Organización de Computadoras

CURSO 2024

TURNO RECURSANTES

CLASE 3 – CIRCUITOS COMBINATORIOS Y SECUENCIALES

Resumen de clase 3

- Lógica digital.
- Álgebra de Boole.
- Circuitos Lógicos Combinacionales
- Circuitos Lógicos Secuenciales
- Registros y memorias

El nivel de lógica digital

- En un circuito digital están presentes dos niveles de tensión (alto y bajo) que se asocian a 2 valores lógicos:
  - Nivel bajo = 0
  - Nivel alto = 1
- Los circuitos electrónicos que se basan en el uso de señales eléctricas de 2 niveles se llaman circuitos digitales.

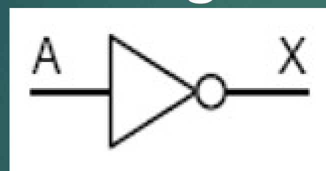
El nivel de lógica digital

- La base de los circuitos digitales son las compuertas.
- Las compuertas son dispositivos electrónicos que pueden realizar distintas funciones con estos dos valores lógicos.
- Las compuertas básicas son:
  - AND
  - OR
  - NOT
  - NAND
  - NOR

El nivel de lógica digital

- Cada compuerta tiene un símbolo para representarla, una notación para escribirla, y una lógica que implementa.
  - El símbolo es un gráfico que identifica la compuerta.

Ejemplo:



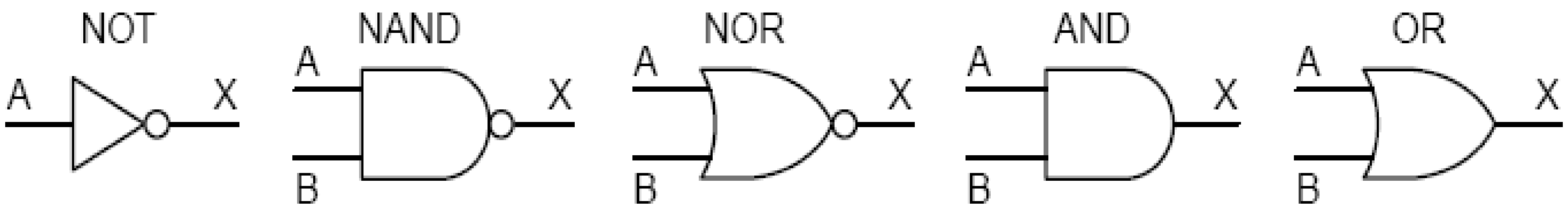
  - La notación es la forma en que se escribe la compuerta.

Ejemplo: A.B (función AND entre las variables A y B)

  - La lógica identifica la operación que implementa la compuerta. Hay distintas formas de identificar la operación, una de ellas es usando una tabla ("tabla de la verdad").

A continuación presentamos las 5 compuertas elementales que permiten implementar cualquier función combinatoria.

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

$X = \overline{A}$ $X = \overline{A \,.\, B}$ $X = \overline{A + B}$ $X = A \,.\, B$ $X = A + B$

Algebra de Boole

- Para describir el comportamiento (análisis) o hacer el diseño (síntesis) de circuitos digitales, se pueden usar herramientas matemáticas que faciliten el trabajo.
- Se requiere definir un conjunto de reglas algebraicas donde las variables y funciones sólo puedan adoptar valores binarios (0 - 1, verdadero-falso).
- Esa estructura algebraica que trata variables que pueden tener 2 valores se conoce como Algebra de Boole, en honor al matemático que la desarrolló.
- El álgebra de Boole permite analizar desde un punto de vista matemático los circuitos digitales.

Algebra de Boole

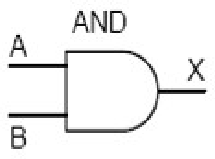
- Usa las funciones: AND, OR y NEGACIÓN.
- La tabla siguiente contiene las propiedades básicas del álgebra de Boole, donde:
  - La función lógica AND se escribe como un . (PUNTO)
    - Ejemplo: A.B
  - La función lógica OR se escribe con un + (operador de suma)
    - Ejemplo: A+B
  - La función lógica NEGACIÓN se escribe como una raya horizontal encima de una variable.
    - Ejemplo: $\overline{A}$

Propiedades básicas del álgebra booleana

	AND	OR
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	(A.B).C=A.(B.C)	(A+B)+C=A+(B+C)
Distributiva	A+(B.C) = (A+B).(A+C)	A.(B+C) = A.B+A.C
Absorción	A.(A+B)=A	A+(A.B)=A
De Morgan	$\overline{A.B}=\overline{A}+\overline{B}$	$\overline{A+B}=\overline{A}.\overline{B}$

Propiedades básicas del álgebra de booleana – Ejemplo

- Ejemplo: Identidad 1 . A = A
  - Considerando la tabla de la verdad de la AND tenemos:



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

X = A . B

si: B = 1

entonces : X = 0 si A = 0

X = 1 si A = 1

es decir: X = A

Aplicación Leyes de De Morgan

- Una de las propiedades más importantes es la ley de De Morgan, porque es útil para resolver problemas.

$$\overline{A . B} = \overline{A} + \overline{B} \quad y \quad \overline{A + B} = \overline{A} . \overline{B}$$

Ejemplo: construir una NOT con NAND

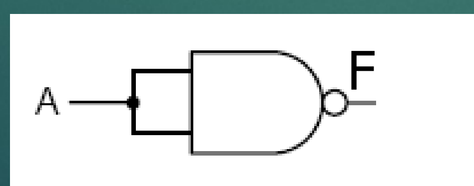
Por definición de NAND : $F = \overline{A . B}$

si hacemos: B = A (unimos la entrada A a la entrada B)

Entonces: $F = \overline{A . A}$

por de Morgan: $F = \overline{A} + \overline{A} = \overline{A}$

Quedando:



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

Conclusión: si una compuerta NAND tiene las 2 entradas unidas, se comporta como un INVERSOR (NOT)

Aplicación Leyes de De Morgan

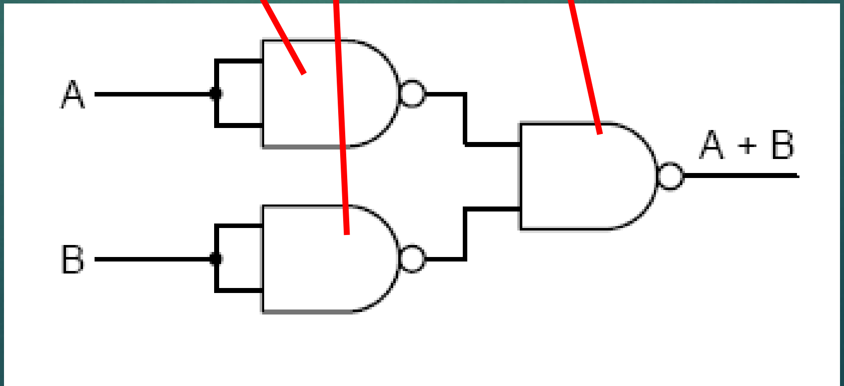
- Ejemplo: construir un OR con NAND

Por definición de OR: $F = A+B = \overline{\overline{A+B}}$

aplicando De Morgan : $\overline{A + B} = \overline{A} \,.\, \overline{B}$

por lo tanto: $F = \overline{\overline{A} \,.\, \overline{B}}$

Del ejemplo anterior: $\overline{A}$ y $\overline{B}$

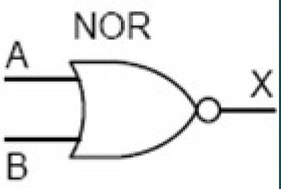


Circuitos Combinacionales o Combinatorios

- Un circuito combinatorio es aquel en el que la (o las) salida(o salidas) sólo depende (o dependen) de los valores presentes en las entradas.
- Si cambian las entradas, pueden cambiar las salidas.
- Los valores previos de las entradas no influyen en los valores de las salidas, solo importan los actuales.

