Organización de Computadoras
2013

Turno Recursantes

Clase 3

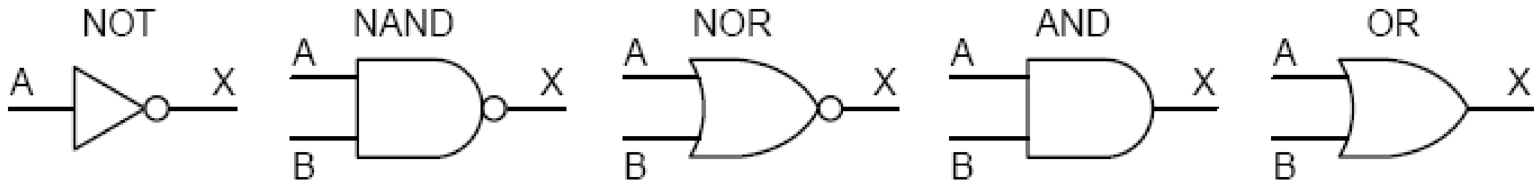
Conceptos básicos

- Lógica digital.
- Álgebra de Boole.
- Circuitos Lógicos Combinacionales
- Circuitos Lógicos Secuenciales

El nivel de lógica digital

- Un circuito digital es en el que están presentes dos valores lógicos
- Compuertas son dispositivos electrónicos que pueden realizar distintas funciones con estos dos valores lógicos
- Como vimos en el Ingreso las compuertas básicas son: AND, OR, NOT, NAND, NOR y XOR

Compuertas: símbolo y descripción funcional



A	X
0	1
1	0

A	B	X
0	0	1
0	1	1
1	0	1
1	1	0

A	B	X
0	0	1
0	1	0
1	0	0
1	1	0

A	B	X
0	0	0
0	1	0
1	0	0
1	1	1

A	B	X
0	0	0
0	1	1
1	0	1
1	1	1

Algebra Booleana

- Para describir los circuitos que pueden construirse combinando compuertas, se requiere un nuevo tipo de álgebra, donde las variables y funciones sólo puedan adoptar valores 0 ó 1: álgebra booleana.

Algebra Booleana

- Puesto que una función booleana de n variables tiene 2^n combinaciones de los valores de entrada, la función puede describirse totalmente con una tabla de 2^n renglones, donde c/u indica un valor de la función (0 ó 1) para cada combinación distinta de las entradas:

=> ***tabla de verdad***

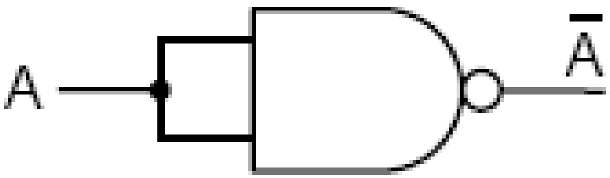
Recordemos algunas identidades del álgebra booleana

Identidad	$1.A=A$	$0+A=A$
Nula	$0.A=0$	$1+A=1$
Idempotencia	$A.A=A$	$A+A=A$
Inversa	$A.\overline{A}=0$	$A+\overline{A}=1$
Conmutativa	$A.B=B.A$	$A+B=B+A$
Asociativa	$(AB).C=A(BC)$	$(A+B)+C=A+(B+C)$
Distributiva	$A+B.C=(A+B).(A+C)$	$A.(B+C)=AB+AC$
Absorción	$A.(A+B)=A$	$A+A.B=A$
De Morgan	$\overline{A.B}=\overline{A}+\overline{B}$	$\overline{A+B}=\overline{A}.\overline{B}$

Leyes de De Morgan

- Ejemplo: construir un NOT con NAND

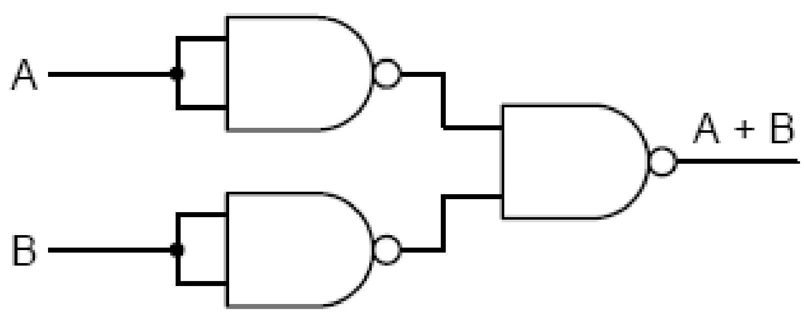
$$F=\overline{A.B}=\overline{A.A}=\overline{A}$$



Leyes de De Morgan

- Ejemplo: construir un OR con NAND

$$F = A+B = \overline{\overline{A+B}} = \overline{\overline{A}\ .\ \overline{B}}$$



Implementación de funciones booleanas

- Escribir la tabla de verdad para la función
- Dibujar una AND para cada término que tiene un 1 en la columna de resultado (con sus entradas apropiadas)
- Invertir las entradas necesarias
- Unir todas las AND a una OR

Implementación

Ejemplo: construir la tabla de verdad e implementar el circuito de una función booleana M, de tres entradas A, B y C, tal que M=1 cuando la cantidad de '1' en A, B y C es $\geq$ 2 y M=0 en otro caso.

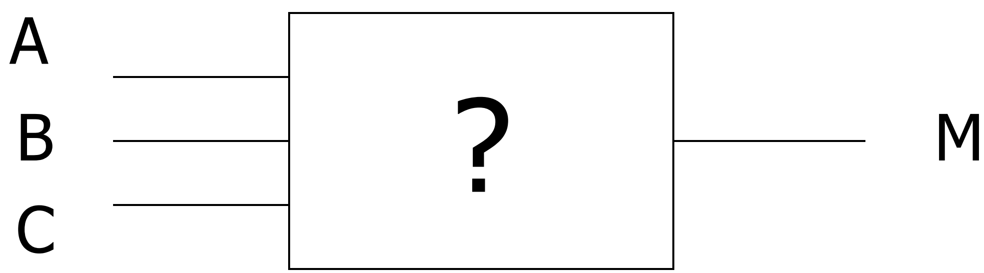


Tabla de verdad

A	B	C	M
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

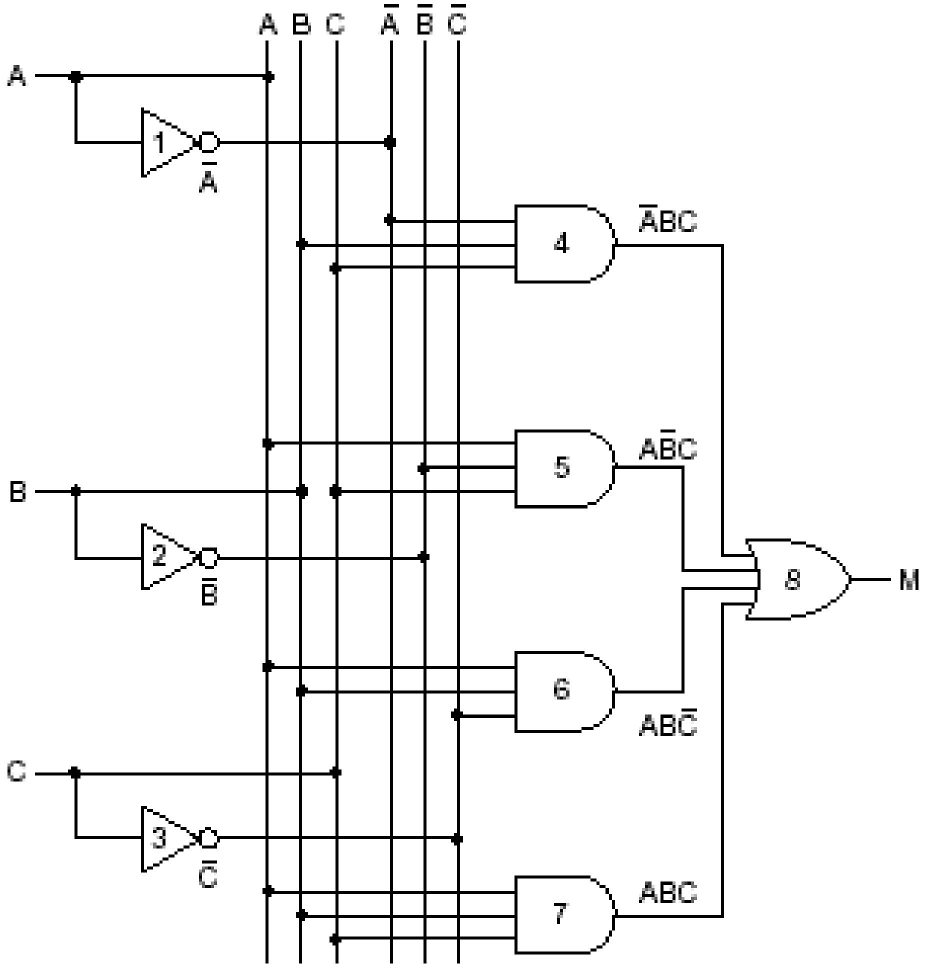
Función M

$$M=\bar{A}BC + A\bar{B}C + AB\bar{C} + ABC$$

- Hay tantos términos como 1s en la tabla
- Cada término vale 1 para una única combinación de A, B y C
- Las variables que valen 0 en la tabla aparecen aquí negadas

Función M (2)

$M=\bar{A}BC + A\bar{B}C + AB\bar{C} + ABC$



Otro ejemplo

Supongamos la siguiente Tabla de Verdad

A	B	M
0	0	0
0	1	1
1	0	1
1	1	0

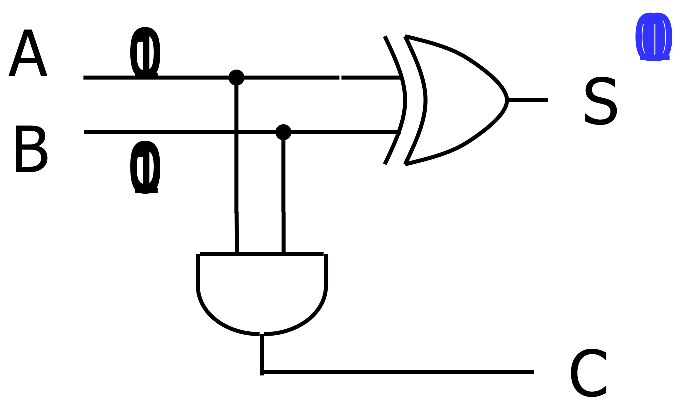
Función $\mathbf{M = \overline{A}B + A\overline{B}}$ $\Rightarrow$ **M = A XOR B**

Recordemos

- ✓ En un AND, basta que una de sus entradas sea 0 para que la función valga 0.
- ✓ En un OR, basta que una de sus entradas sea 1 para que la función valga 1.
- ✓ Hacer el XOR con 1 invierte el valor de la variable.
- ✓ Hacer el XOR con 0 deja el valor de la variable como estaba.

