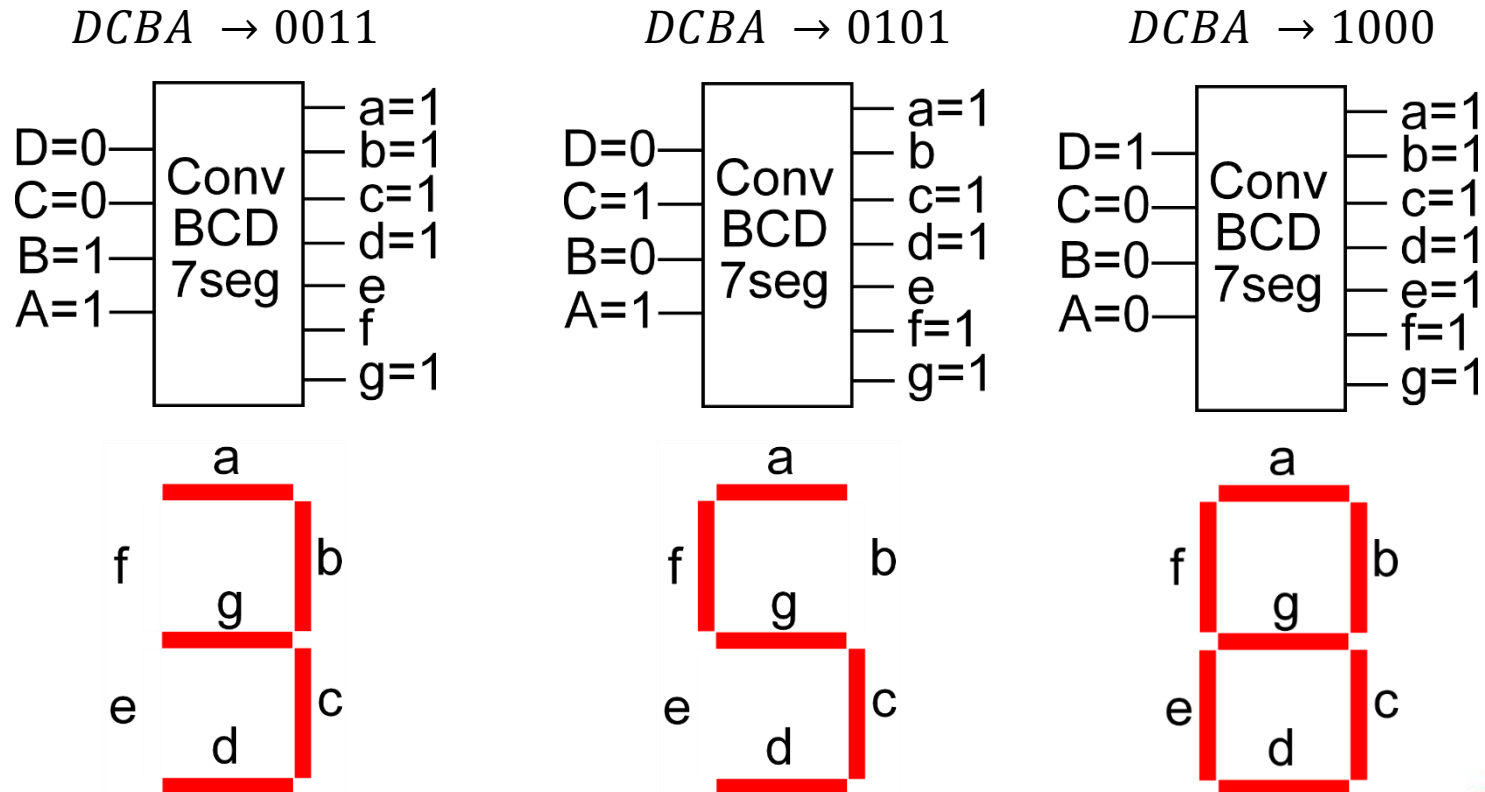


$DCBA \rightarrow 1100 \rightarrow \text{error}$

Si hay un error, las salidas quedan a gusto del diseñador

CONVERSION DE CÓDIGO

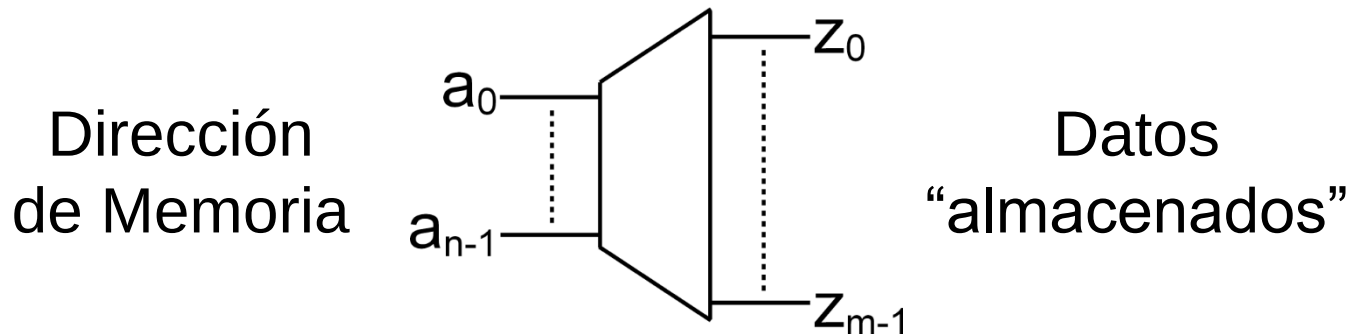
➤ Ejemplo: 7447 BCD a display 7 segmentos



[Link simulador Falstad](#)

MEMORIA ROM

- Una memoria ROM (Read Only Memory) solo se puede leer, es no volátil y su contenido está predefinido
- Se puede interpretar como un convertor de código



$a_7 \dots a_0 \rightarrow 00110011$

$a_7 \dots a_0 \rightarrow 11010101$

$a_7 \dots a_0 \rightarrow 00110011$

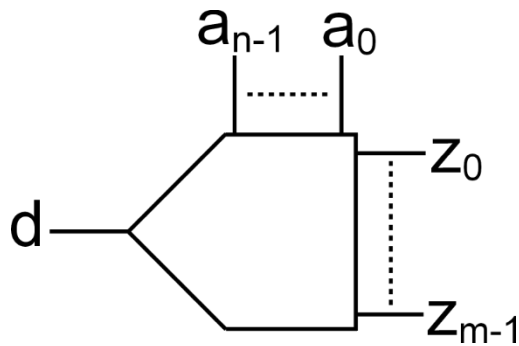
$z_7 \dots z_0 \rightarrow 011100101$

$z_7 \dots z_0 \rightarrow 001001011$

$z_7 \dots z_0 \rightarrow 011100101$

DEMULTIPLEXOR

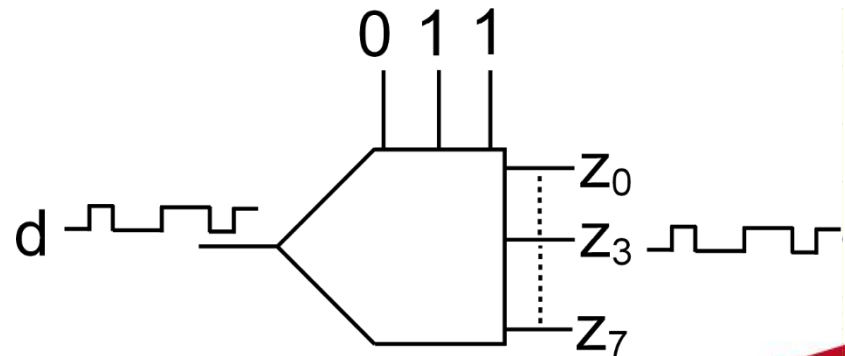
- Un demultiplexor es un circuito que copia el valor de la entrada de datos (d) en la salida (z) indicada por el valor de las señales de control (a).



Con n bits de control:
 $2^n = m$ salidas

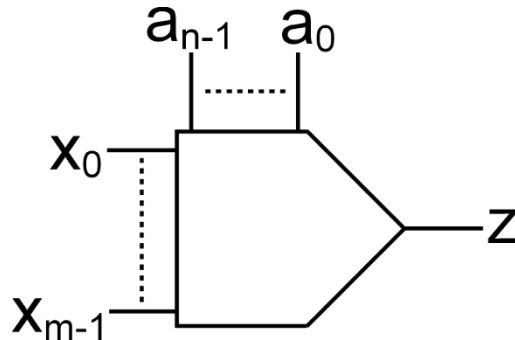
DMUX4, DMUX8...