Circuitos combinatorios - Tabla de la verdad

- El comportamiento de un circuito combinatorio se puede describir mediante una tabla binaria denominada Tabla de la verdad.
- Es una tabla donde:
  - En las columnas se representan las entrada(s) y salida(s).
  - En las filas se indican las diferentes combinaciones de las entradas y los valores que toma la salida para cada combinación de las entradas.
- Es obligatorio que se indiquen todas las combinaciones de las entradas.



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

Circuitos combinatorios - Tabla de la verdad

- 2 variables binarias pueden tener 4 posibles combinaciones:

  0 0 0 1 1 0 1 1

- En general, n variables binarias pueden tener $\boxed{2^n}$ combinaciones distintas
- Quiere decir que si una función depende de n entradas, la función puede describirse totalmente con una tabla de 2^n renglones o filas (2^n combinaciones distintas), y n+1 columnas
- Para cada fila (o combinación de las entradas) se debe definir el valor de la función (0 o 1).

Diseño o síntesis de circuitos combinatorios

- El diseño o síntesis es el proceso de obtención de un circuito combinatorio que cumpla las reglas para las que es diseñado.
- Hay varias formas de diseñar o sintetizar circuitos combinatorios.
- Una estrategia (sistemática) se basa en el uso de la tabla de la verdad.

Síntesis de circuitos combinatorios

- Supongamos que tenemos que diseñar un circuito combinatorio como el siguiente:

- Para el proceso de síntesis usando la tabla de la verdad debemos realizar los siguientes pasos.

Síntesis de circuitos combinatorios

- 1) Escribir la tabla de verdad para la función a diseñar (deben estar todas las posibles combinaciones de las entradas y definidos los valores de la salida para cada combinación).
- 2) “Unir” con AND cada término (combinación de las entradas) que tiene un 1 en la columna de la salida con:
  - las entradas en 1 sin invertir, y
  - las entradas en 0 invertidas .
- 3) “Sumar” con OR todas las AND.

Síntesis de circuitos combinatorios

Problema:

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 ≥ 2 y M=0 en otro caso.

Síntesis de circuitos combinatorios

1°: Construir la tabla de la 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

1er

2do

3er

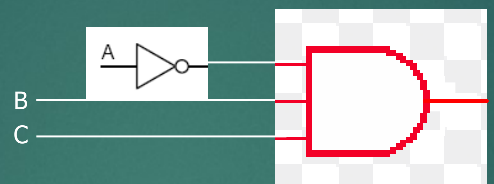
4to

Síntesis de circuitos combinatorios

2°: Unir con AND todas las filas con la función de salida en 1

Términos en 1:

1er término: $\overline{A}.B.C$



2do término: $A.\overline{B}.C$

3er término: $A.B.\overline{C}$

4to término: $A.B.C$

Síntesis de circuitos combinatorios

3°: Sumar con OR todas las AND

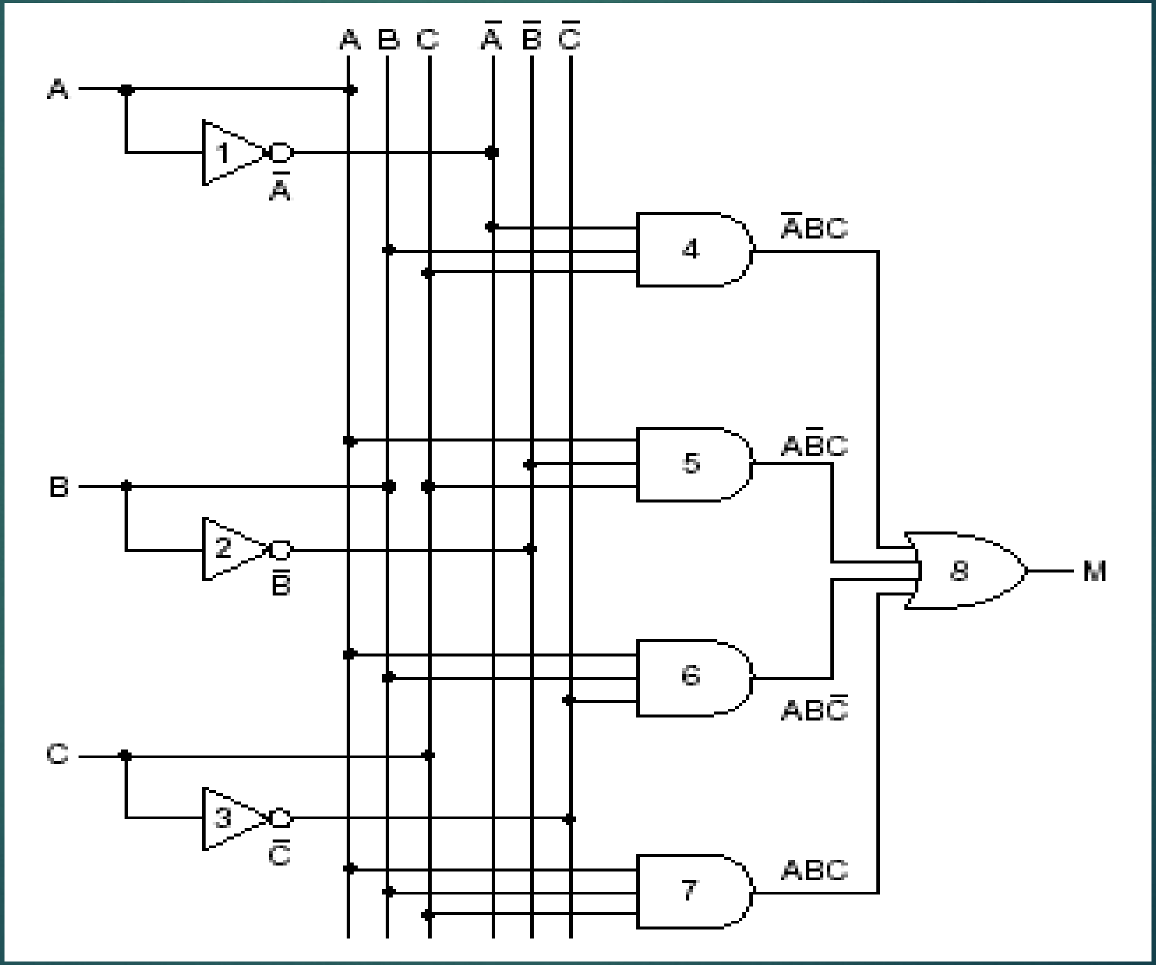
$$M = \overline{A}.B.C + A.\overline{B}.C + A.B.\overline{C} + A.B.C$$

Observar:

- Hay tantos términos como 1s en la tabla
- Cada término vale 1 para una única combinación de A, B y C
- En cada término, las variables que valen 0 en la tabla, se deben negar (es decir, invertir).