Circuitos combinatorios

- Ejemplo



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

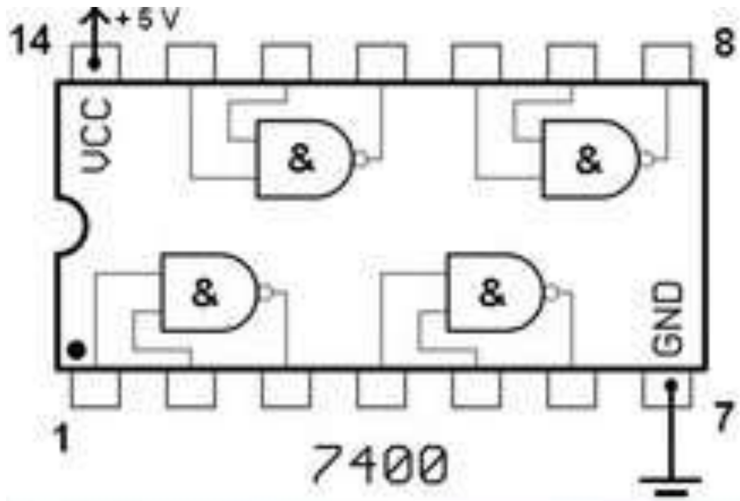
S representa la suma aritmética de 2 bits y **C** es el acarreo

Semi-sumador ó Half adder

Circuitos Combinacionales o Combinatorios

- Responden a los valores lógicos en las entradas, la salida está determinada exclusivamente por los valores de las entradas en ese instante.
- Si cambia la entrada, cambia la salida.
- Los valores pasados de las entradas no influyen en los valores de las salidas.

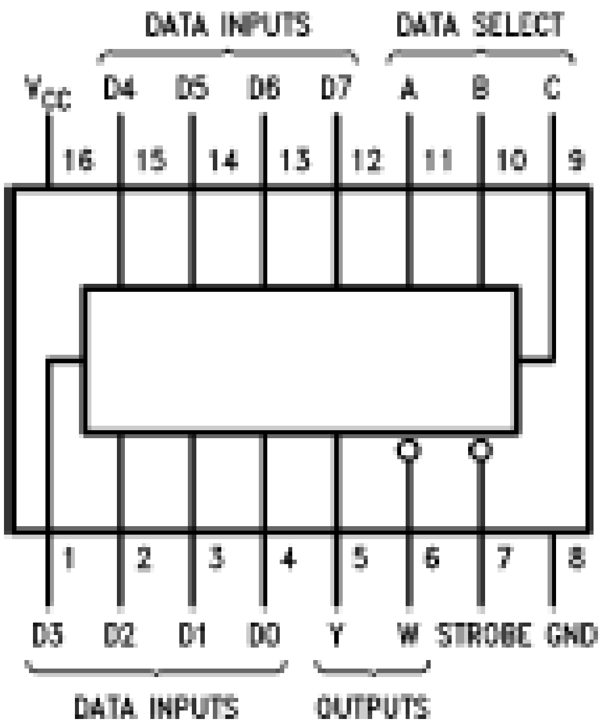
Puertas lógicas en un chip

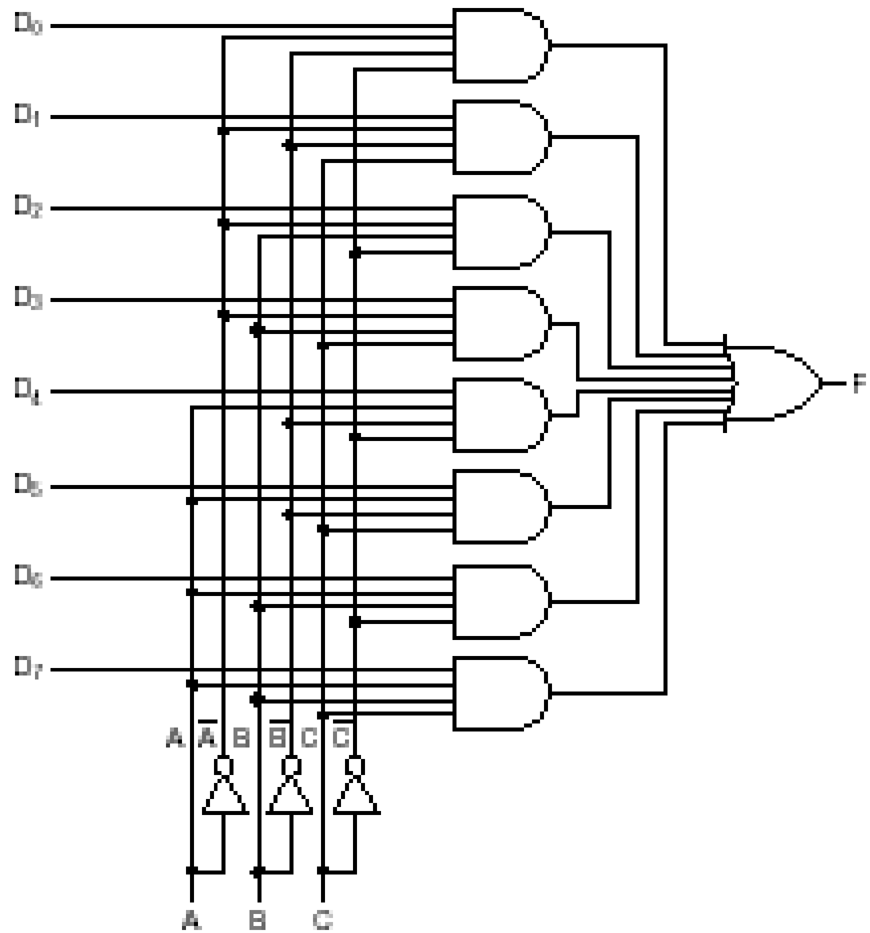


Ejemplo 1

Multiplexor de 8 entradas

- 74151



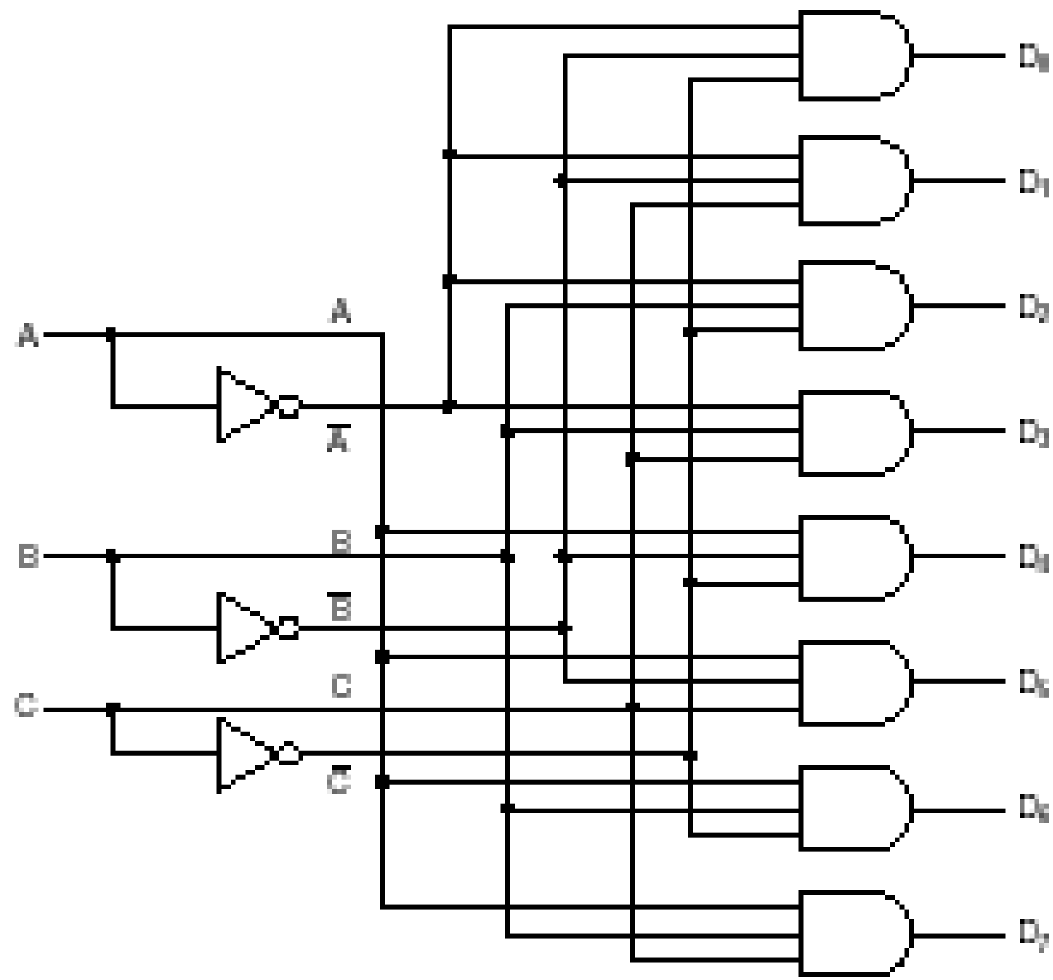


Según valor de entradas A, B y C

$F=D_x$

Ejemplo 2

Decodificador 3 a 8



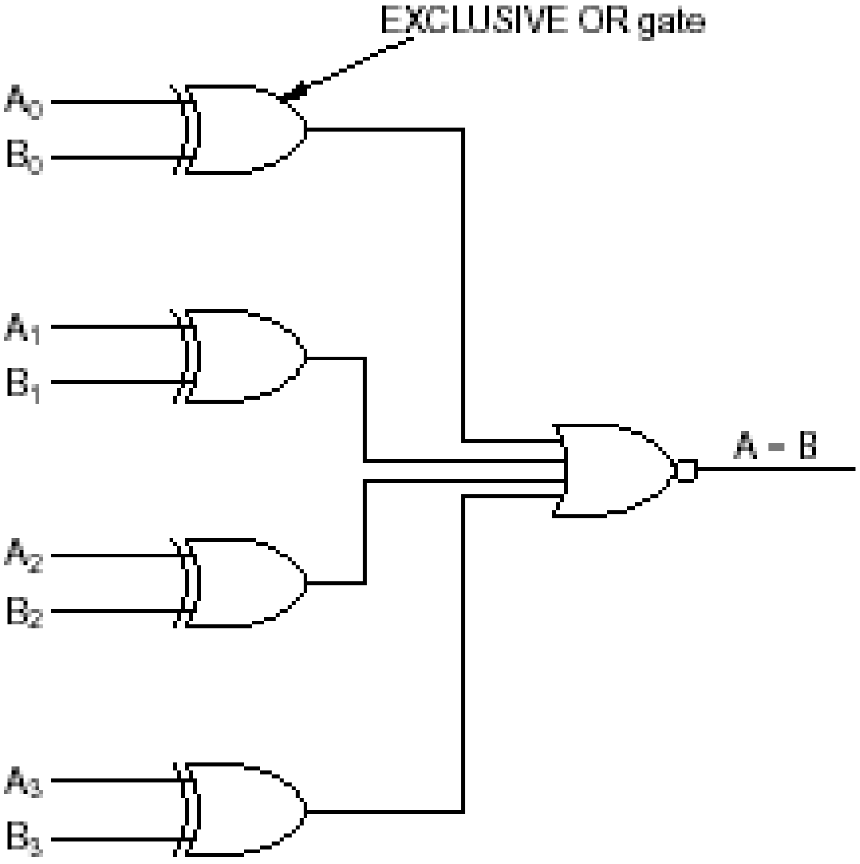
Para cada combinación de las entradas A, B y C

sólo UNA de las salidas D_x vale ‘1’