Ejemplo para 3 bits de control y 8 salidas



MULTIPLEXOR

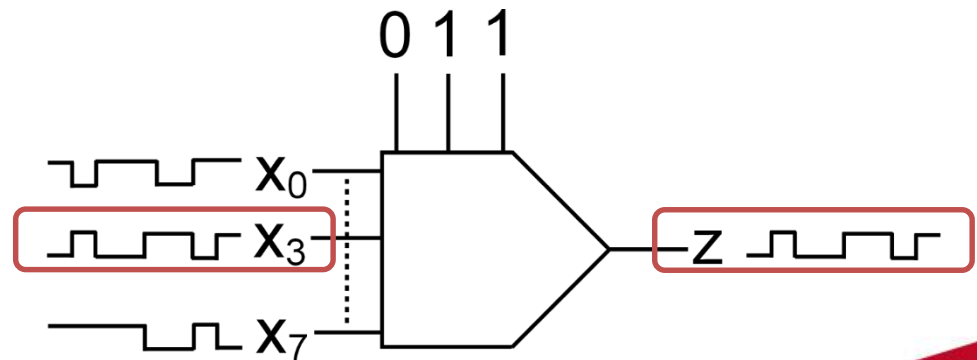
- El multiplexor es un circuito que permite seleccionar una de las entradas de datos (x) y copiar su valor a la salida (z). La entrada seleccionada depende del valor que se dé a las entradas de control (a).



Con n bits de control:
 $2^n = m$ entradas

MUX4, MUX8...

Ejemplo para 3 bits de control y 8 entradas



[Link simulador Falstad](#)

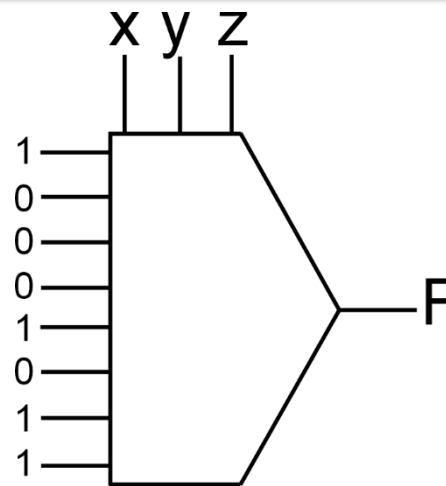
APLICACIONES DEL MULTIPLEXOR

- Un multiplexor de n entradas de control es un modulo lógico universal: permite sintetizar cualquier función lógica de n variables.

Tabla verdad de la función F que se quiere implementar

x	y	z	F
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

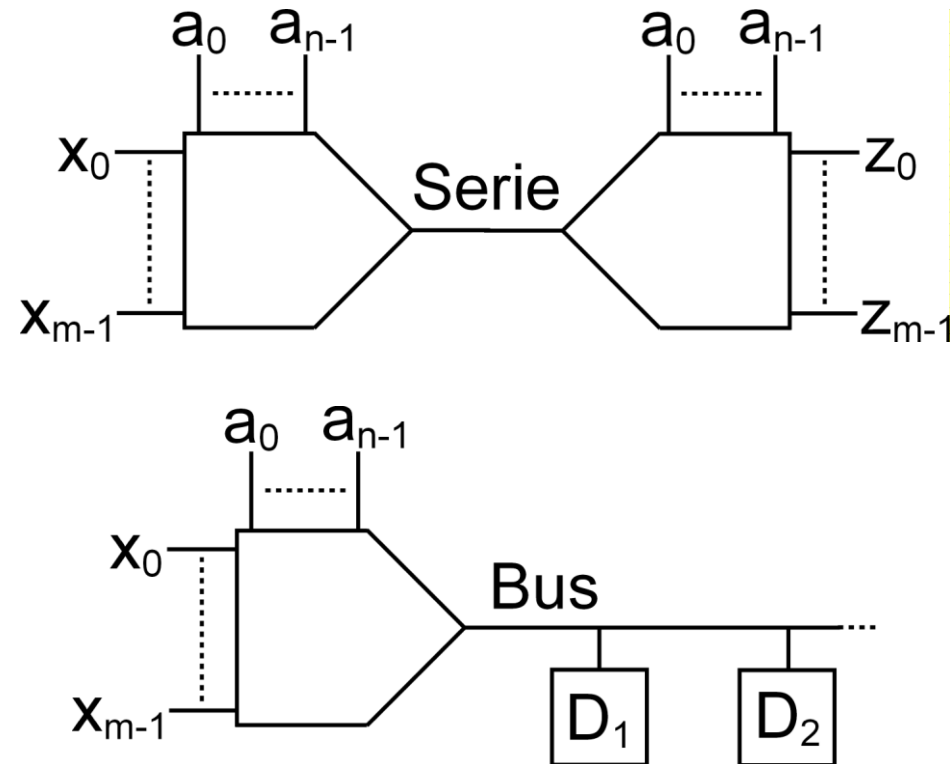
Las variables de control del multiplexor son las variables de la función a sintetizar y las entradas de datos tienen los valores 0 ó 1 correspondientes al valor de la función para cada combinación de variables.



Implementación de la función F con un multiplexor

APLICACIONES DEL MULTIPLEXOR

- Selector de datos. Sirve para convertir información en paralelo en información serie.
- Acceso a buses. El control del acceso a un bus para enviar información al mismo se puede hacer de forma cómoda mediante un multiplexor a través del cual pasen todas las entradas.

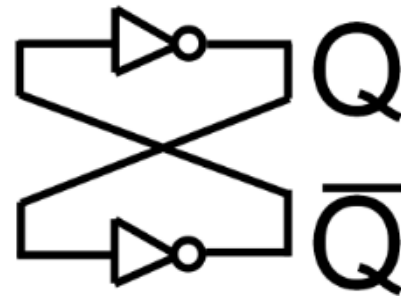
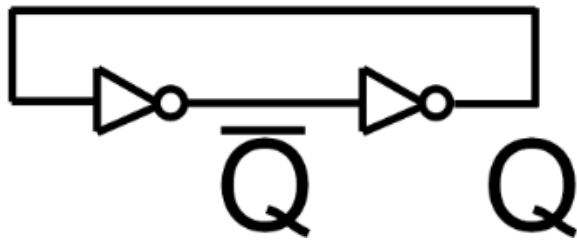


TEMA 7 – SISTEMAS DIGITALES

- Introducción. IFF
- Sistemas combinacionales
- **Biestables**
- Sistemas secuenciales

BIESTABLES

- Es un circuito capaz de almacenar un bit de información, gracias a un lazo de realimentación
- El más sencillo se basa en dos puertas NOT en serie



- El circuito se puede encontrar en dos situaciones estables (de ahí el nombre de biestable):

Estado 0 o Reset: $Q = 0$

Estado 1 o Set: $Q = 1$

TIPOS DE BIESTABLES

- Para incluir entradas que nos permitan llevar al biestable a uno de sus estados, vamos a cambiar las puertas NOT por NAND o NOR
- La filosofía es que el valor almacenado ($Q = 0$ ó 1) se mantiene hasta que las entradas provoquen un cambio.
- Tipos de biestables:
 - Asíncronos (latch SR o D):
 - Q puede cambiar al cambiar cualquier entrada
 - Síncronos (flip-flop D, JK o T):
 - tienen una señal de control (Clock) que indica cuándo pueden cambiar el valor de Q

LATCH SR ASÍNCRONO

- Existe versión basada en dos puertas NOR y en dos puertas NAND. Veamos la NOR:
- Si analizamos el circuito nos encontramos con tres posibilidades para las entradas que determinan las salidas y una posibilidad ($S=R=0$) para la que las salidas no están influenciadas por las entradas.

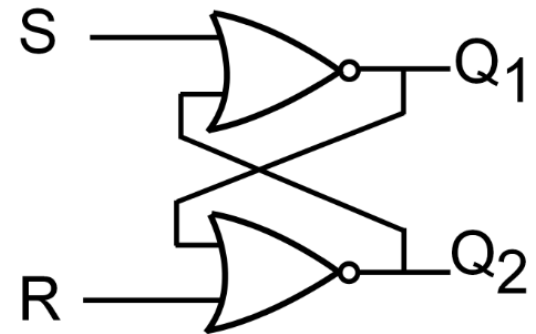
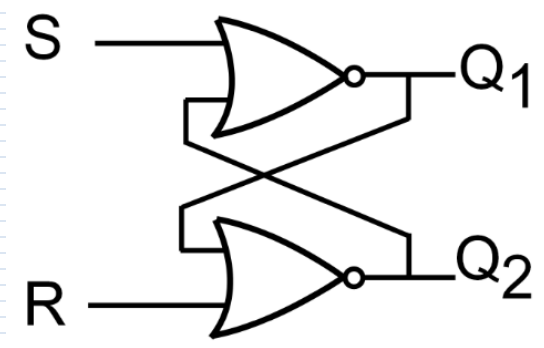


Tabla verdad

S	R	Q ₂	Q ₁
0	0	$\overline{Q_1}$	$\overline{Q_2}$
1	0	1	0
0	1	0	1
1	1	0	0

LATCH SR ASÍNCRONO

[Link simulador Falstad](#)



S	R	Q ₂	Q ₁
0	0		
1	0		
0	1		
1	1		

LATCH SR ASÍNCRONO

- Denominamos
 - Estado 0 cuando $Q_2 = 0$ y $Q_1 = 1$
 - Estado 1 cuando $Q_2 = 1$ y $Q_1 = 0$
- Entonces, el circuito presenta un tercer estado ($Q_1 = Q_2 = 0$). No nos interesa, por lo que evitaremos que el sistema vaya a ese estado. Es el **estado prohibido**.
- Así, las salidas Q_2 y Q_1 son siempre una la negada de la otra, por lo que las denominamos Q y $\bar{Q}$. Y lo que es más importante, la combinación $S = R = 0$, mantiene el estado por lo que el biestable mantendrá el último estado escrito.