Síntesis – Resultado final

$$M = \bar{A}.B.C + A.\bar{B}.C + A.B.\bar{C} + A.B.C$$



Síntesis de circuitos combinatorios

Problema:
dada la siguiente tabla de la verdad, identificar la función.

Esta función se llama OR Exclusivo. Es una compuerta nueva que tiene el siguiente símbolo:

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

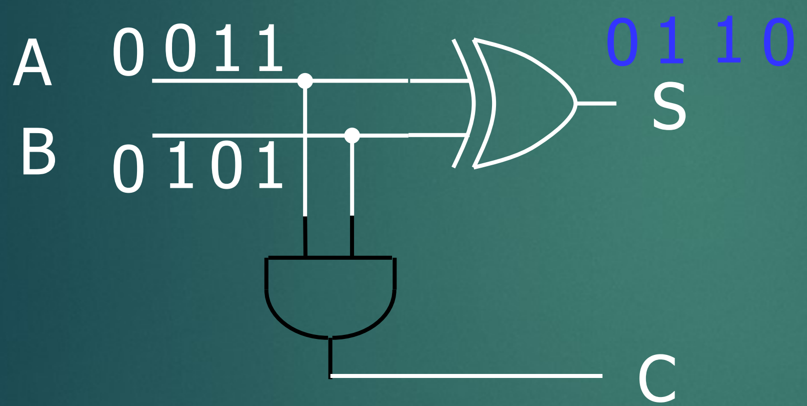
Función $\mathbf{M = \overline{A}B + A\overline{B}}$ $\Rightarrow$ $\boxed{\mathbf{M = A\ XOR\ B}}$

Propiedades de las compuertas AND, OR y XOR

- 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 de una variable A con 1 invierte el valor de la variable.
- Hacer el XOR de una variable A con 0 deja el valor de la variable como estaba.

Otros circuitos combinatorios

- Ejemplo: analizar el siguiente circuito combinatorio

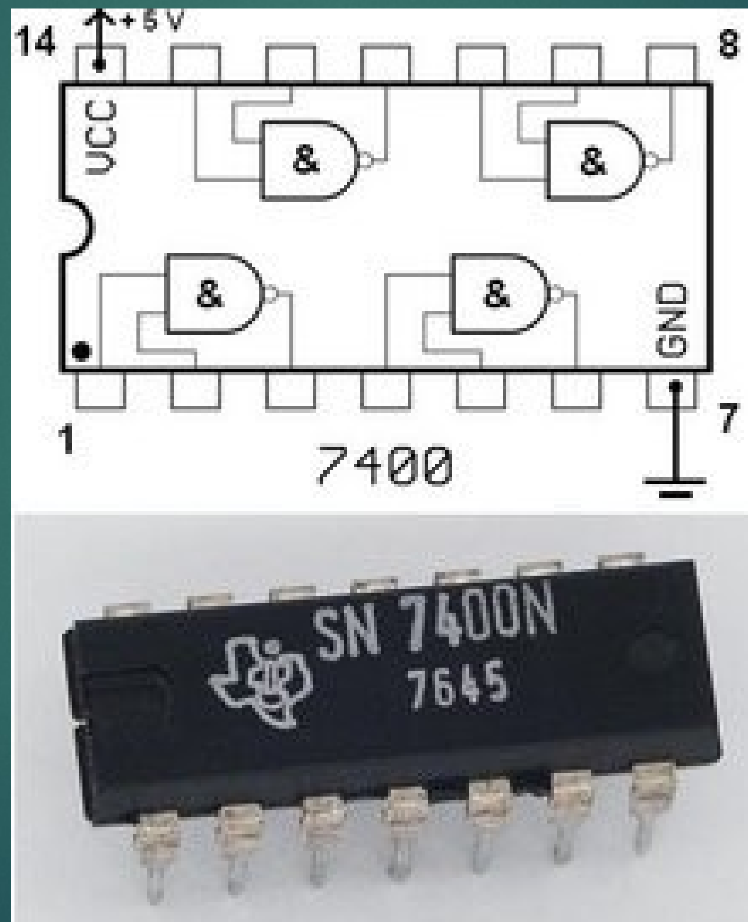


A B	S	C
0 0	0	0
0 1	1	0
1 0	1	0
1 1	0	1

- Resultado:
  - **S** representa la suma aritmética de 1 bit
  - **C** es el acarreo
  - Circuito semisumador (Half adder)