Ejemplo 3

Comparador de 4 bits

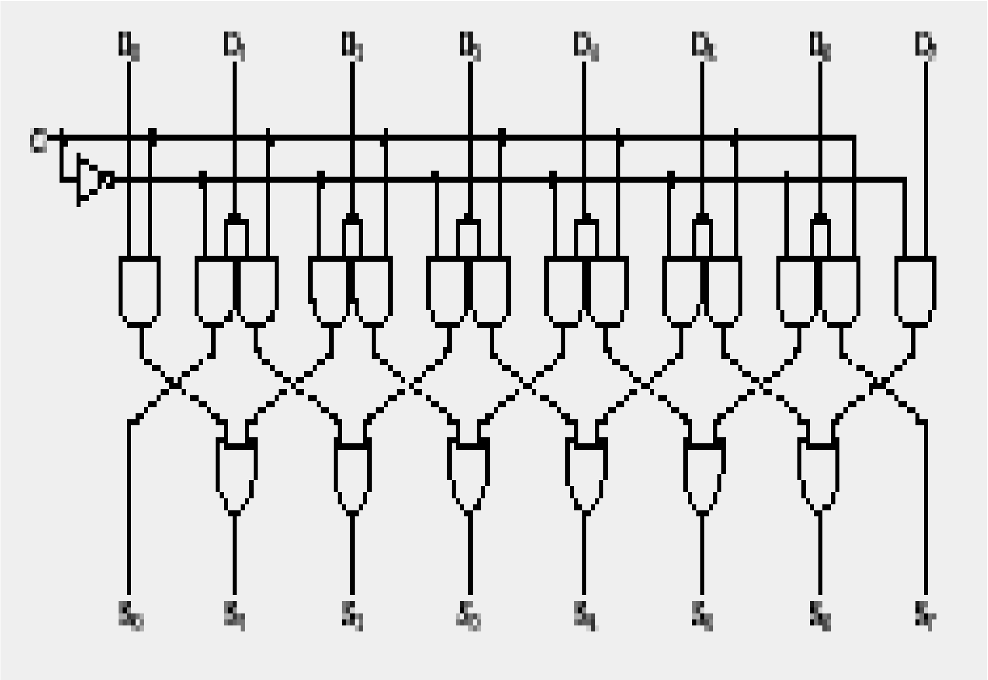
Si todos los bits A_i son iguales a los B_i la salida es '1'



Ejemplo 4

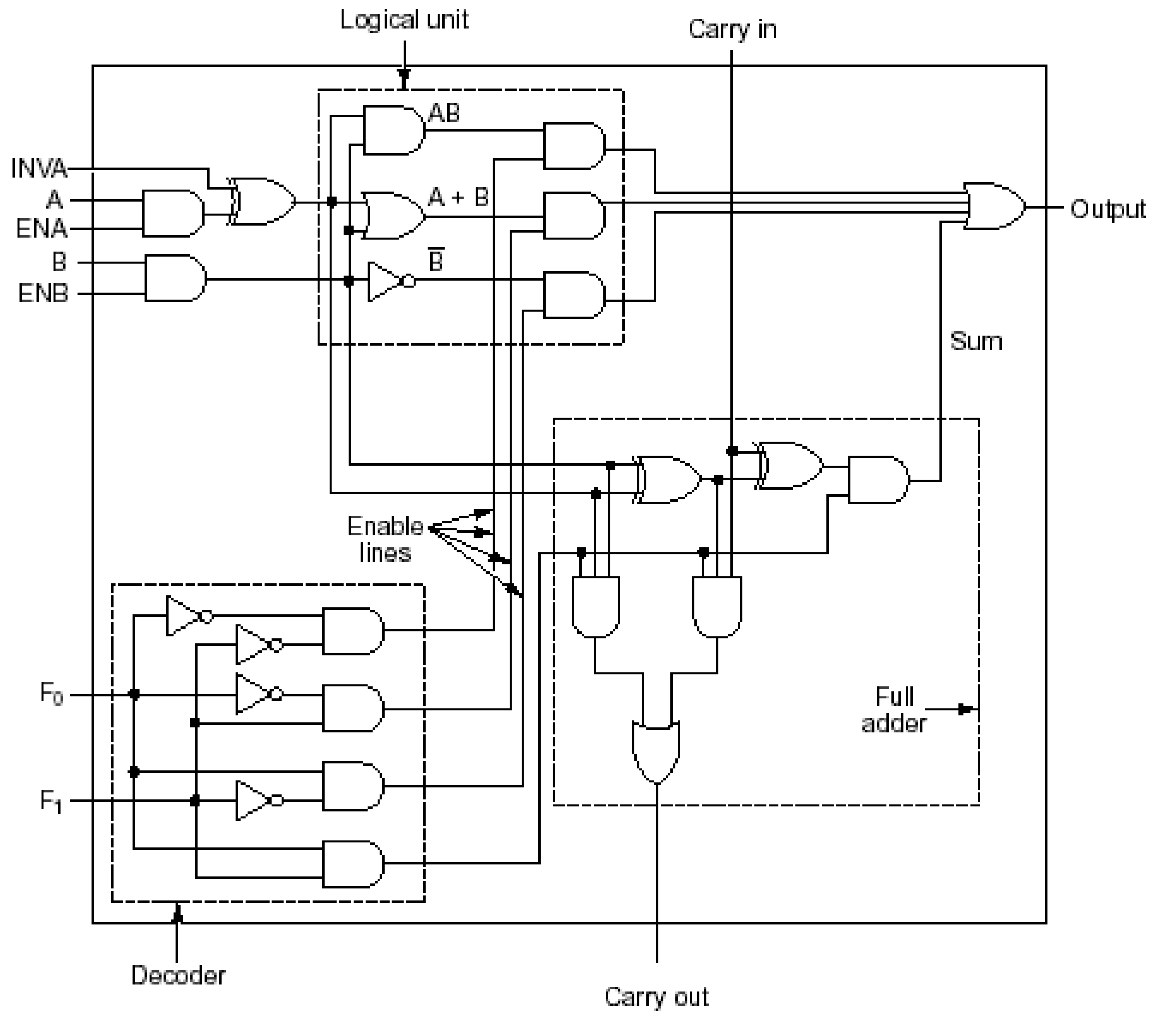
Desplazador de 1 bit

Según el valor de la entrada C se ‘correrán’ un lugar a derecha o izquierda.



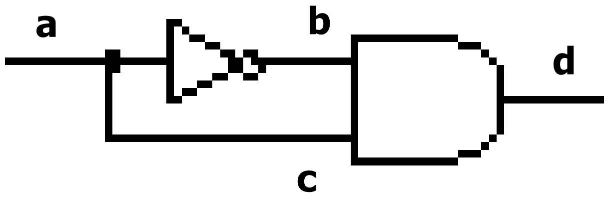
Ejemplo 5

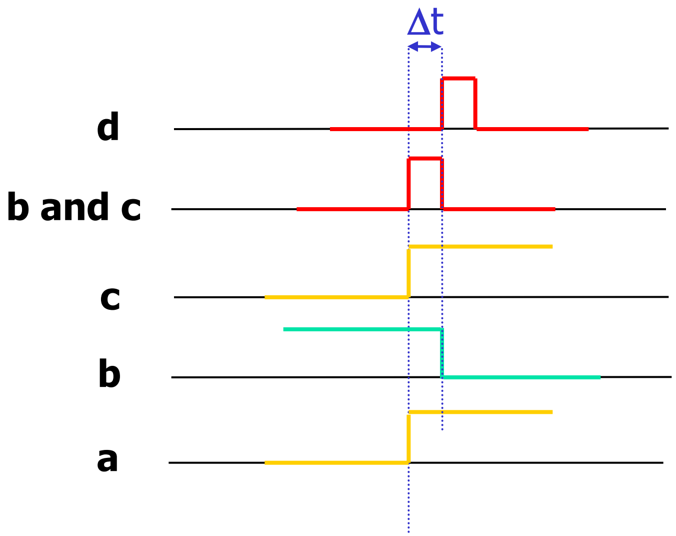
1 bit de ALU



Según F_1F_0 será la función que se realizará sobre A y B.

Respuesta temporal



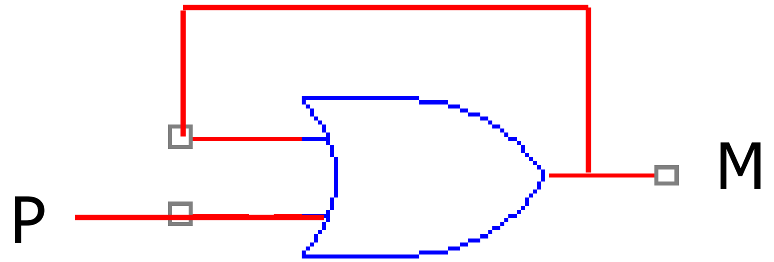


Suponemos que los retardos de compuerta Δt son iguales

Circuitos Secuenciales

- Las salidas dependen tanto de las entradas como del estado interno del circuito.
  - ¿Qué es el estado interno del circuito?
- Tienen la característica de “almacenar” valores lógicos internamente.
- Estos valores se almacenan aunque las entradas no estén.

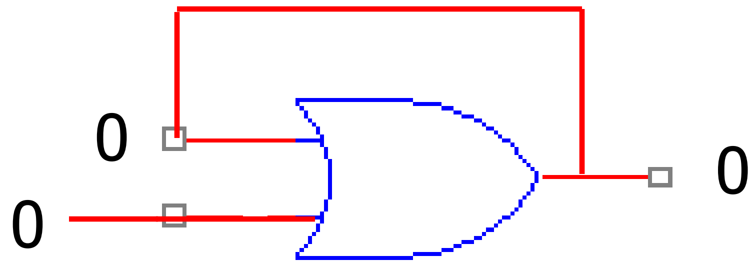
¿Cómo se almacena un valor lógico?