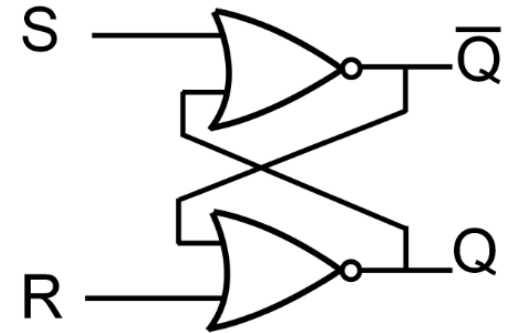


Tabla verdad

S	R	Q	$\bar{Q}$
0	0	Q	$\bar{Q}$
1	0	1	0
0	1	0	1
1	1	0	0

DINÁMICA

- El sistema permanece estable mientras $S=R=0$. Se dice que ambas entradas están inactivas.
- Si activamos S ($S=1, R=0$) el sistema irá al estado 1. Se denomina S porque lleva al estado Set.
- Si activamos R ($R=1, S=0$) el sistema irá al estado 0. Se denomina R porque lleva al estado Reset.
- El circuito recuerda la última activación en S o R.
- Las dos entradas no pueden activarse a la vez, ya que el sistema iría al estado prohibido.

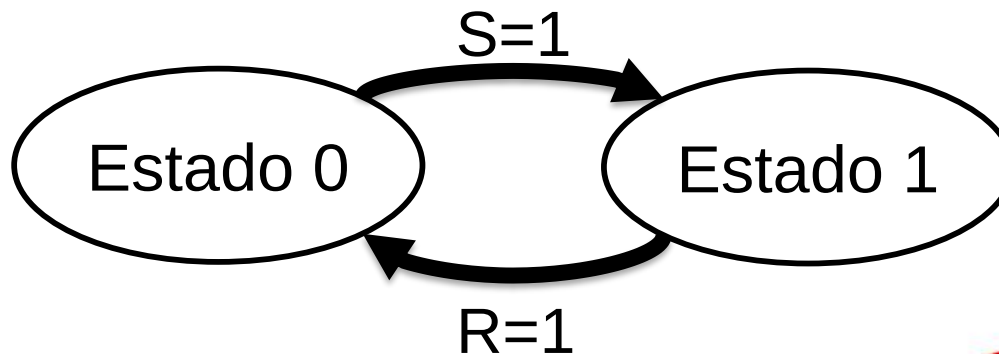
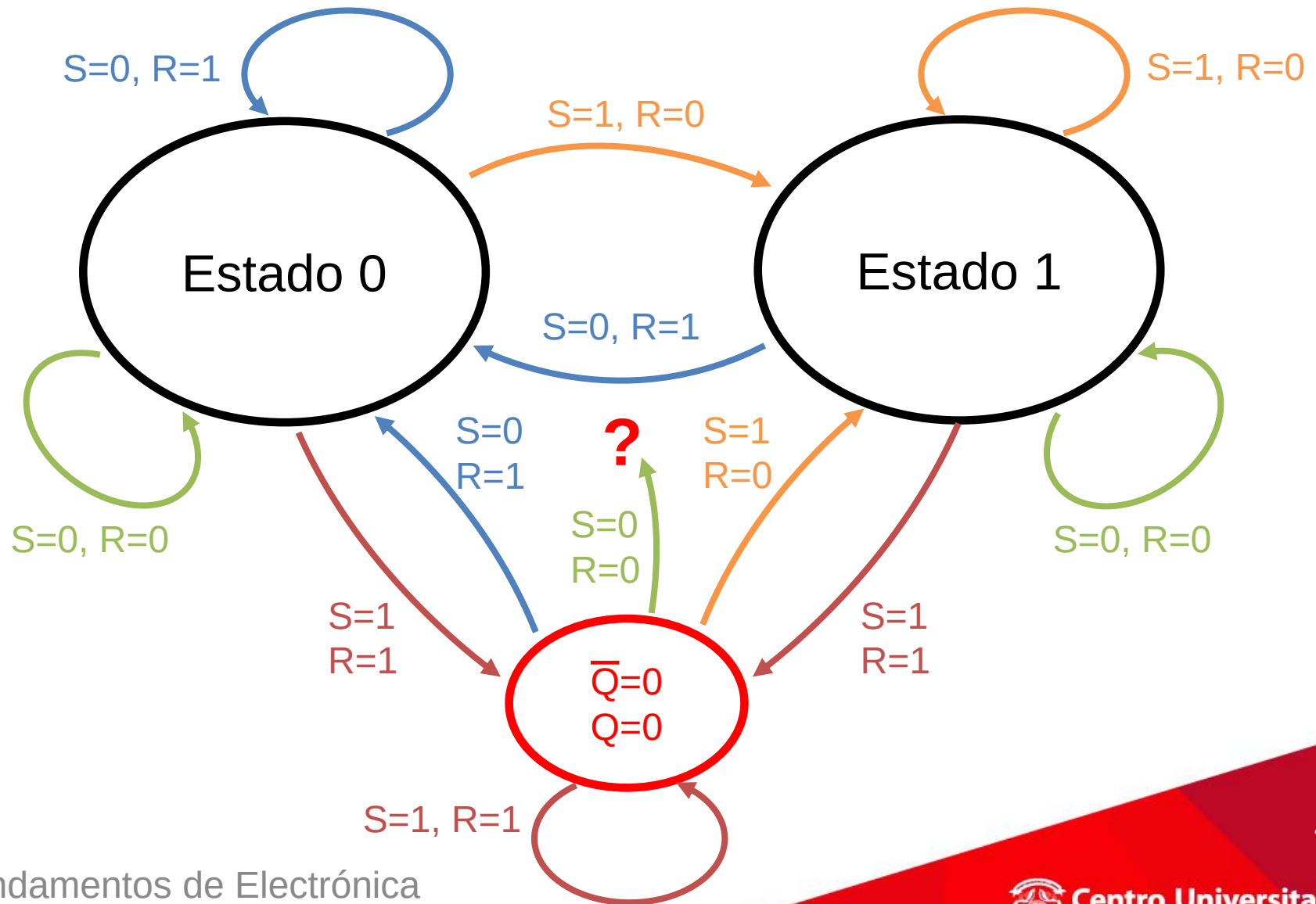
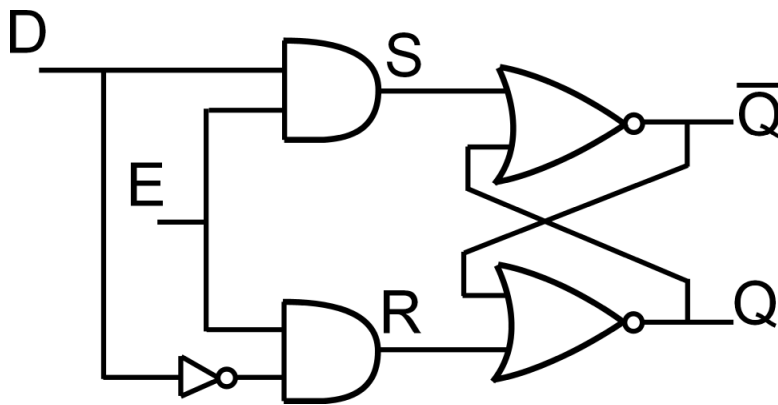


DIAGRAMA DE ESTADOS



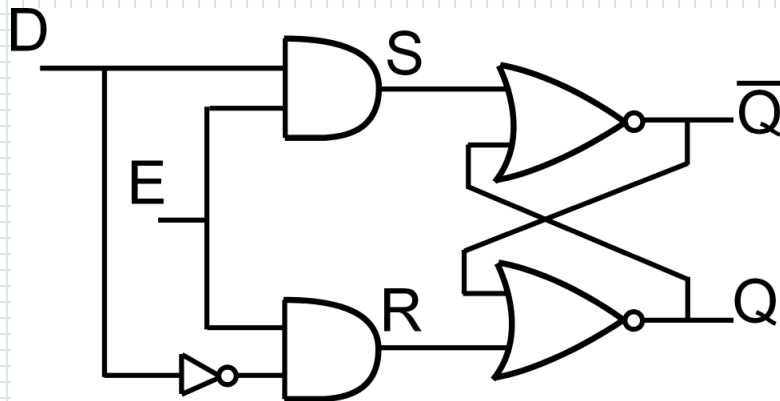
LATCH D

- Está formado por un latch SR y lógica adicional para evitar el estado prohibido
- Por definición, si la señal de control “Enable” (E) está inactiva, el biestable mantiene el estado y si está activa el valor de la entrada D se escribe en la salida Q.
- El Enable puede ser activo en bajo o en alto. Si el Enable es activo en alto:



E	D	S	R	Q	$\bar{Q}$
0	-	0	0	Q	$\bar{Q}$
1	1	1	0	1	0
1	0	0	1	0	1

LATCH D



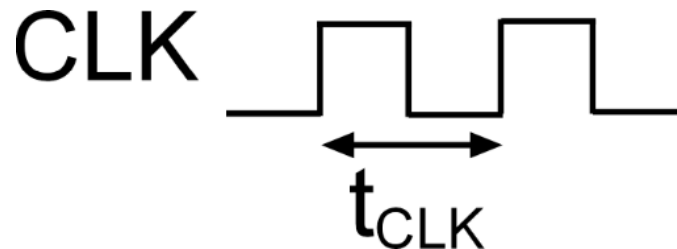
S	R	Q	$\bar{Q}$
0	0	Q	$\bar{Q}$
1	0	1	0
0	1	0	1
1	1	0	0

[Link simulador Falstad](#)

E	D	S	R	Q	$\bar{Q}$
0	-				
1	1				
1	0				

CLOCK (RELOJ)

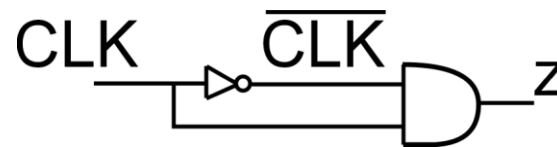
- Clock: Señal formada por una serie de pulsos intermitentes con un ancho específico.