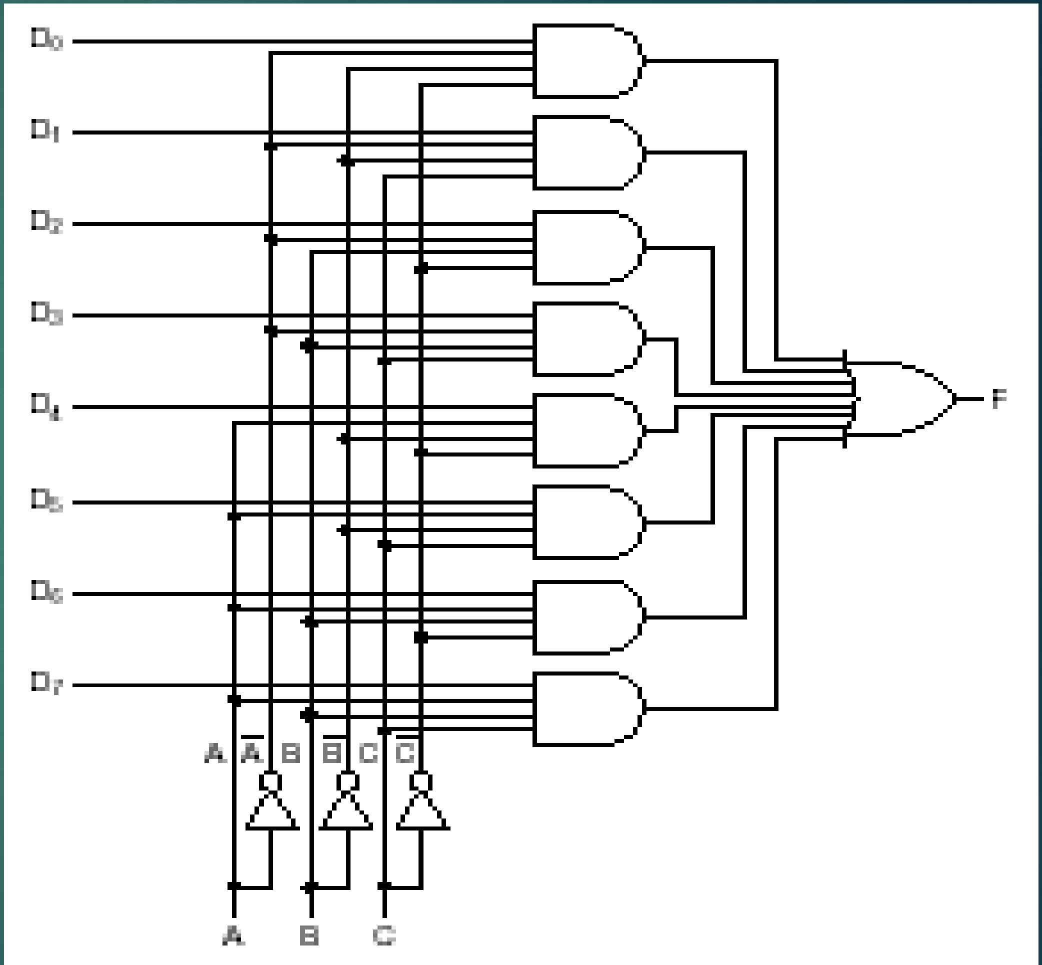
Puertas lógicas en un chip

- Compuertas NAND en un circuito integrado (de baja escala de integración):



Ejemplo circuito combinatorio 1: Multiplexor de 8 entradas

A	B	C	F
0	0	0	D_0
0	0	1	D_1
0	1	0	D_2
0	1	1	D_3
1	0	0	D_4
1	0	1	D_5
1	1	0	D_6
1	1	1	D



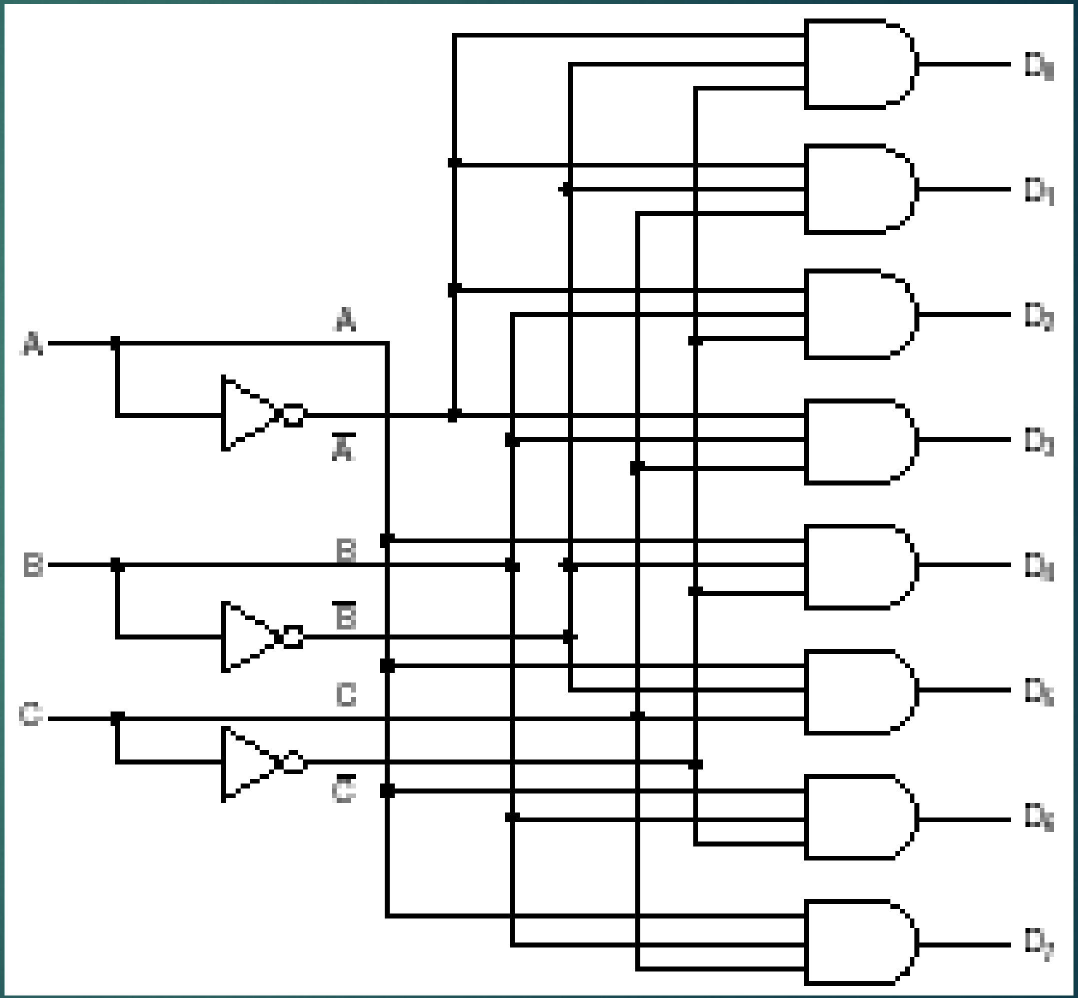
Según valor de entradas A, B y C

$F=D_x$

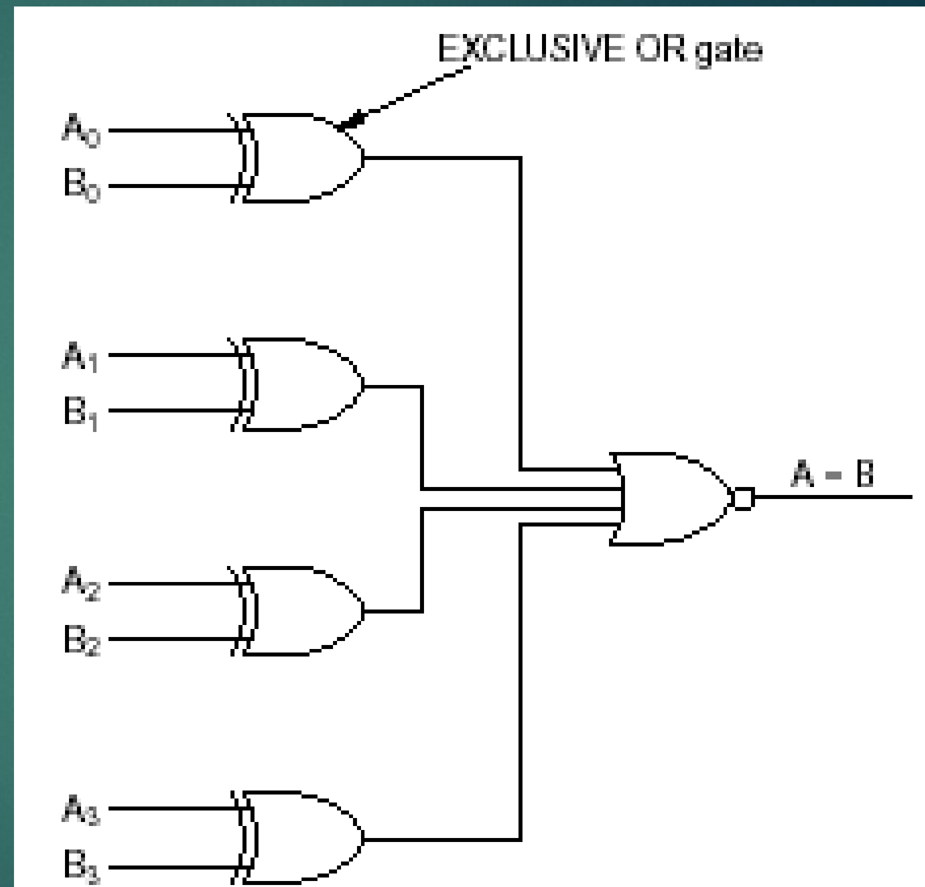
Ejemplo circuito combinatorio 2: Decodificador 3 a 8