- La salida es también entrada
- En ningún circuito combinatorio una salida transportaba información hacia la entrada
- La ecuación lógica

$$M=M+P$$

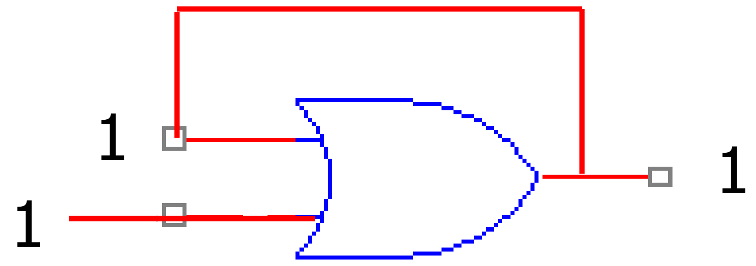
¿Cómo se ...?(2)



- Supongamos que P=0 y M=0

$$M=M+P=0+0=0$$

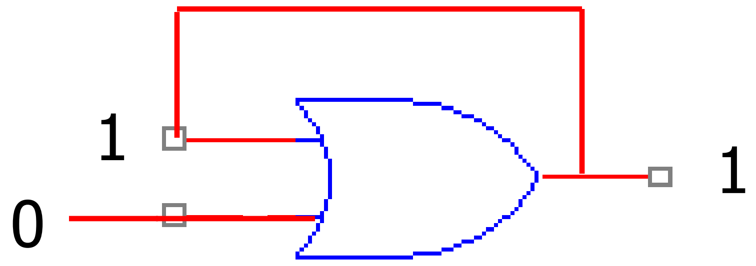
¿Cómo se ...?(3)



- Ahora P=1

$$M=M+P=1+1=1$$

¿Cómo se ...?(4)

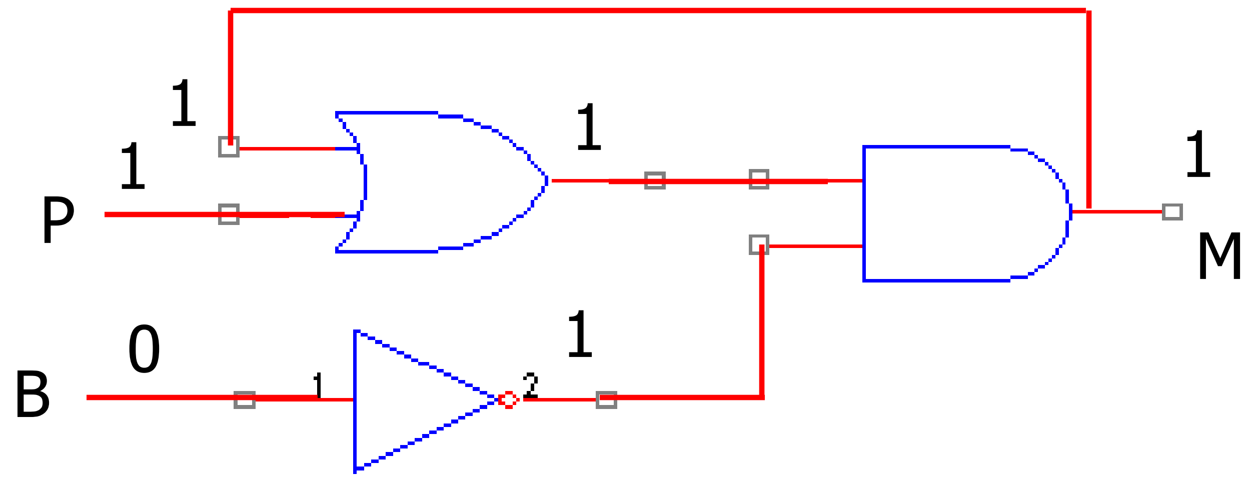


- Ahora P=0

$$M=M+P=1+0=1$$

- Una vez que la salida M toma el valor 1 no hay forma de volver a 0

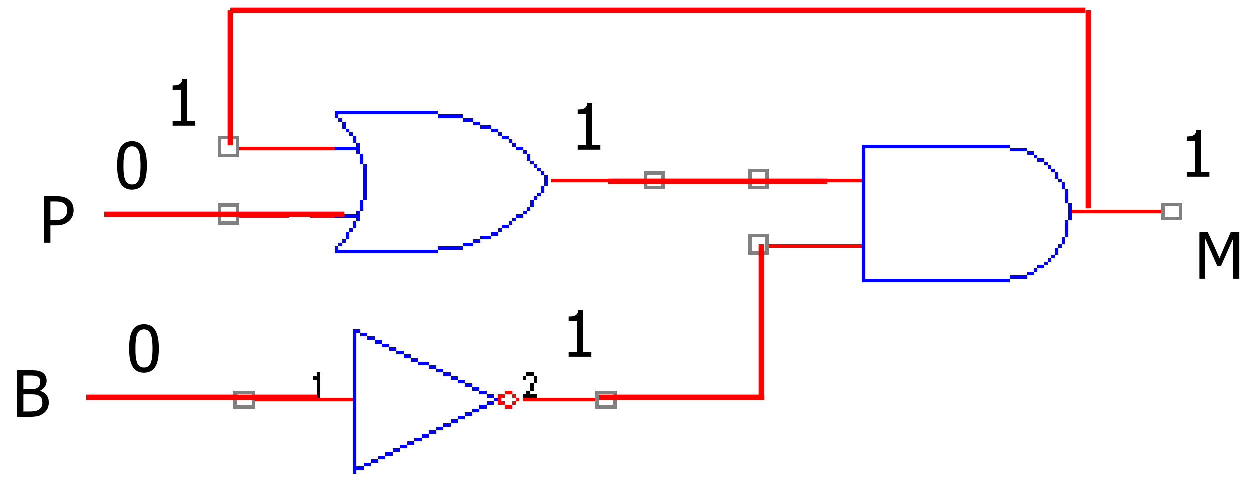
¿Cómo se ...?(5)



Ahora P=1 y B=0, M=1

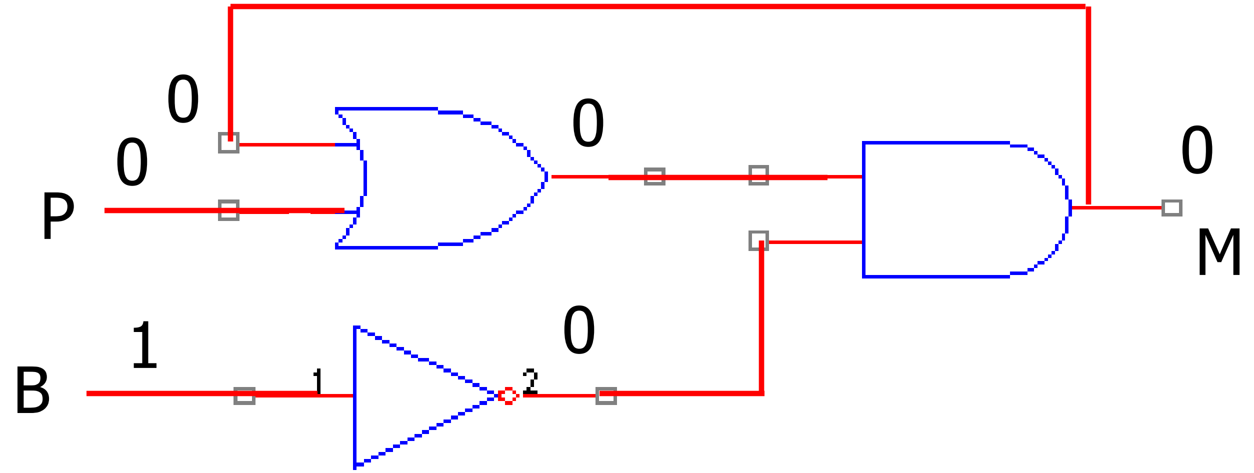
$$M=(M+P).\overline{B}$$

¿Cómo se ...?(6)



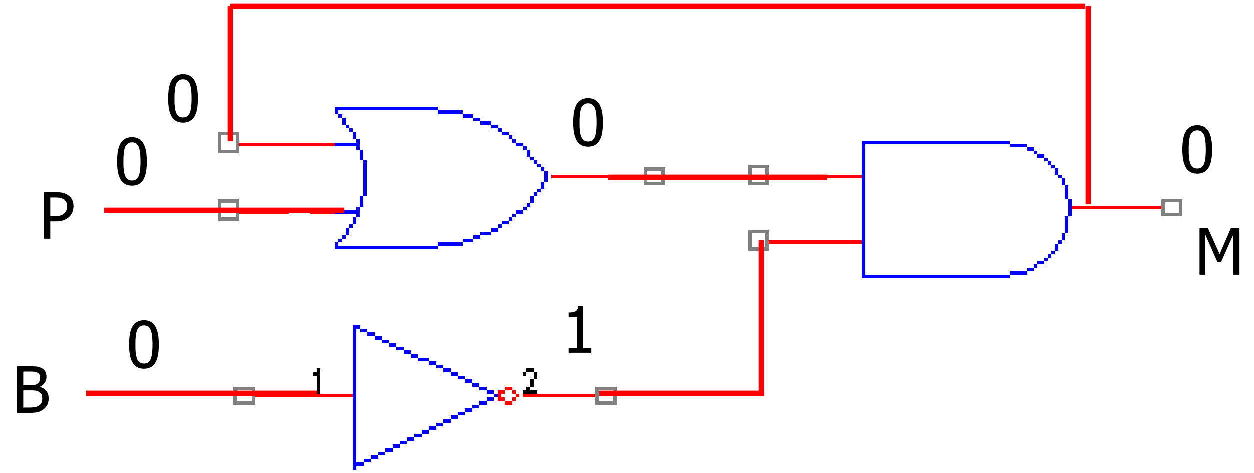
- Si ahora P=0 y B=0, M=1. Nada cambia.

¿Cómo se ...?(7)



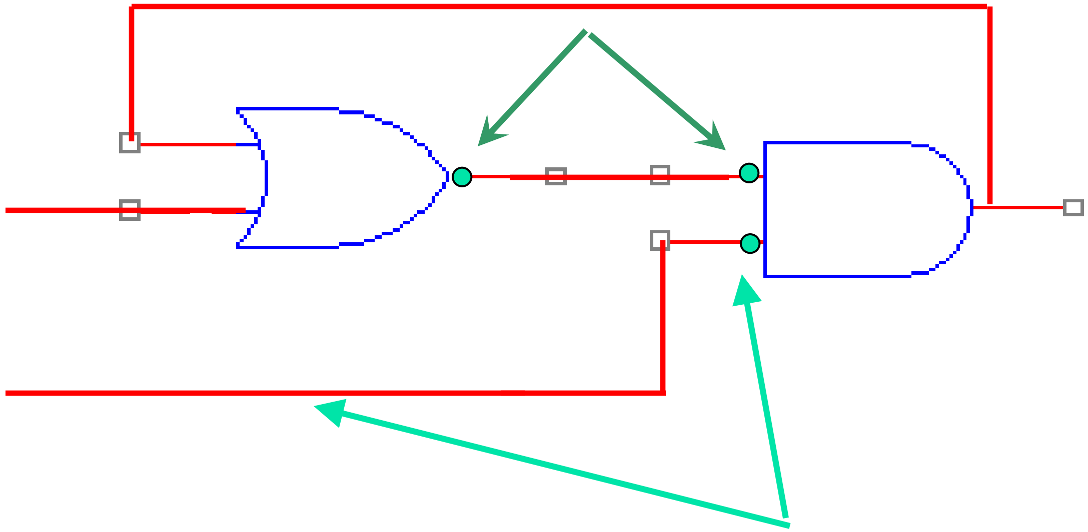
- Si ahora P=0 y B=1, M=0.