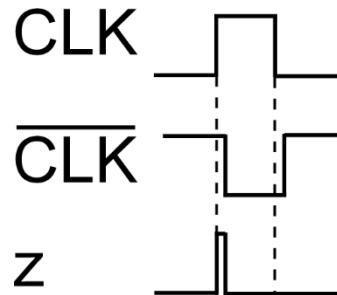
- Tiempo de ciclo del reloj t_{CLK} : intervalo entre los flancos correspondientes de dos pulsos consecutivos.
- Para marcar un instante temporal para sincronizar varios biestables, usamos los flancos (de subida o bajada)

DETECTOR DE FLANCO

- Circuito que tiene salida activa (en alto o bajo) solo cuando se produce un flanco (de subida o bajada)
- Versión activa en alto para flanco de subida:

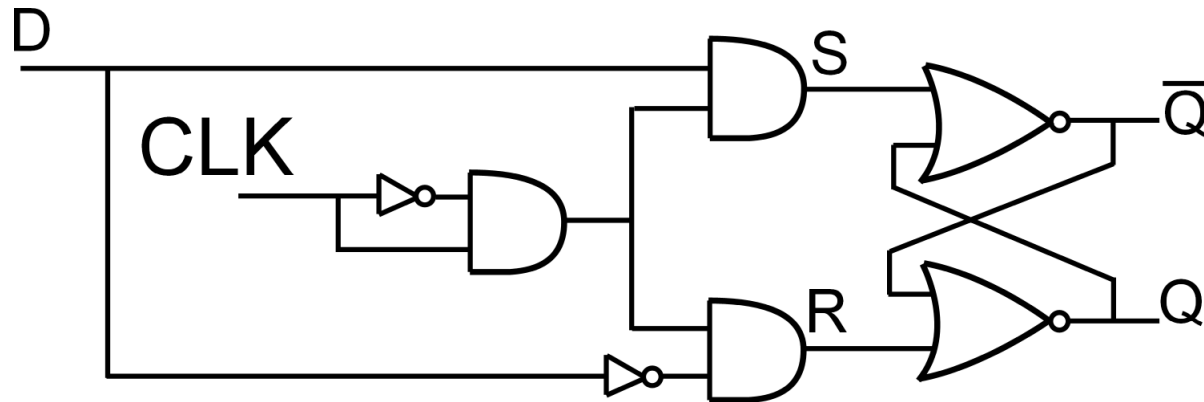


- La salida z será 1 solo cuando se produce el flanco de subida, debido al retardo que se produce en el inversor



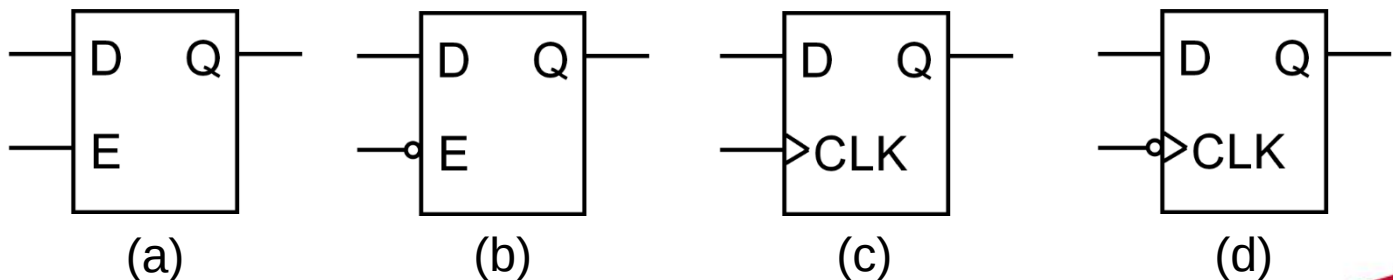
FLIP FLOP D

- Formado por un latch D y un detector de flanco
- El biestable solo es activo (Q toma el valor de D), cuando el reloj tiene una transición. Así se pueden sincronizar los cambios de varios flip flop.
- Versión activa con flanco de subida (latch D activo en alto y detector de flanco con salida en alto con flanco de subida de reloj)

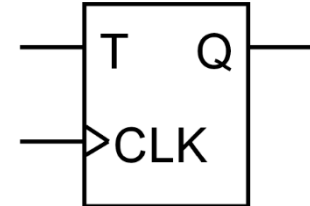
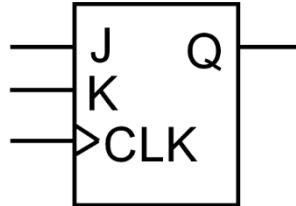
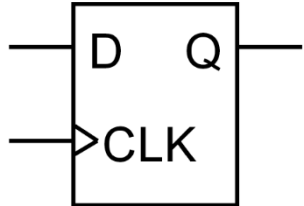


BIESTABLES TIPO D

- Un latch es activado por nivel en el sentido de que se pueden producir transiciones de estado mientras la señal de control (enable) esté en nivel 1 (o 0) (**level triggered**)
- Un flip-flop es activado por flanco de subida en el sentido de que la transición de estado se produce sólo cuando la señal de control (reloj) pasa de 0 a 1 (o de 1 a 0) (**edge triggered**)
- Latch D activado con nivel alto (a) o con nivel bajo (b)
- Flip-Flop D activado con flanco de subida (c) o de bajada (d)



TIPOS DE FLIP FLOP: D, JK Y T



CLK	D	Q	$\bar{Q}$
0	-	Q	$\bar{Q}$
1	-	Q	$\bar{Q}$
↓	-	Q	$\bar{Q}$
↑	0	0	1
↑	1	1	0

CLK	J	K	Q	$\bar{Q}$
0	-	-	Q	$\bar{Q}$
1	-	-	Q	$\bar{Q}$
↓	-	-	Q	$\bar{Q}$
↑	0	0	Q	$\bar{Q}$
↑	1	0	1	0
↑	0	1	0	1
↑	1	1	$\bar{Q}$	Q