Para cada combinación de las entradas A, B y C

sólo UNA de las salidas D_x vale ‘1’



Ejemplo circuito combinatorio 3: Comparador de 4 bits



La salida es 1 si:

$A_0 = B_0$ y

$A_1 = B_1$ y

$A_2 = B_2$ y

$A_3 = B_3$

Es decir : $A_i = B_i$

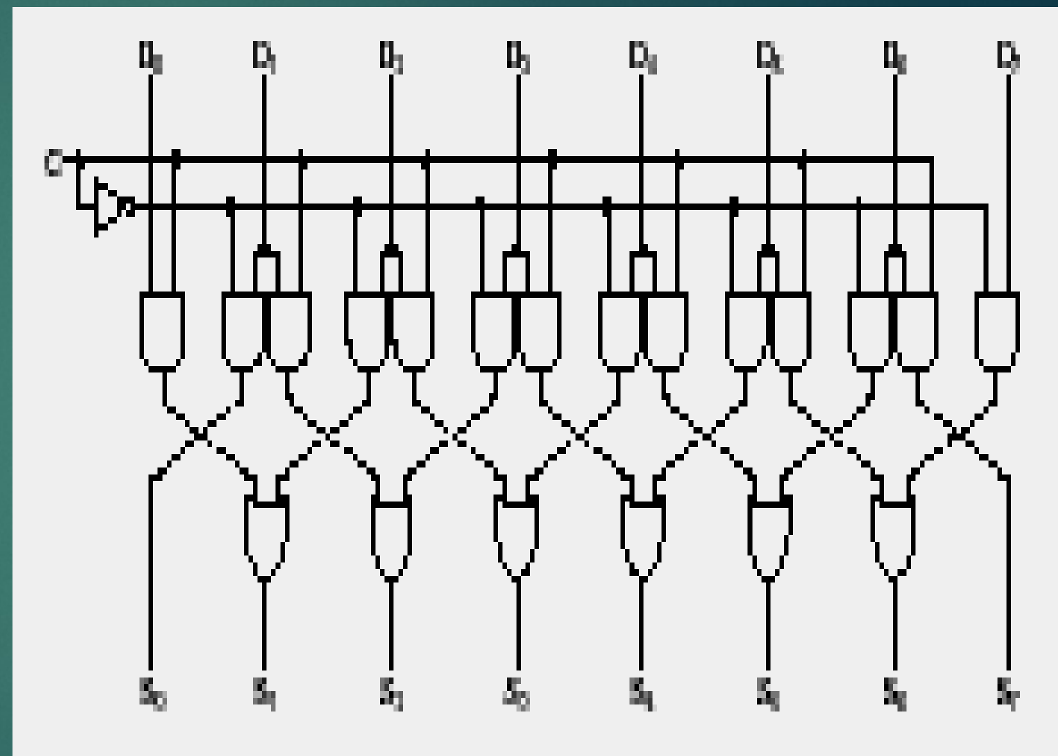
Ejemplo circuito combinatorio 4: Desplazador de 1 bit

Si:

- C = 0, entonces las entradas Di se desplazan un lugar a izquierda

y si:

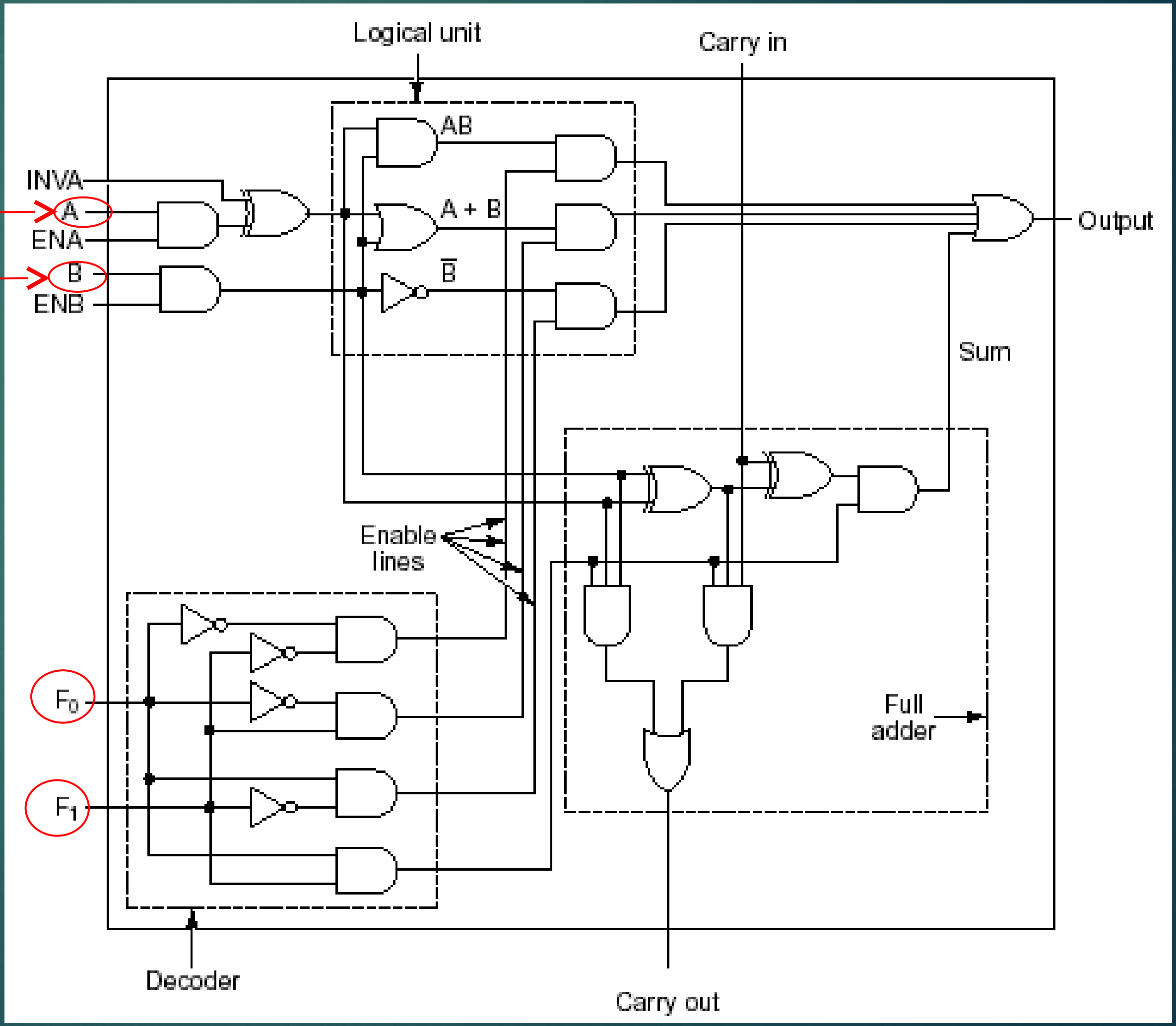
- C = 1, entonces las entradas Di se desplazan un lugar a derecha