¿Cómo se ...?(8)

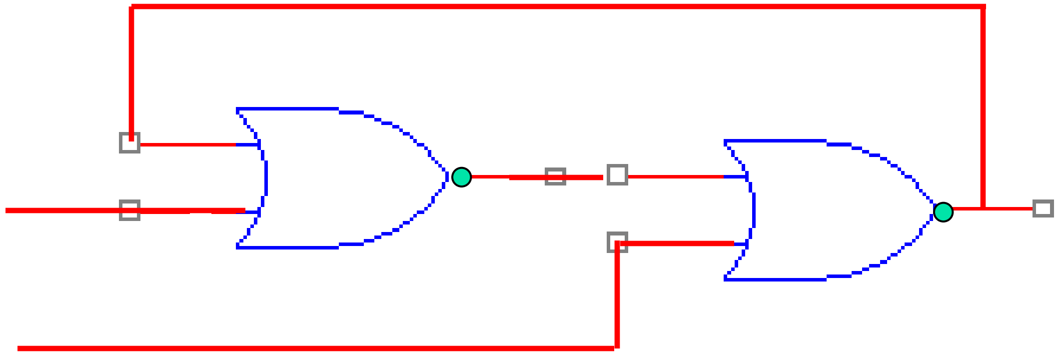


- Si ahora P=0 y B=0, M=0.
- P puede cambiar y se reflejará en M

¿Cómo se ...?(9)

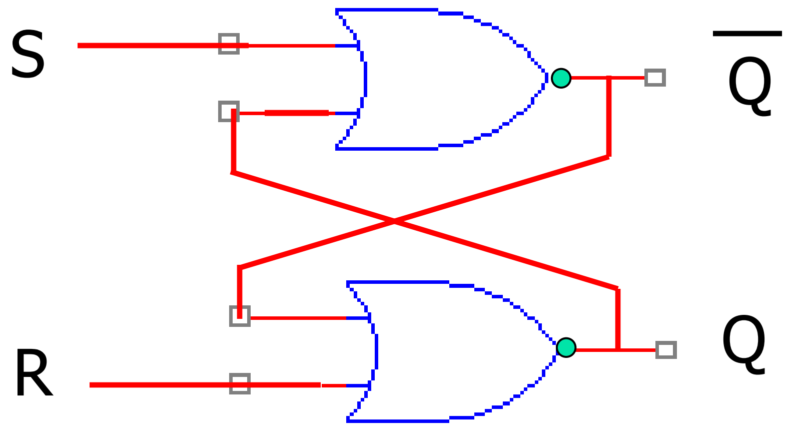


¿Cómo se ...?(10)



- Finalmente queda así

FLIP-FLOP SR

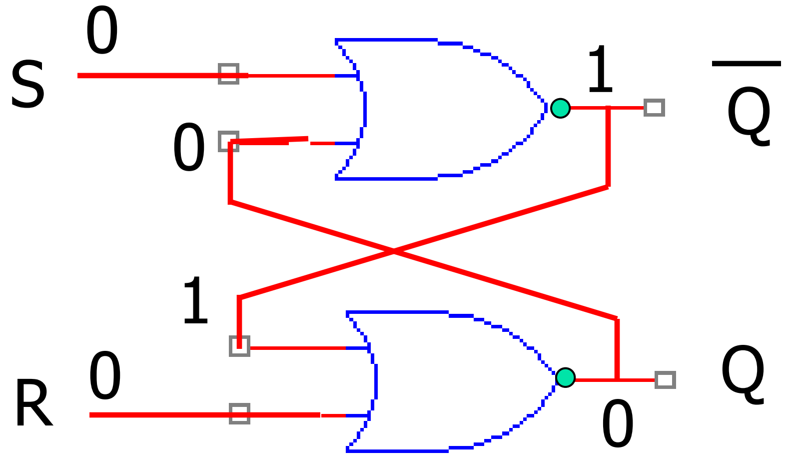


S	R	Q_{n+1}
0	0	Q_n
0	1	0
1	0	1
1	1	Prohibido

FLIP-FLOP SR(2)

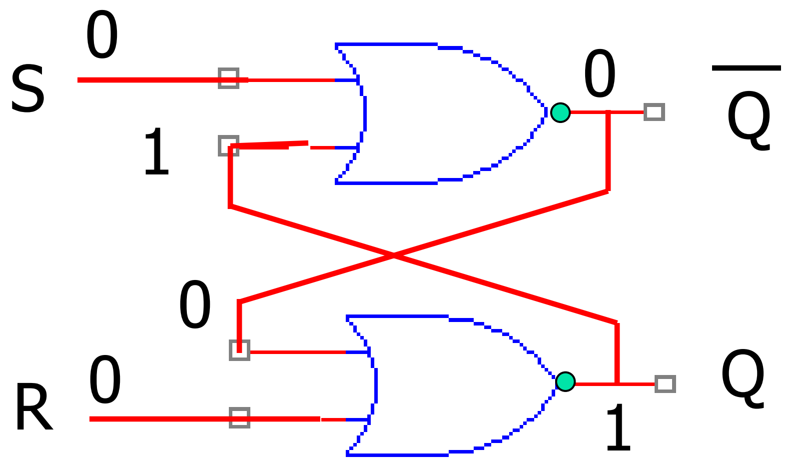
- Aparece la salida Q_{n+1}
- Q_n= salida anterior
- S = Set = poner a 1
- R = Reset = poner a 0
- Las salidas Q y $\overline{Q}$ son complementarias

FLIP-FLOP SR(3)



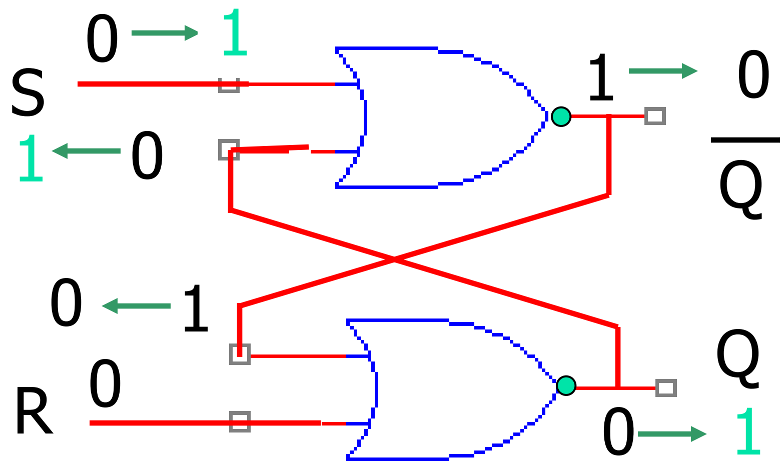
Supongamos S y R = 0 y Q = 0

FLIP-FLOP SR(4)



- Supongamos S y R = 0 y Q = 1
  - Por lo que “recuerda” cual era el estado anterior.

FLIP-FLOP SR(5)



- Si ahora S=1 y R=0

Memoria

- Se puede construir con un flip-flop una memoria de 1 bit.
- Se llama biestable porque el circuito posee sólo 2 estados posibles de funcionamiento, se queda en cada uno de ellos, salvo que las entradas provoquen un cambio.

Secuenciales - Clasificación

- Según la manera en que las salidas respondan a las señales lógicas presentes en la entrada, los biestables se clasifican en:
  - SR
  - J-K
  - D
  - T

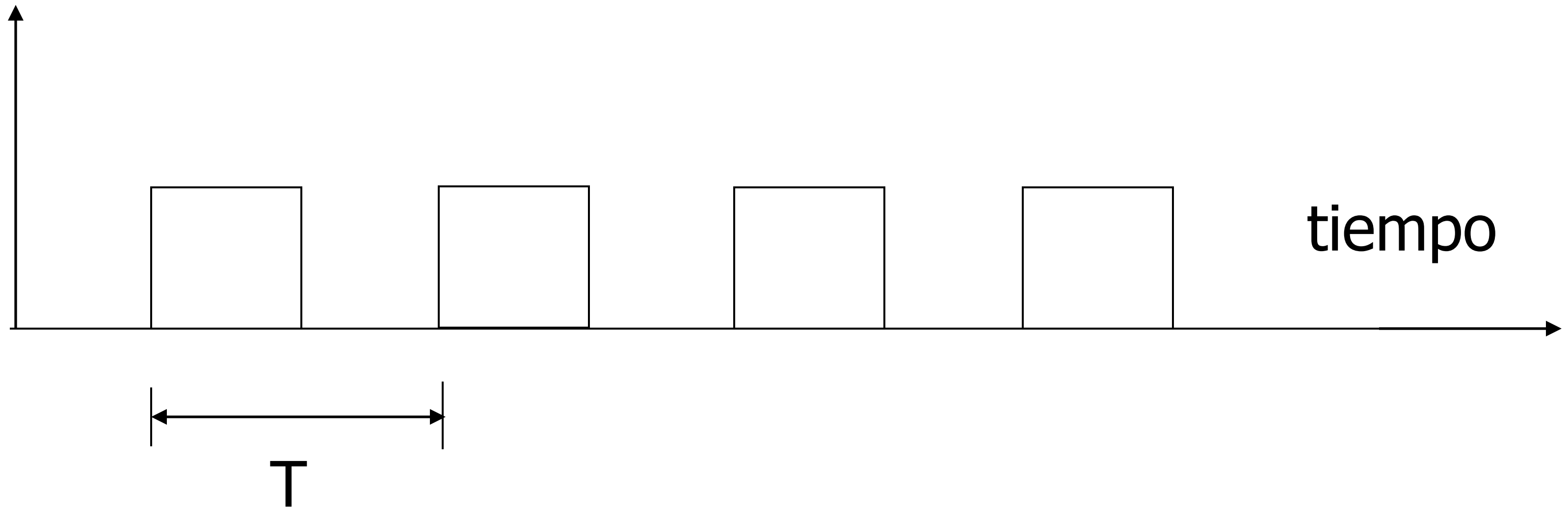
Secuenciales – Clasificación(2)

- Respecto del instante en que pueden cambiar dichas salidas, pueden ser:
  - Asincrónicos: cuando en la entrada se establece una combinación, las salidas cambiarán
  - Sincrónicos: la presencia de una entrada especial, determina “cuando” cambian las salidas acorde a las entradas

Reloj: “señal especial”

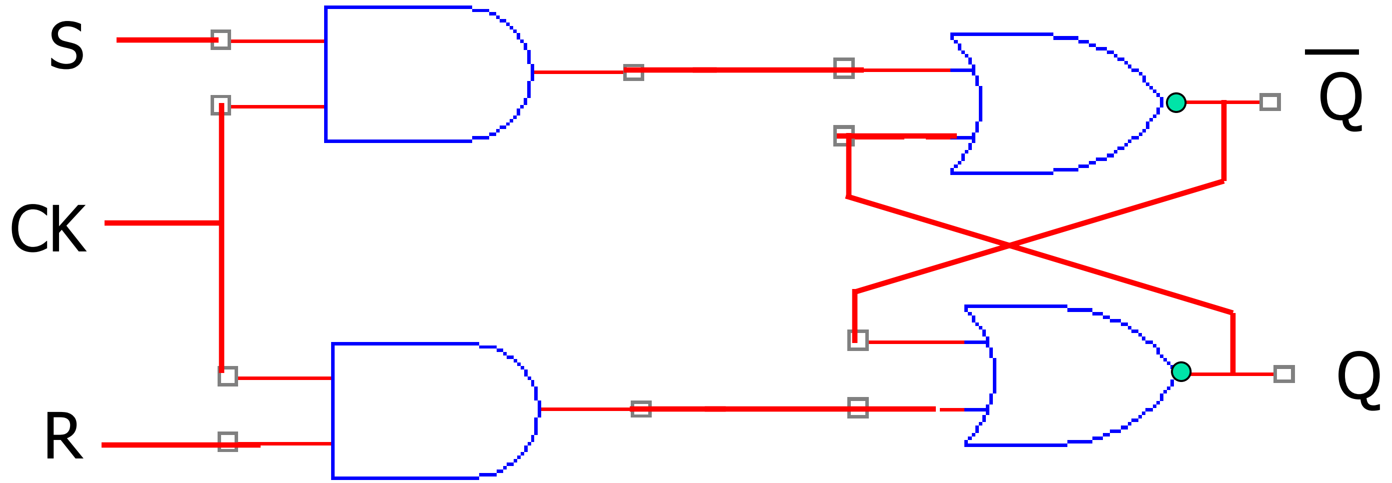
- El orden en que ocurren los sucesos es importante.
- A veces los sucesos deben ocurrir simul-taneamente.
- Reloj: es una señal de tiempo precisa que determina cuando se producen eventos.