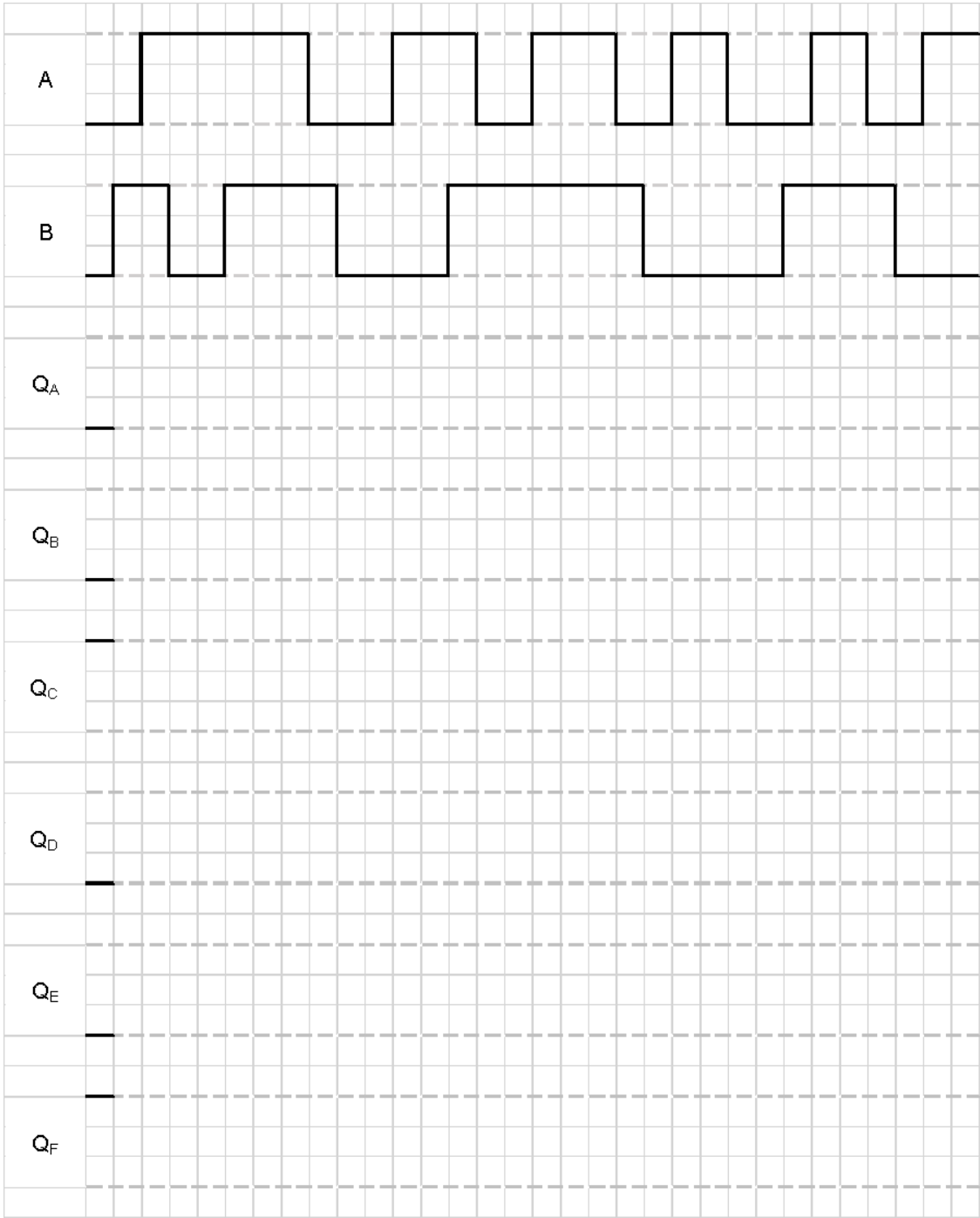
CLK	T	Q	$\bar{Q}$
0	-	Q	$\bar{Q}$
1	-	Q	$\bar{Q}$
↓	-	Q	$\bar{Q}$
↑	0	Q	$\bar{Q}$
↑	1	$\bar{Q}$	Q

[Link simulador Falstad](#)

EJERCICIO 1

Dado un biestable con entrada de control A (Enable para Latch/Clock para Flip Flop) y entrada de datos B (D para tipo D, T para tipo T), complete el cronograma en función del tipo de biestable:

1. QA: Latch tipo D activo en alto
2. QB: Latch tipo D activo en bajo
3. QC: Flip Flop tipo D activo en subida
4. QD: Flip Flop tipo D activo en bajada
5. QE: Flip Flop tipo T activo en subida
6. QF: Flip Flop tipo T activo en bajada



D	Q	$\bar{Q}$
0	0	1
1	1	0

T	Q	$\bar{Q}$
0	Q	$\bar{Q}$
1	$\bar{Q}$	Q

J	K	Q	$\bar{Q}$
0	0	Q	$\bar{Q}$
1	0	1	0
0	1	0	1
1	1	$\bar{Q}$	Q

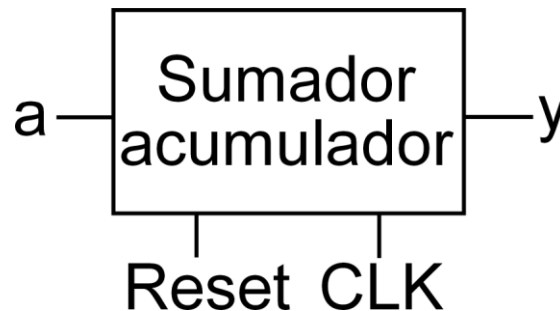
TEMA 7 – SISTEMAS DIGITALES

- Introducción. IFF
- Sistemas combinacionales
- Biestables
- Sistemas secuenciales



SISTEMAS SECUENCIALES

- Incluyen biestables
- Salida depende de las entradas y del estado:
 - Latch/Flip Flop
 - Registros
 - Contadores
 - Memorias SRAM
- Ejemplo: sumador - acumulador

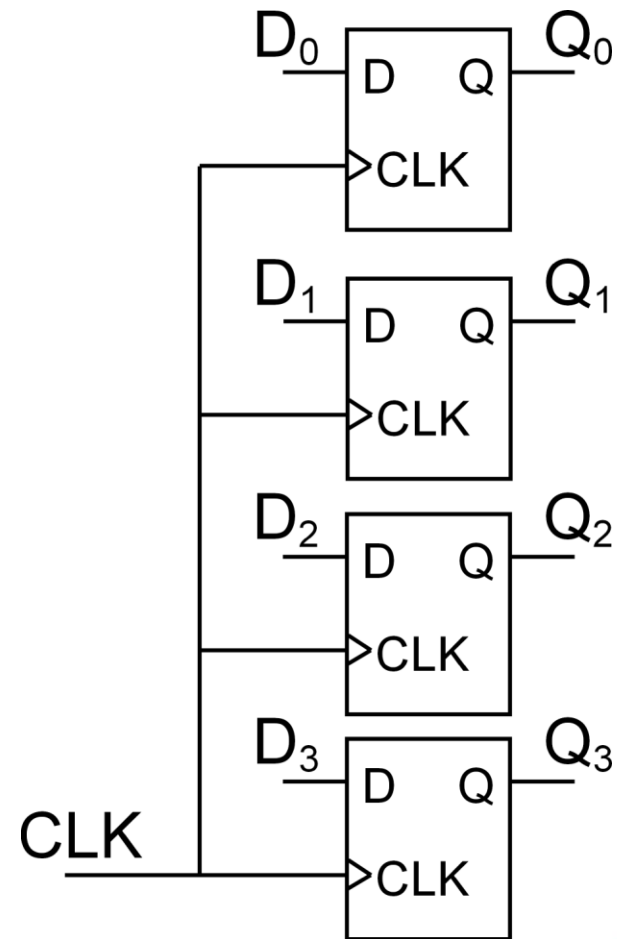


AGRUPACIONES DE FLIP FLOPS

- Flip flops D en paralelo: Registros de almacenamiento
- Flip flops D en serie: Registros de desplazamiento
- Flip flops T en serie: contador asíncrono
- Flip flops T en paralelo: contador síncrono
- Matriz de flip flops D: Memoria SRAM

REGISTRO DE ALMACENAMIENTO

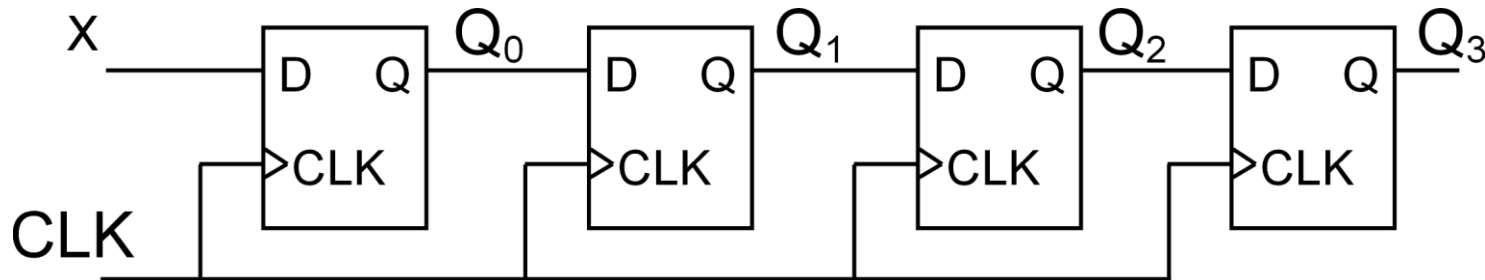
- Conexión en paralelo de flip-flops D
- Se produce una carga síncrona de los datos D_i en la salida correspondiente Q_i
- El dato queda almacenado hasta la siguiente activación del reloj



REGISTRO DE DESPLAZAMIENTO

- Conexión en serie de flip-flops D

[Link simulador Falstad](#)



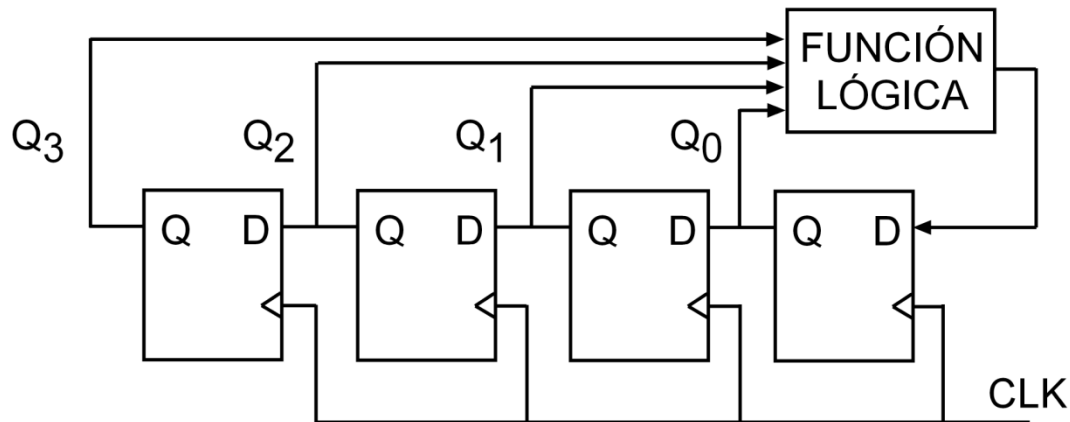
- En cada activación del reloj se produce un desplazamiento de los datos hacia la derecha, ya que cada biestable toma el valor que tiene en su entrada

$$Q_{i+1} = Q_i \quad (Q_0 = X)$$

- No se produce un desplazamiento en cadena porque el tiempo de retardo de cada biestable es superior al de activación del reloj