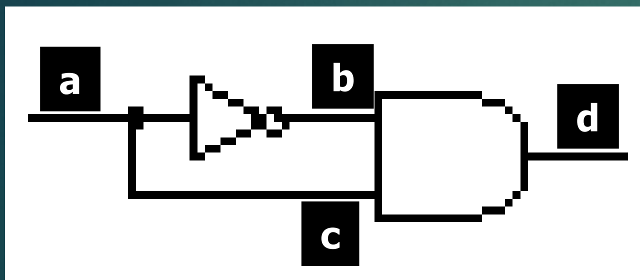
Ejemplo circuito combinatorio 5: ALU de 1 bit

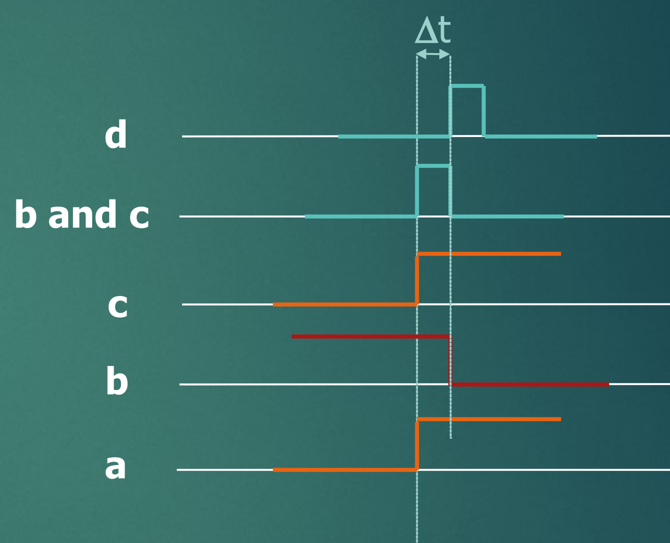
Según F_1F_0 será la función que se realizará sobre A y B.

F0	F1	Output
0	0	A.B
0	1	A+B
1	0	$\overline{B}$
1	1	suma con carry



Respuesta temporal de circuitos digitales





a	b	c	d
0	1	0	0
1	0	1	0

Δt : Retardo de compuerta

Suponemos que:

1)Los retardos en los cables son 0

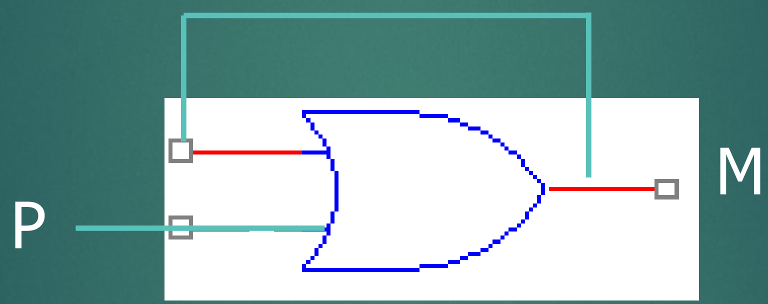
2)Los retardos en las compuerta son todos iguales a Δt

Circuitos Secuenciales

- Los circuitos secuenciales son circuitos donde las salidas dependen tanto de las entradas como del "estado interno" del circuito.
  - ¿Qué es el estado interno del circuito?
- Los circuitos secuenciales tienen la característica de **retener ("almacenar")** internamente valores.
- Estos valores internos pueden mantenerse aunque las entradas se hayan modificado: concepto de "almacenamiento".

Circuito secuencial elemental

- Para ver como se comporta un circuito secuencial, consideremos un circuito combinatorio con la salida conectada a una de las entradas.



- La ecuación lógica de este circuito es:

$$M = M + P$$

- Notar que en ningún circuito combinatorio una salida transportaba información hacia la entrada.

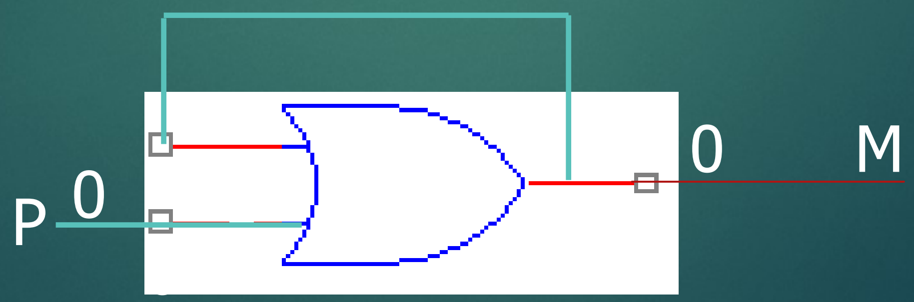
Respuesta del circuito secuencial elemental

Para analizar la respuesta de un circuito secuencial se debe hacer lo siguiente:

1er PASO: Partir de un estado cualquiera, usualmente uno fácil. Supongamos, por ejemplo: P=0 y M=0

entonces: M = M + P = 0 + 0 = 0

El resultado es consistente con la suposición inicial.



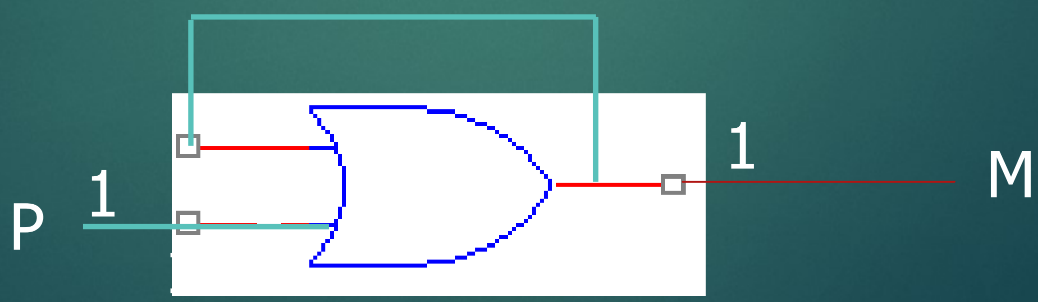
Cumple con las condiciones de partida.