Reloj (Clock) (CLK)



Cada tiempo T, la señal se repite

Flip-Flop SR sincrónico



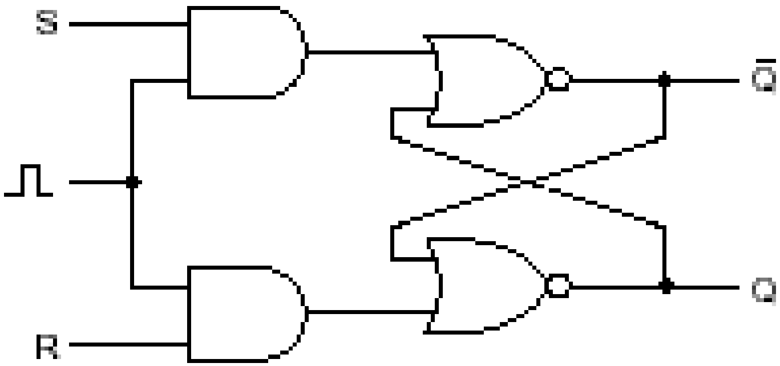
- S y R son las entradas que tendrán efecto cuando CK tome el valor 1.

Tabla de comportamiento: SR sincrónico

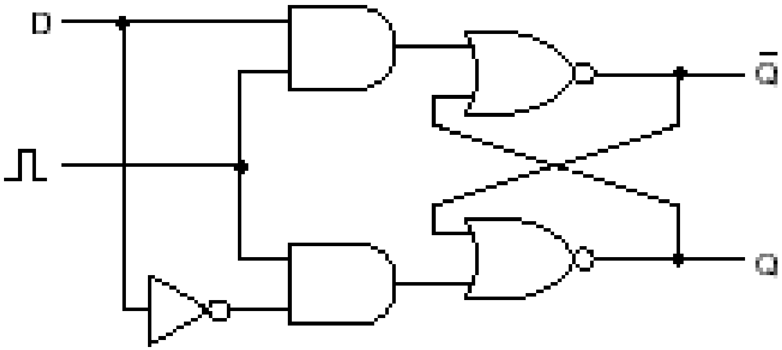
CK	S	R	Q_{n+1}
1	0	0	Q_n
1	0	1	0
1	1	0	1
1	1	1	Prohibido
0	x	x	Q_n

Flip-Flop D

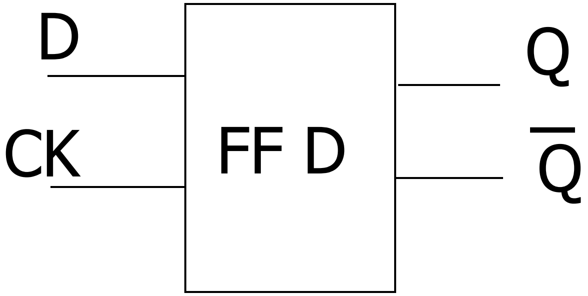
- En el FF SR hay que aplicar 2 entradas diferentes para cambiar de estado.



- El FF D permite aplicar una sola entrada para cambiar la salida.



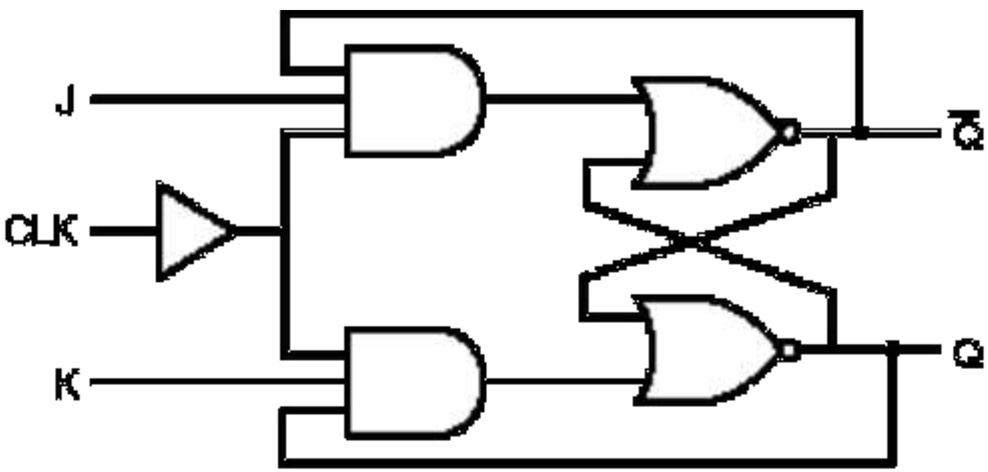
Flip-Flop D



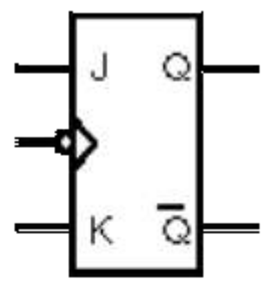
D	Q_{n+1}
0	0
1	1

con CK=1

Flip Flop J-K



Circuit

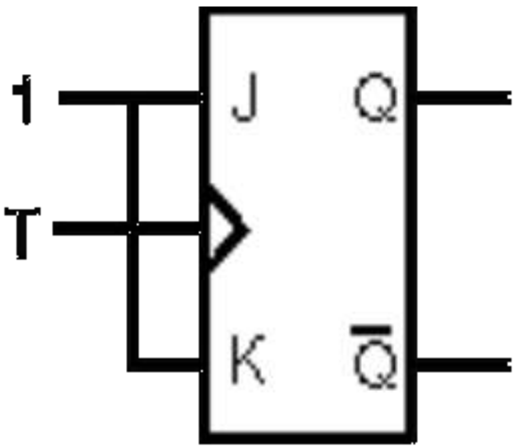


Symbol

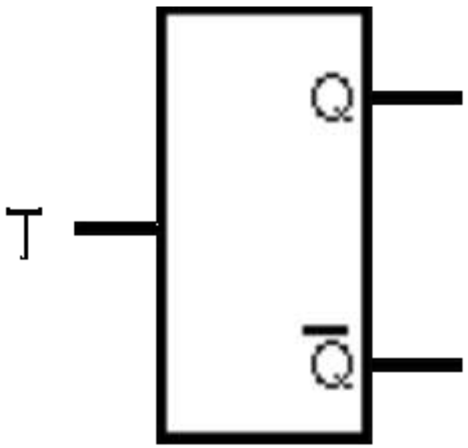
J	K	Q_{n+1}
0	0	Q_n
0	1	0
1	0	1
1	1	$\overline{Q_n}$

Flip Flop T

- La salida Q cambiará de 0 a 1 o 1 a 0 en cada pulso de la entrada T.



Circuit

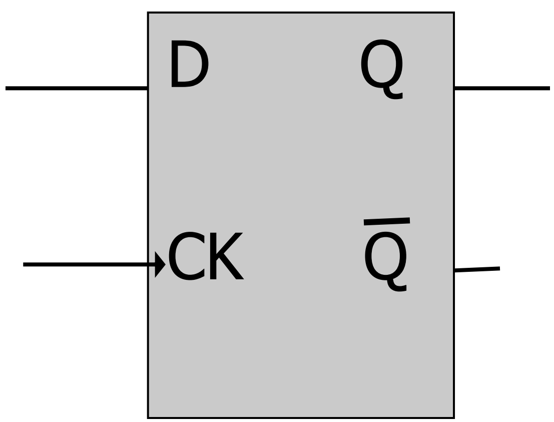


Symbol

Recordando un bit

- Con una señal (CK) se copia el valor de D en Q
- Sin esa señal, el valor de Q permanece igual