REGISTROS CON REALIMENTACIÓN

- Registro de desplazamiento (a la izquierda) realimentado

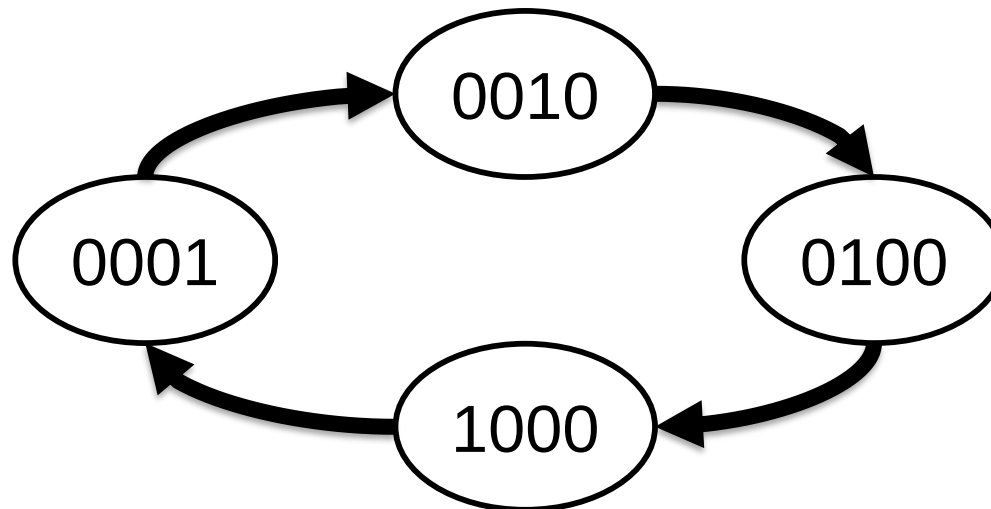
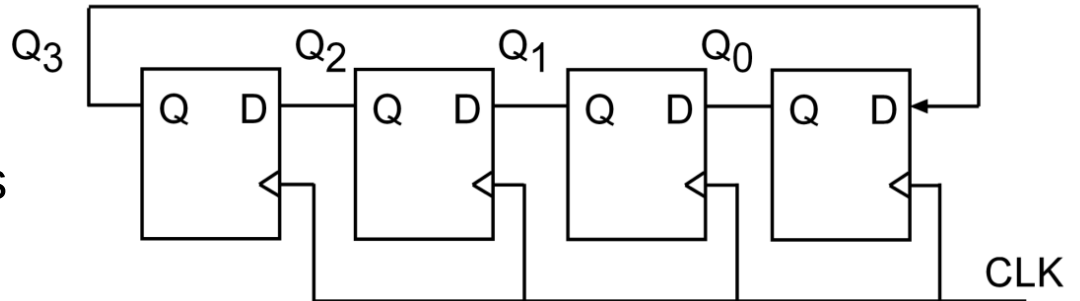


- Estado definido por el valor de las salidas: $Q_3Q_2Q_1Q_0$
- Número de estados limitado ($\leq 2^N$) que se repiten cíclicamente
- Suelen requerir inicialización (o corrección de estados no permitidos)
- Ejemplo: contadores de anillo y doble anillo

CONTADOR EN ANILLO

➤ Contador de anillo

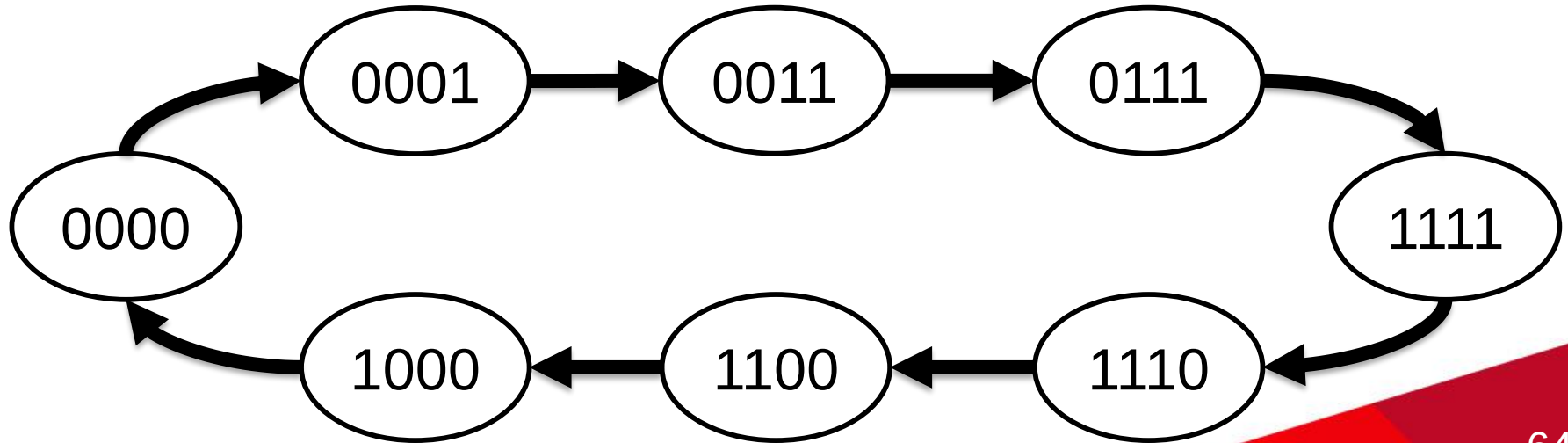
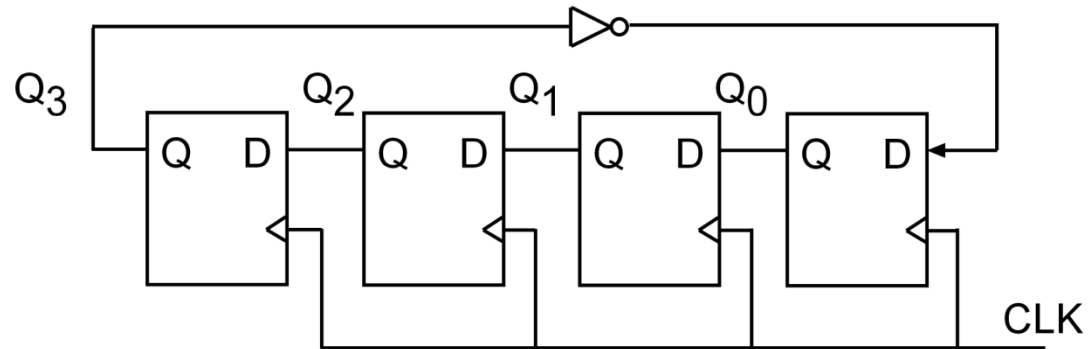
- Salida serie →
Entrada serie
- N estados permitidos