Respuesta del circuito secuencial elemental

2do PASO: cambiar alguna variable de entrada, y determinar los demás estados.

Si P=1 $\qquad$ M=M+P=1+1=1

El resultado es consistente con la suposición inicial. Cumple con las condiciones de partida.

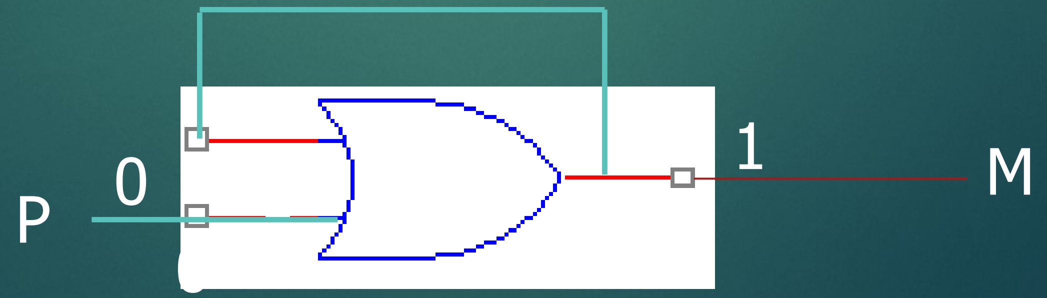


Respuesta del circuito secuencial elemental

3er PASO: volver a cambiar alguna variable de entrada, y determinar los demás estados.

Si P=0 M=M+P=1+0=1

El resultado es consistente con la suposición inicial. Cumple con las condiciones de partida.



Respuesta del circuito secuencial elemental

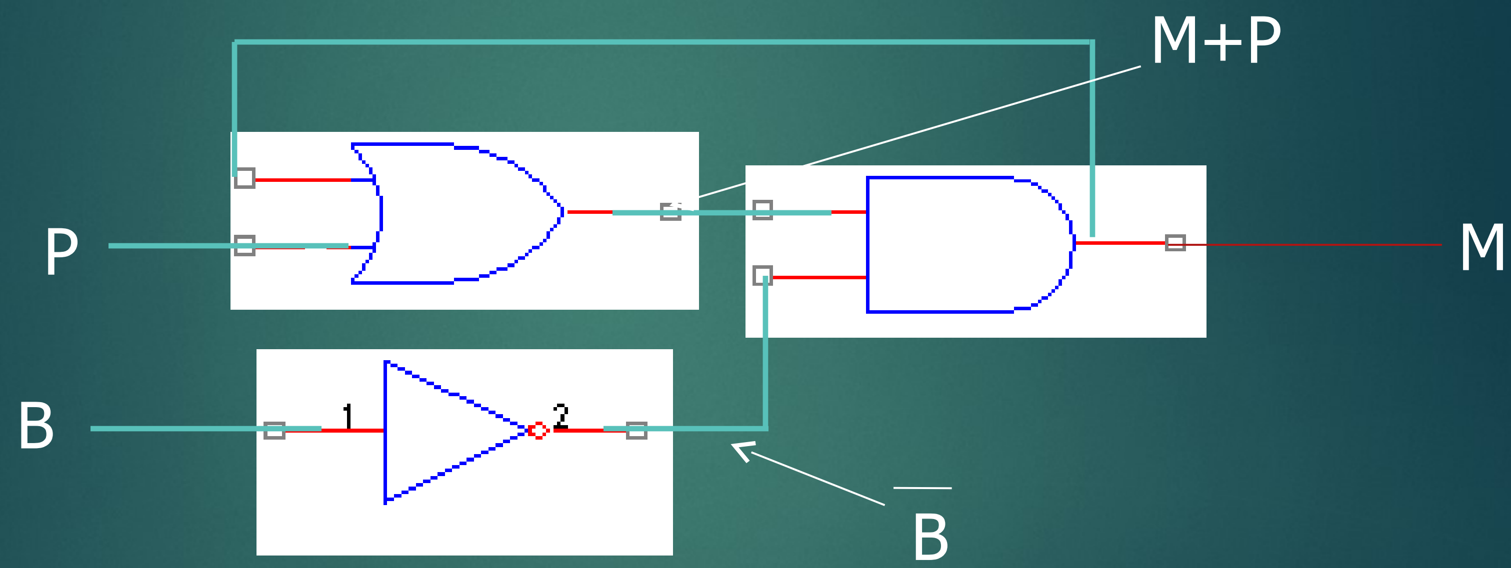
- Podríamos volver a cambiar alguna variable de entrada, y determinar los demás estados, pero el resultado siempre va a ser el mismo: M = 1

  independientemente del valor de P.

- La salida queda en 1, porque no depende únicamente de P sino que también del valor que tenía M anteriormente.
- Conclusión: quedó “almacenado” el estado anterior (P=1).
- Inconveniente: una vez que la salida M toma el valor 1 no hay forma de volver a 0.

Circuito secuencial básico

- Veamos el circuito secuencial básico siguiente:



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

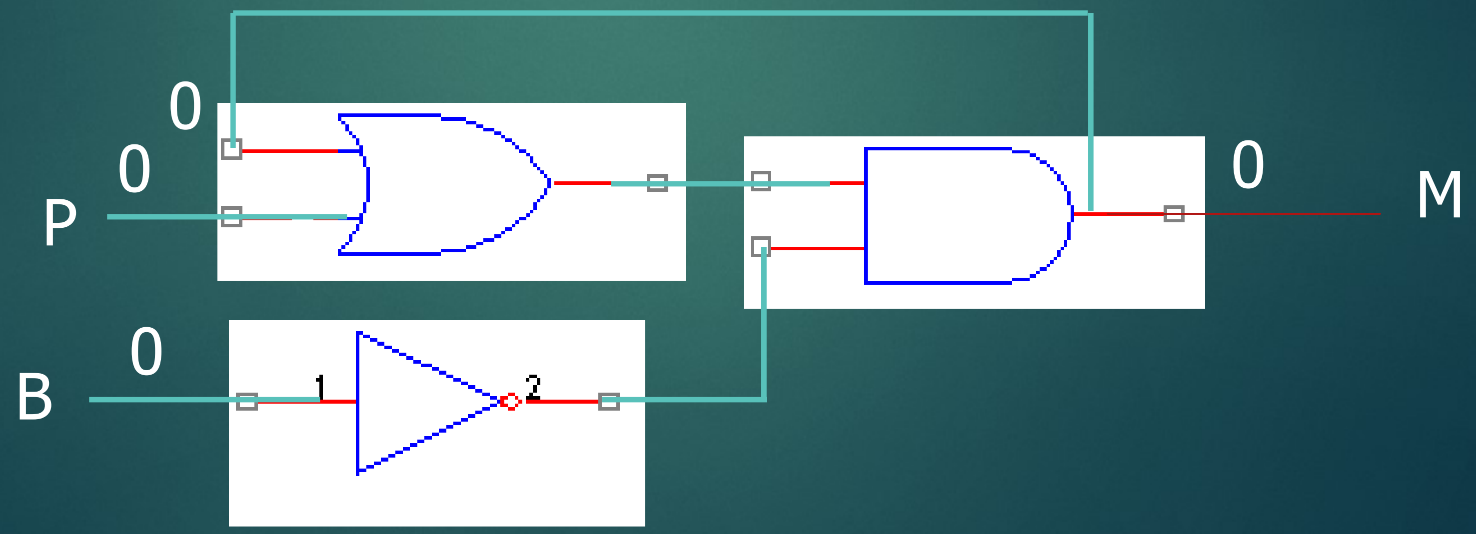
Respuesta del circuito secuencial básico

Para el análisis del comportamiento del circuito secuencial básico aplicamos la misma metodología de antes.