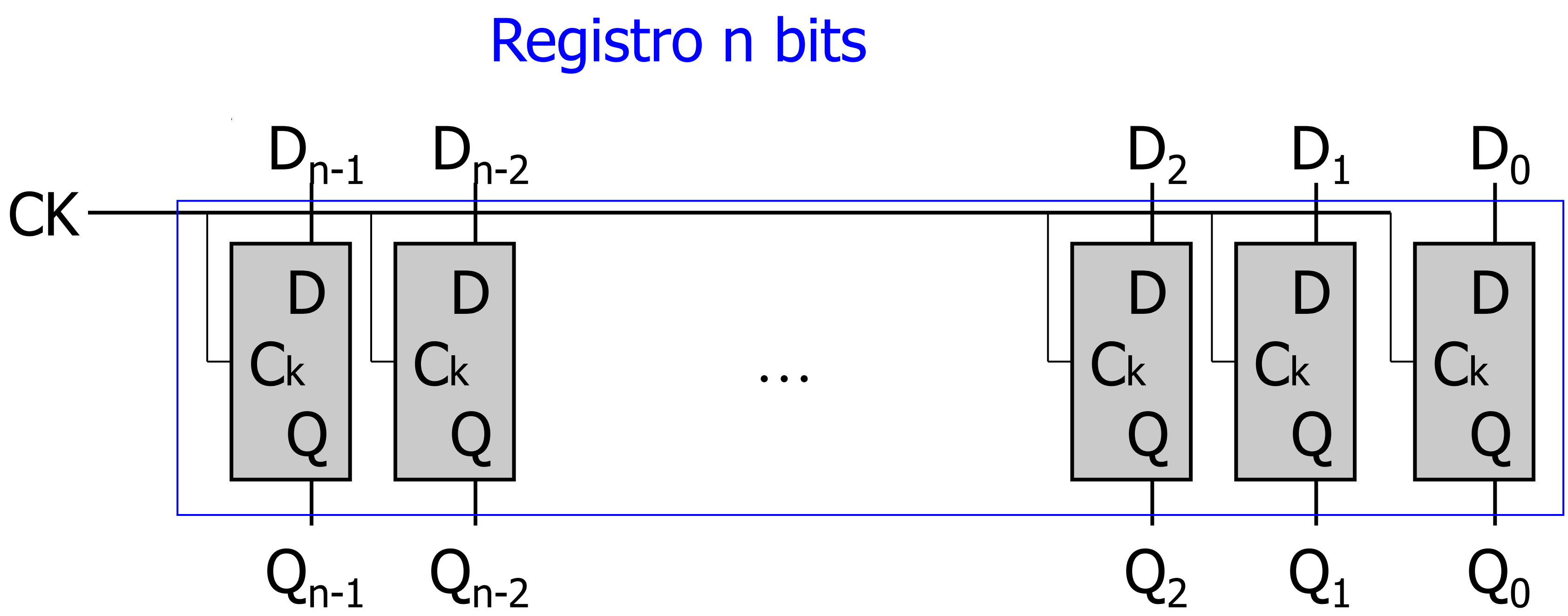
Puedo recordar un Bit



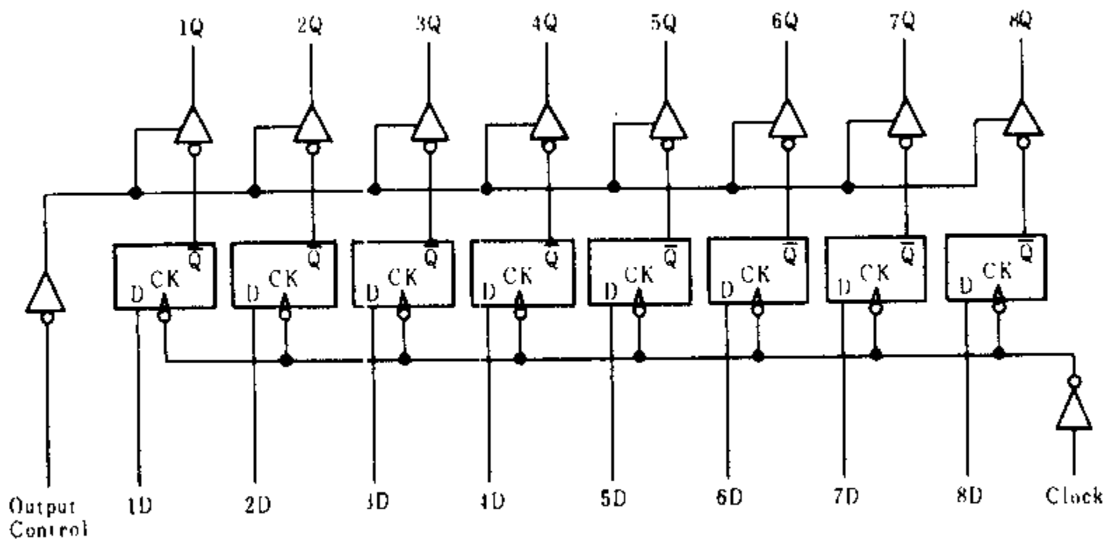
CK	D	Q
0	0	q
0	1	q
1	0	0
1	1	1

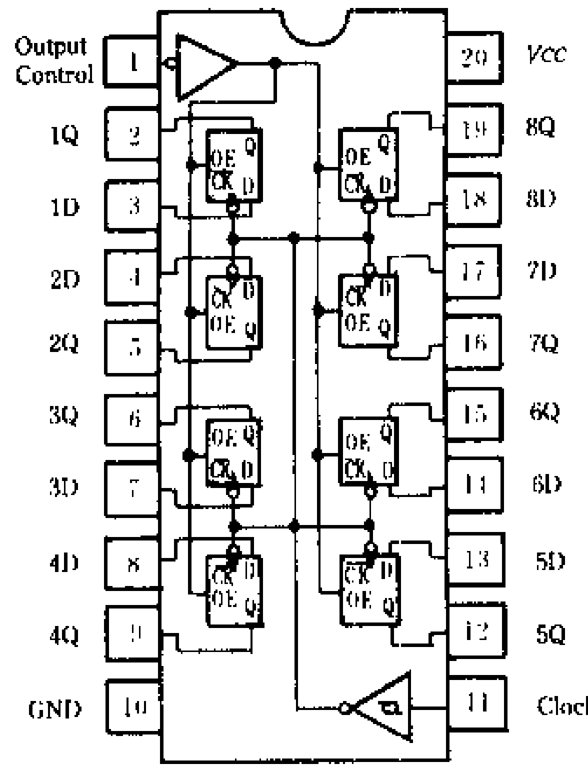
Recordando n bits

- Si CK actúa sobre n bits simultáneamente



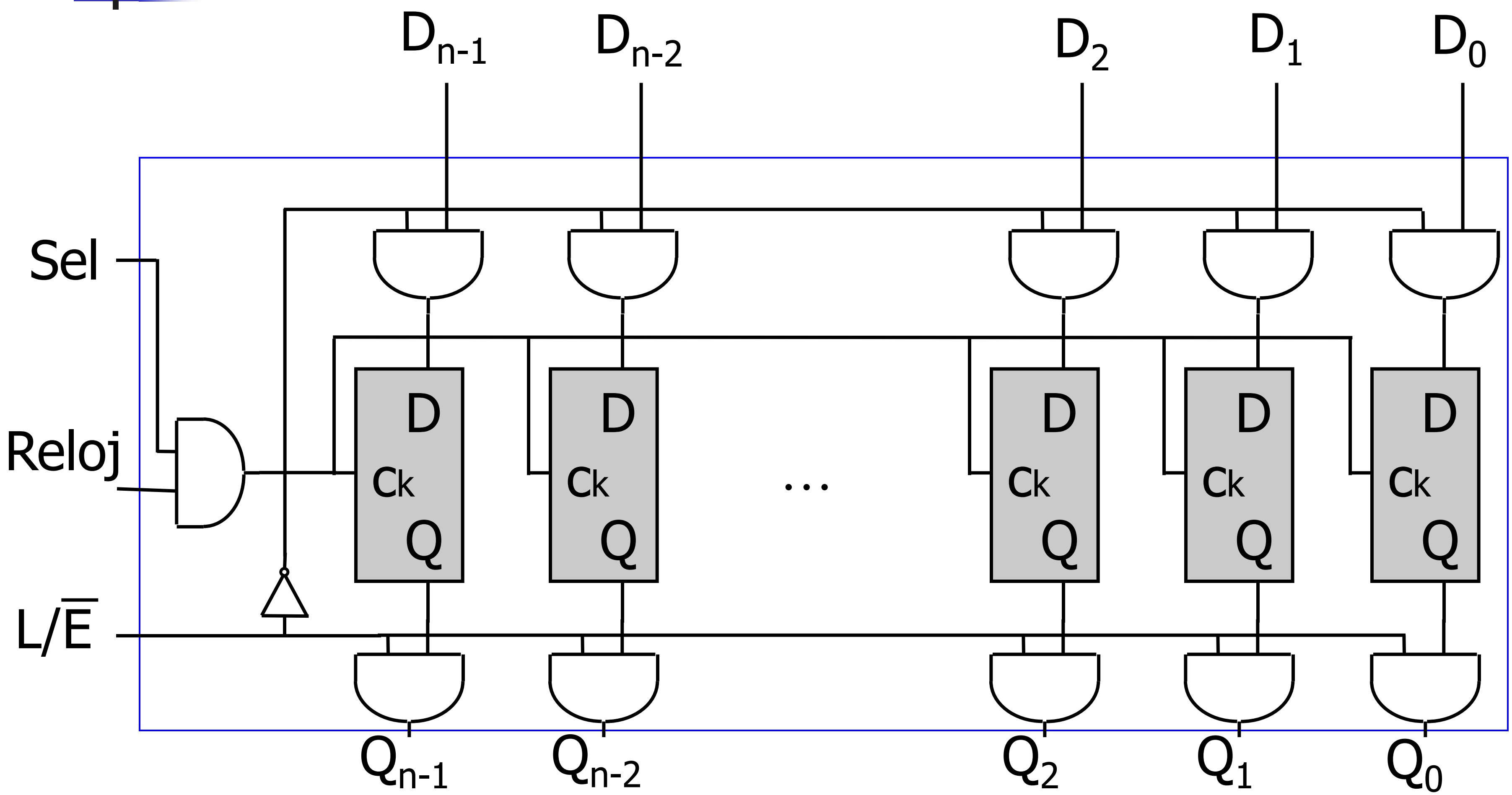
Chip con 8 FF-D (74LS374)



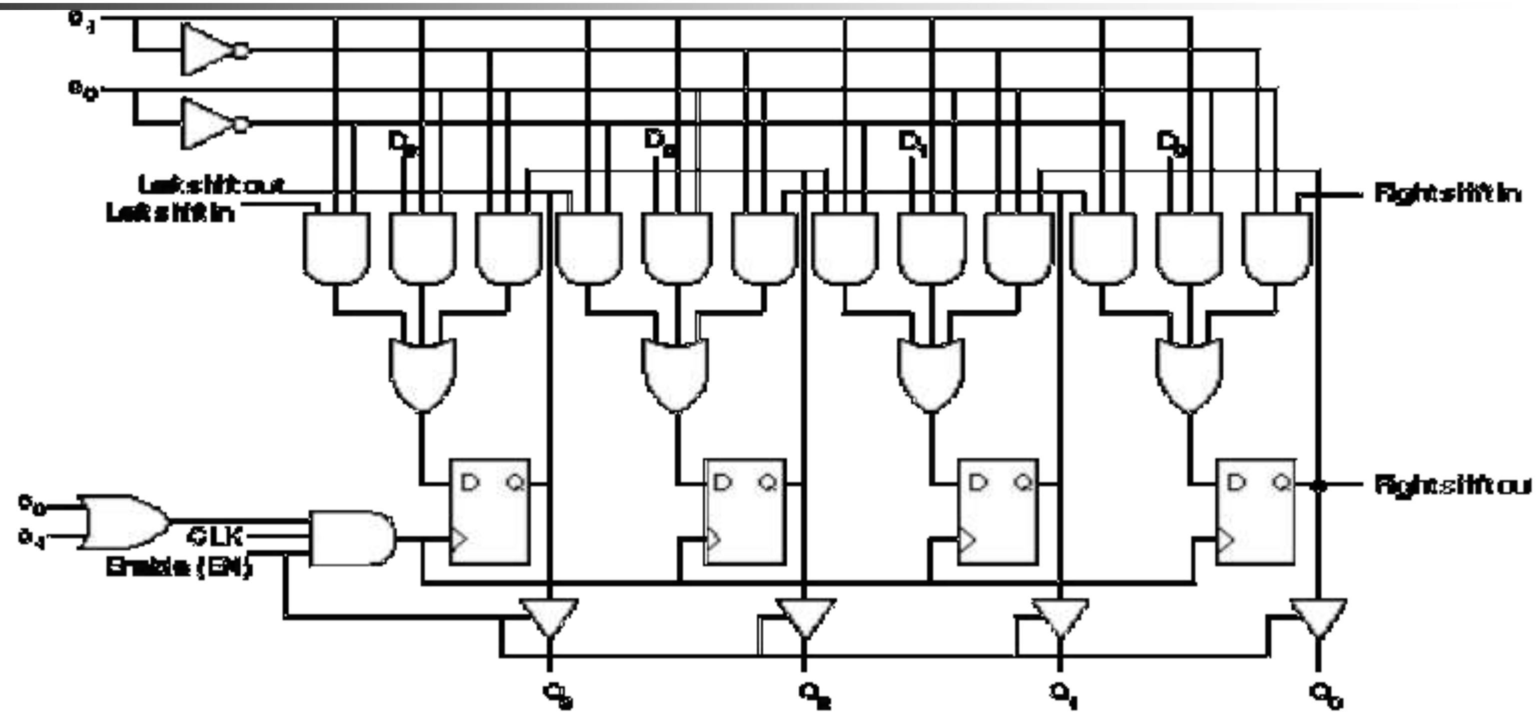


(Top View)