CONTADOR DE DOBLE ANILLO

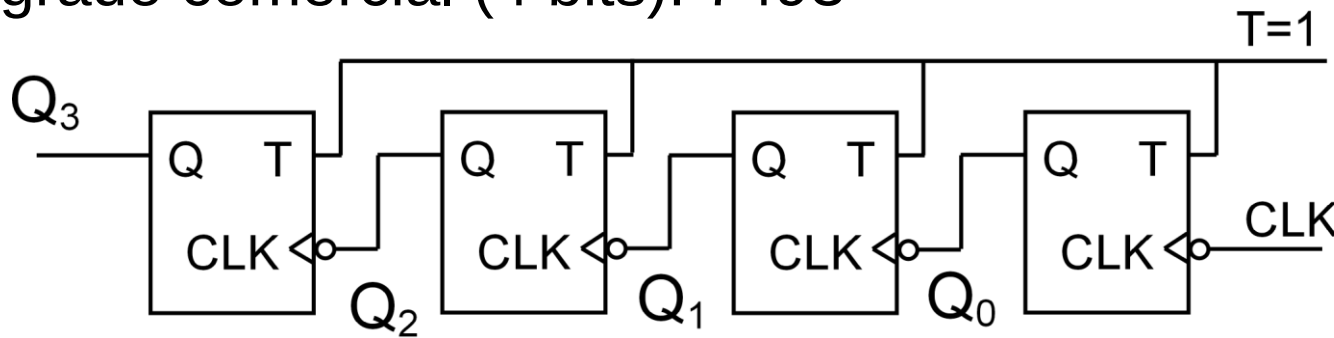
➤ Contador de doble anillo

- Salida serie negada →
Entrada serie
- $2N$ estados permitidos

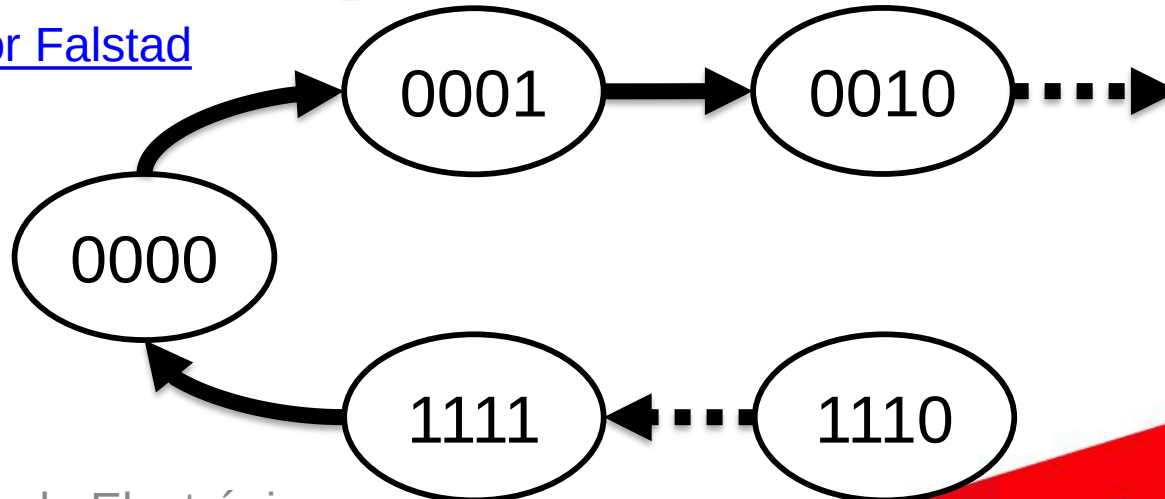


CONTADORES ASÍNCRONOS

- Conexión “serie” de flip-flops T (todos activos con $T=1$)
- Reloj (activo en flanco de bajada) no común: **Asíncrono**
- Integrado comercial (4 bits): 7493



[Link simulador Falstad](#)

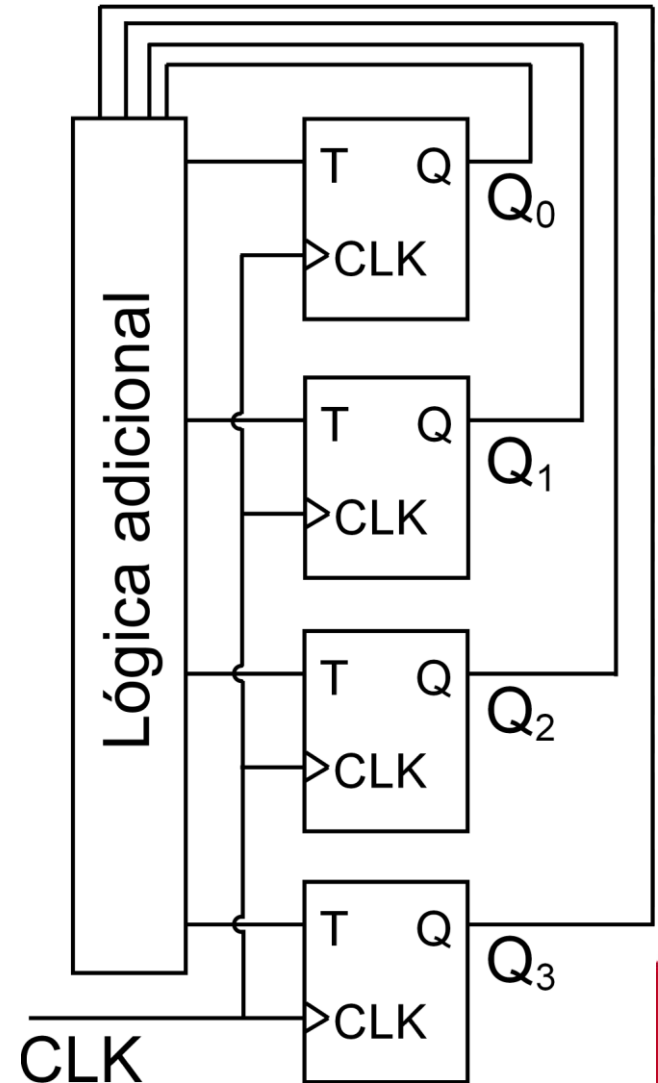


CONTADORES SÍNCRONOS

- Conexión “paralelo” de flip-flops T
- Requiere lógica adicional (mayor complejidad)
- Retardo no acumulativo (mayor velocidad)
- Reloj común: **síncrono**
- Integrado comercial (4 bits): 74163

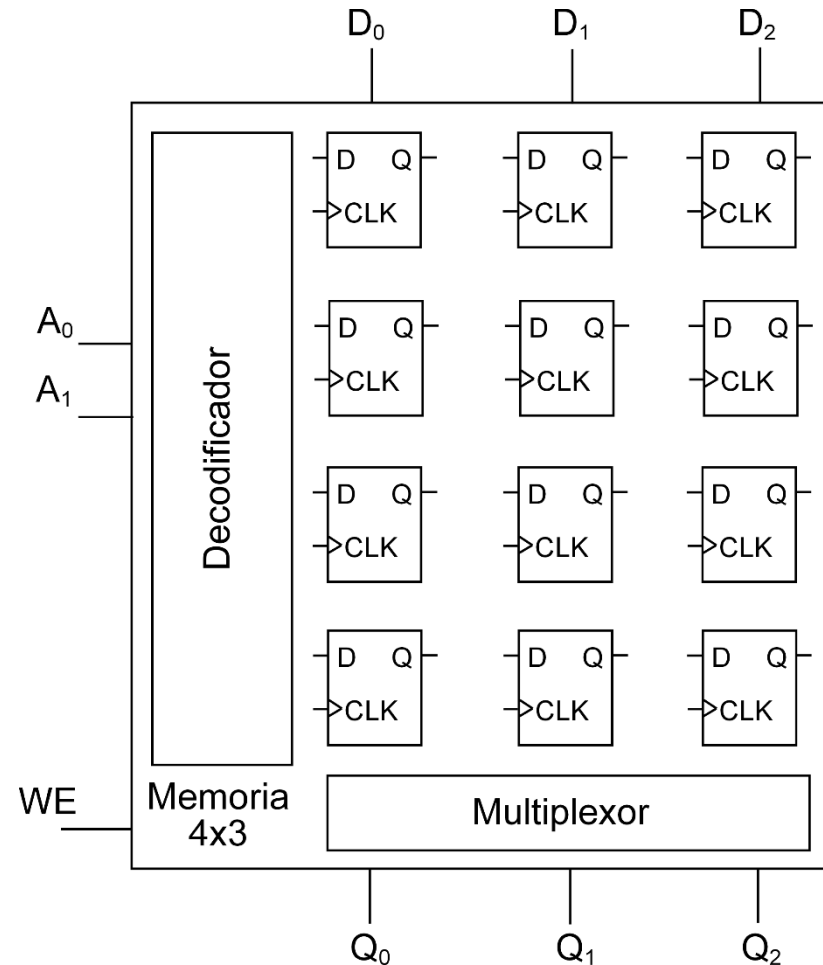
[Link simulador Falstad](#)

[Link simulador Falstad](#)



MEMORIA SRAM

- Static Random Access Memory
- Estructura:
 - Matriz $m \times n$ de Flip Flop D
 - Decodificador
 - n Multiplexores
- Características:
 - Volátil
 - Estática
 - Escribe/lee por filas
 - Capacidad: 1 bit por biestable

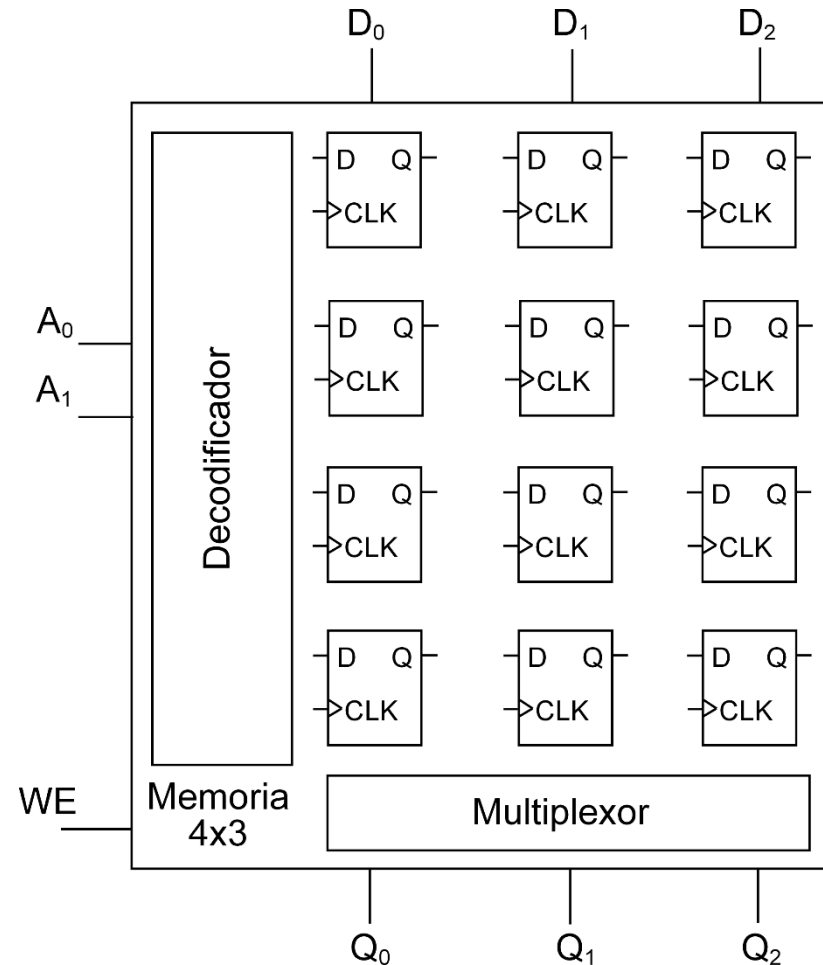


MEMORIA SRAM

➤ Ejemplo:

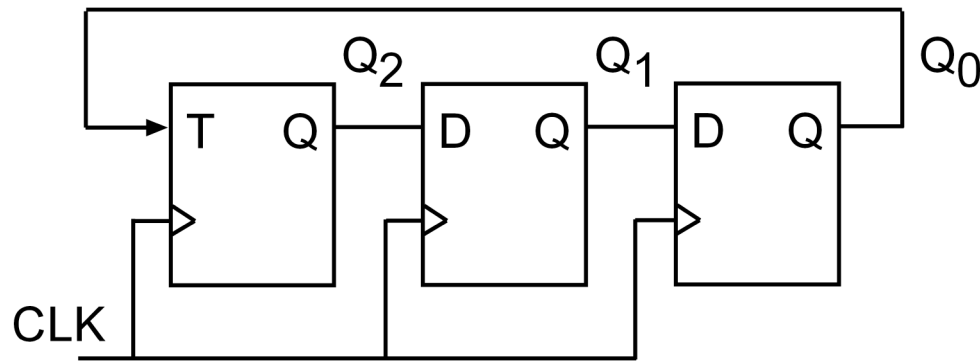
- 3 datos de entrada (D):
3 columnas de biestables
- 2 para la dirección (A): El decodificador indica una única fila. 4 filas de biestables
- 3 datos de salida (Q): los multiplexores llevan a la salida la fila indicada por el decodificador
- 1 control de escritura (WE: write enable)
 - WE = 0: lectura
 - WE = 1: escritura

➤ Total: $4 \times 3 = 12$ bits

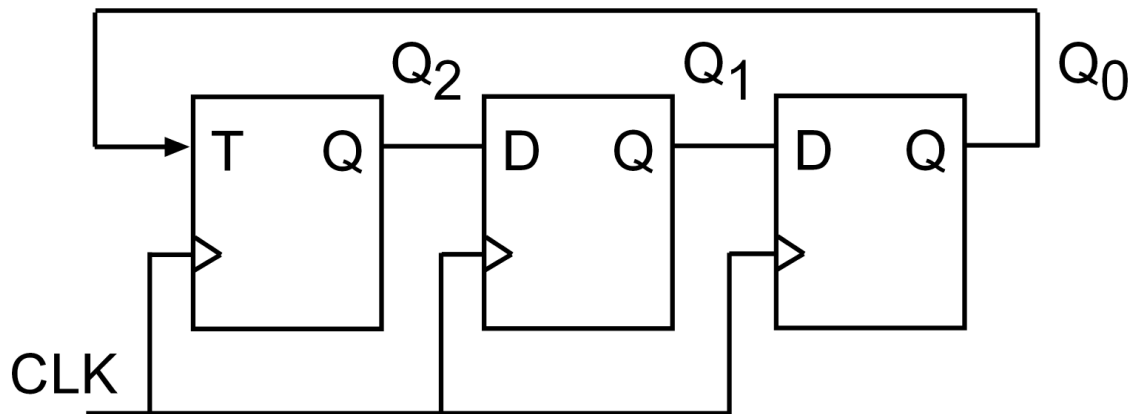


EJERCICIO 2

- Dado el siguiente circuito formado por 3 flip-flops (uno T y dos D), todos activos por flanco de subida de reloj.



- Obtenga la secuencia de todos los estados posibles $Q_2Q_1Q_0$. ¿Cuántos ciclos cerrados se forman?



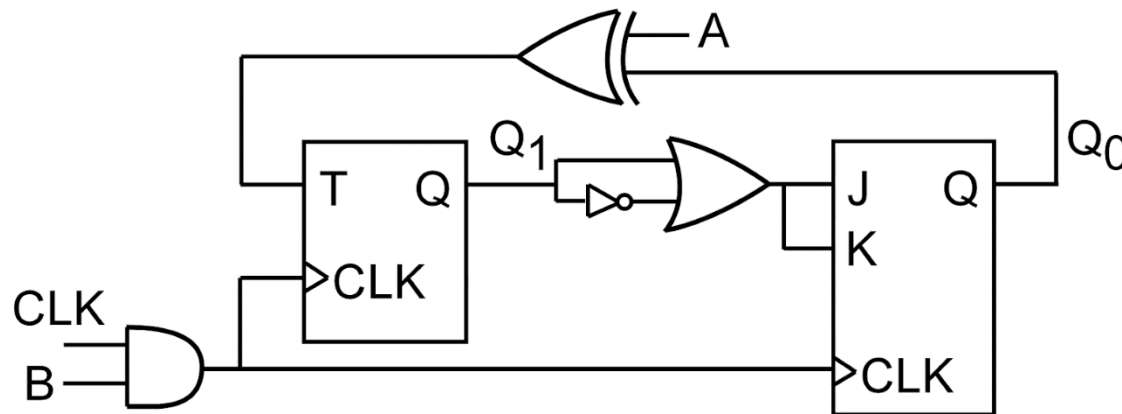
D	Q	$\bar{Q}$
0	0	1
1	1	0

T	Q	$\bar{Q}$
0	Q	$\bar{Q}$
1	$\bar{Q}$	Q

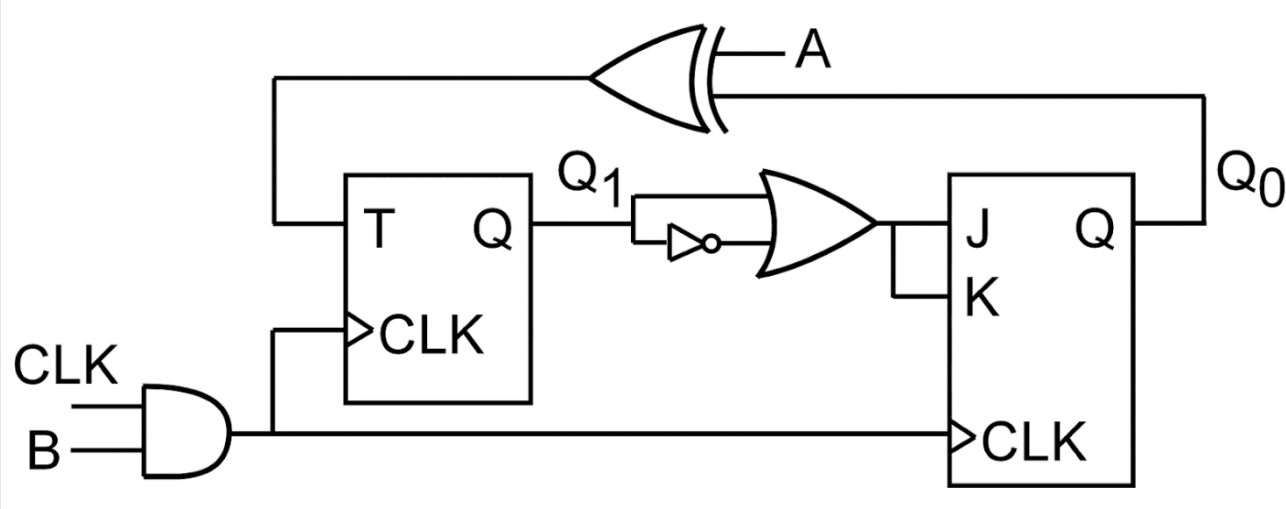
[Link simulador Falstad](#)

EJERCICIO 3

- Dado el siguiente circuito basado en 2 flip-flops (uno T y uno JK) ambos activos por flanco de subida de reloj.



- Obtenga la secuencia de todos los estados posibles Q_1Q_0 para:
- $B = 1, A = 0$
 - $B = 1, A = 1$
- ¿Qué ocurre si $B = 0$?



T	Q	$\bar{Q}$
0	Q	$\bar{Q}$
1	$\bar{Q}$	Q

J	K	Q	$\bar{Q}$
0	0	Q	$\bar{Q}$
1	0	1	0
0	1	0	1
1	1	$\bar{Q}$	Q

[Link simulador Falstad](#)