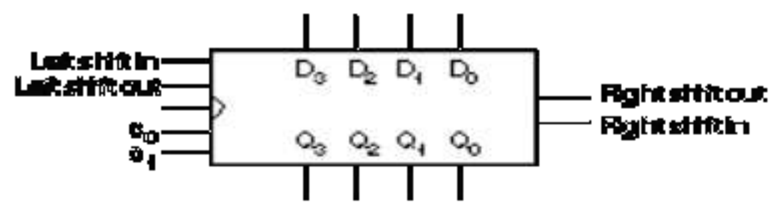
Selección y operaciones



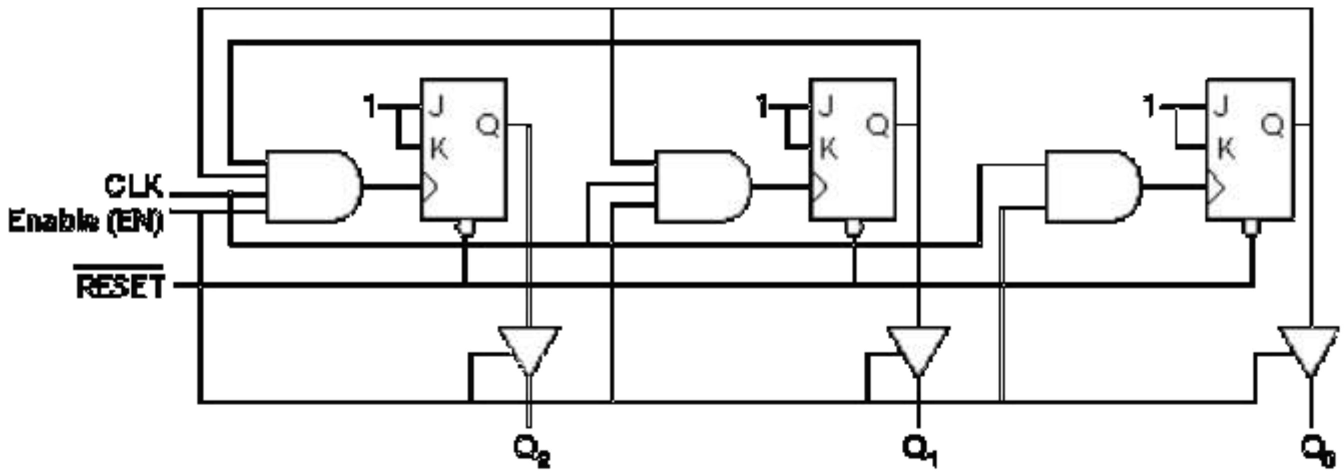
Registro con desplazamiento

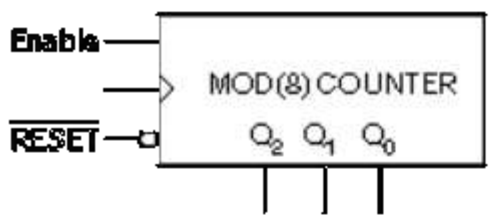


Control		Function
s_1	s_0	
0	0	No change
0	1	Shift left
1	0	Shift right
1	1	Parallel load



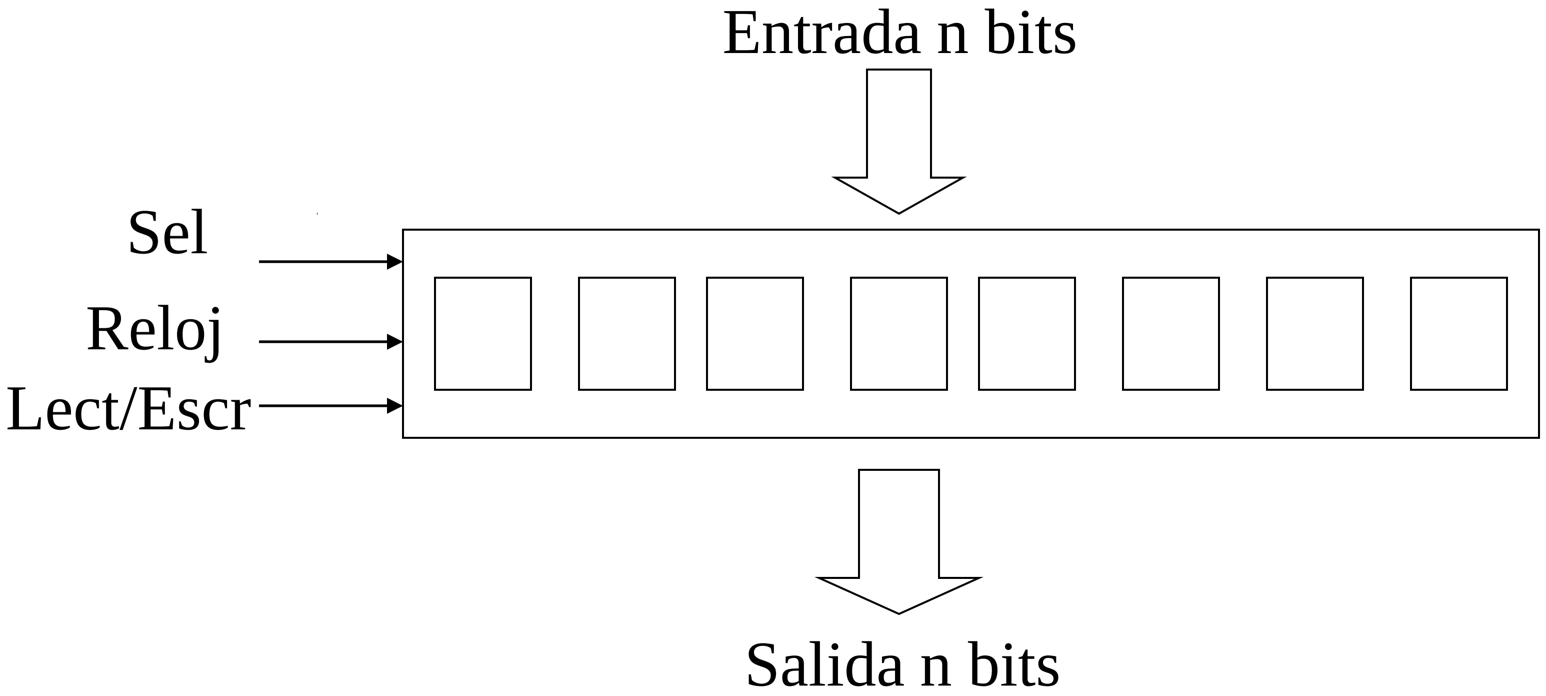
Contador módulo 8





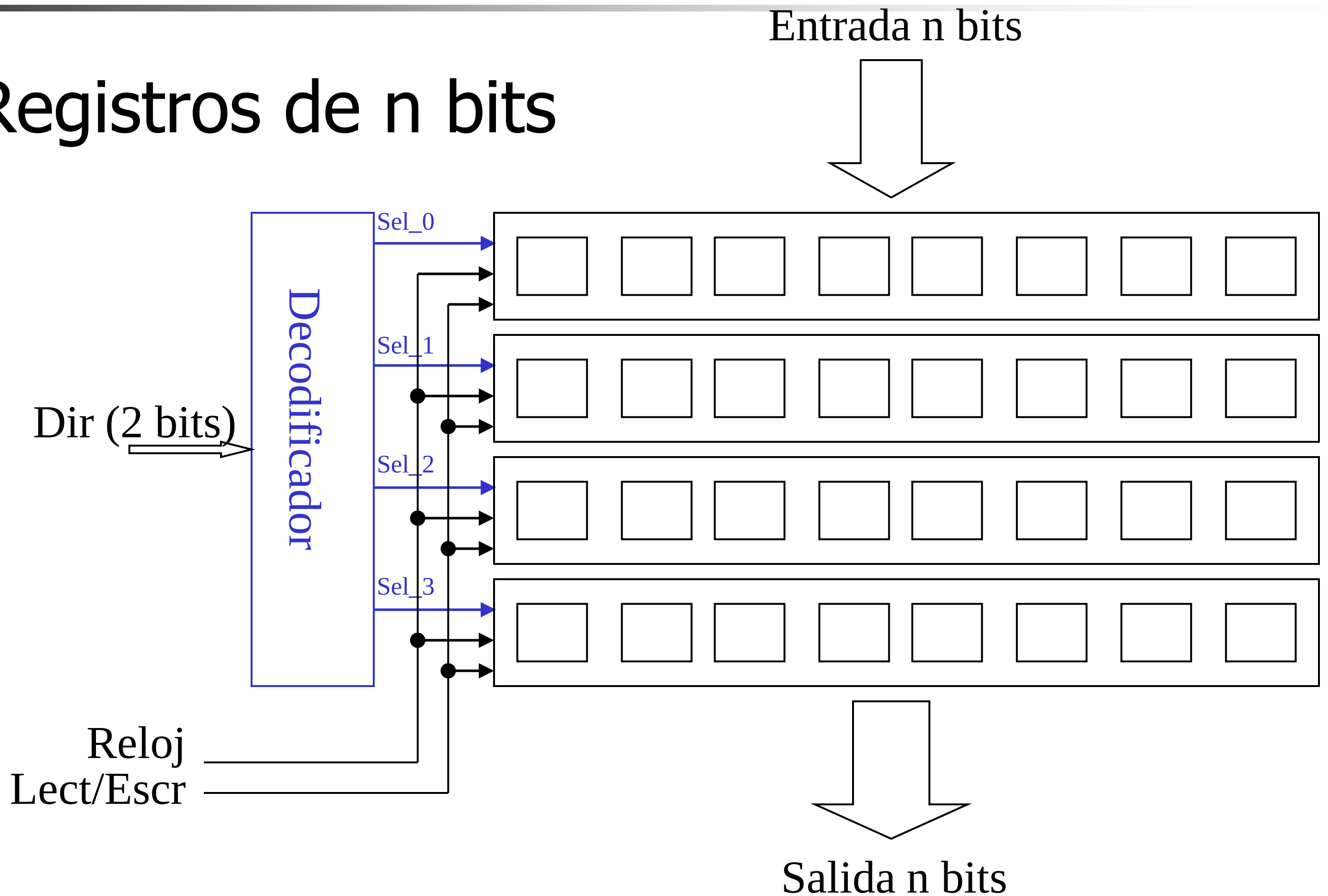
Timing behavior

Un Registro



Varios Registros

4 Registros de n bits



mayor información …

- Operaciones Lógicas
  - Apunte 3 de Cátedra
- Circuitos Secuenciales
  - Apunte 5 de Cátedra
- Apéndice A: Lógica digital (A.3., A.4.)
  - Stallings, 5ta Ed.
- Capítulo 3: Lógica digital y representación numérica
  - Apuntes COC - Ingreso 2013