1er paso: supongamos que P=0 , B=0, y M=0

$$M=(M+P).\overline{B}=(0+0).1=0$$

Igual que en el circuito elemental, al principio si P = 0, M = 0.



Respuesta del circuito secuencial básico

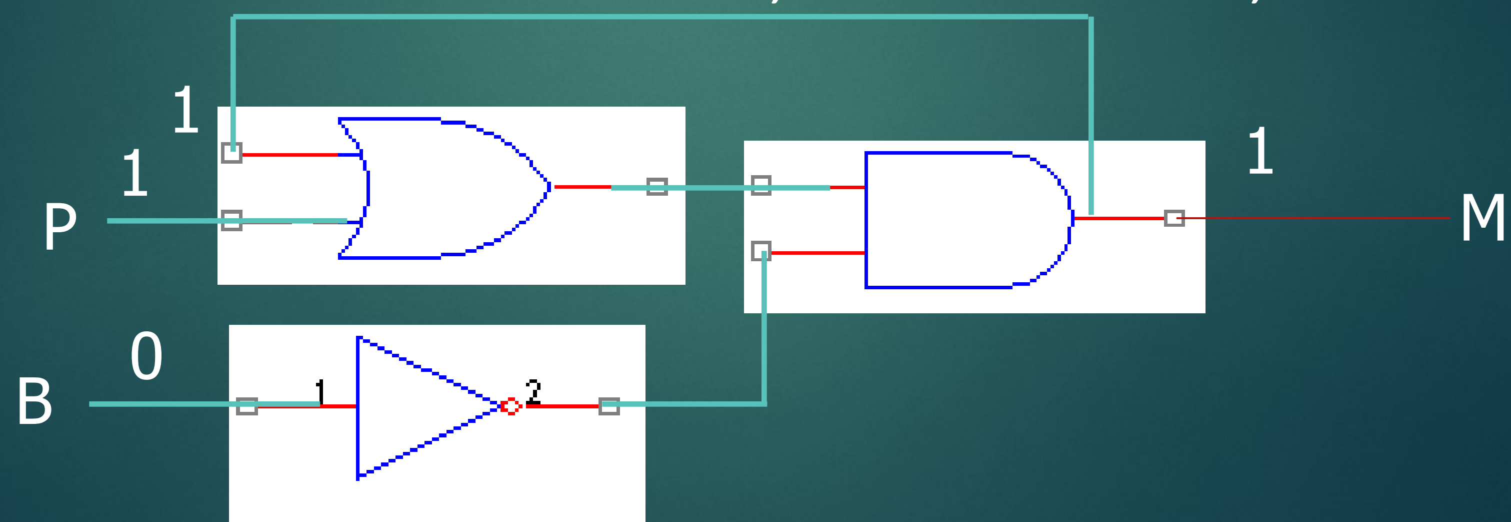
2do paso: cambiamos una variable de entrada, por ejemplo P.

Si P=1 y B=0

(es decir cambiamos P, pero no cambiamos B)

$$M=(M+P).\overline{B} =(1+1).1=1$$

igual que en el circuito elemental, si P cambia a 1, M cambia a 1.



Respuesta del circuito secuencial básico

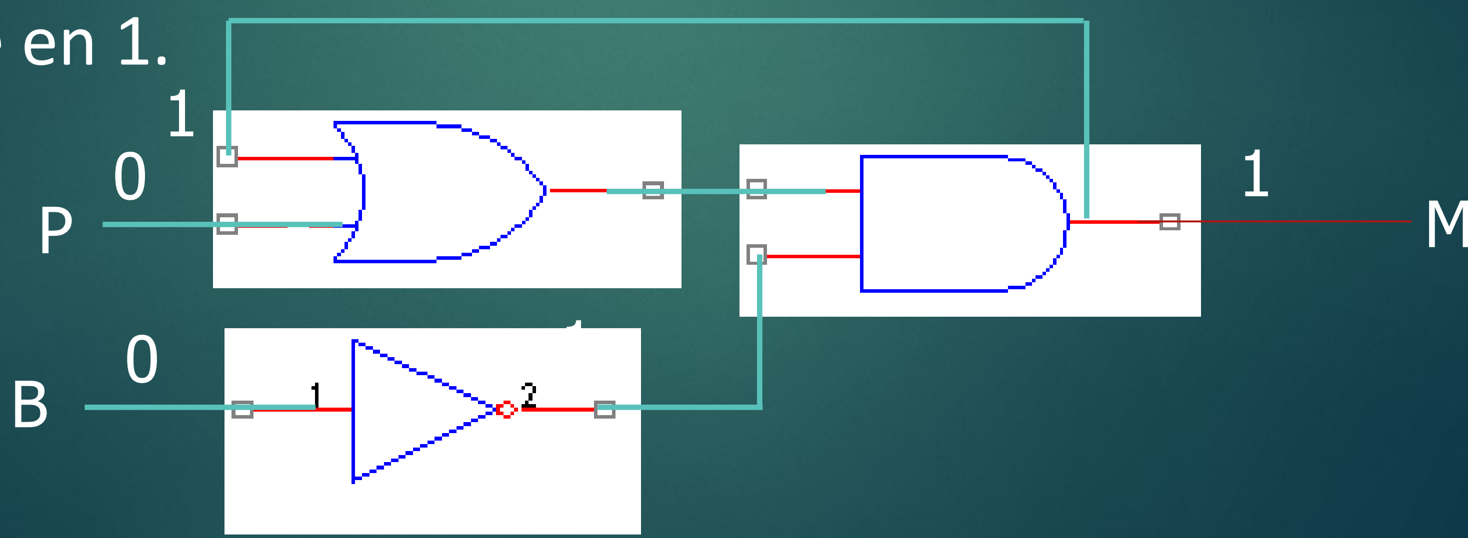
3er paso: cambiamos nuevamente la variable de entrada P.

Si P=0 y B=0

(es decir cambiamos nuevamente P, pero no cambiamos B)

$M=(M+P).\overline{B} =(1+0).1=1$

igual que en el circuito elemental, si P vuelve a 0, M se mantiene en 1.



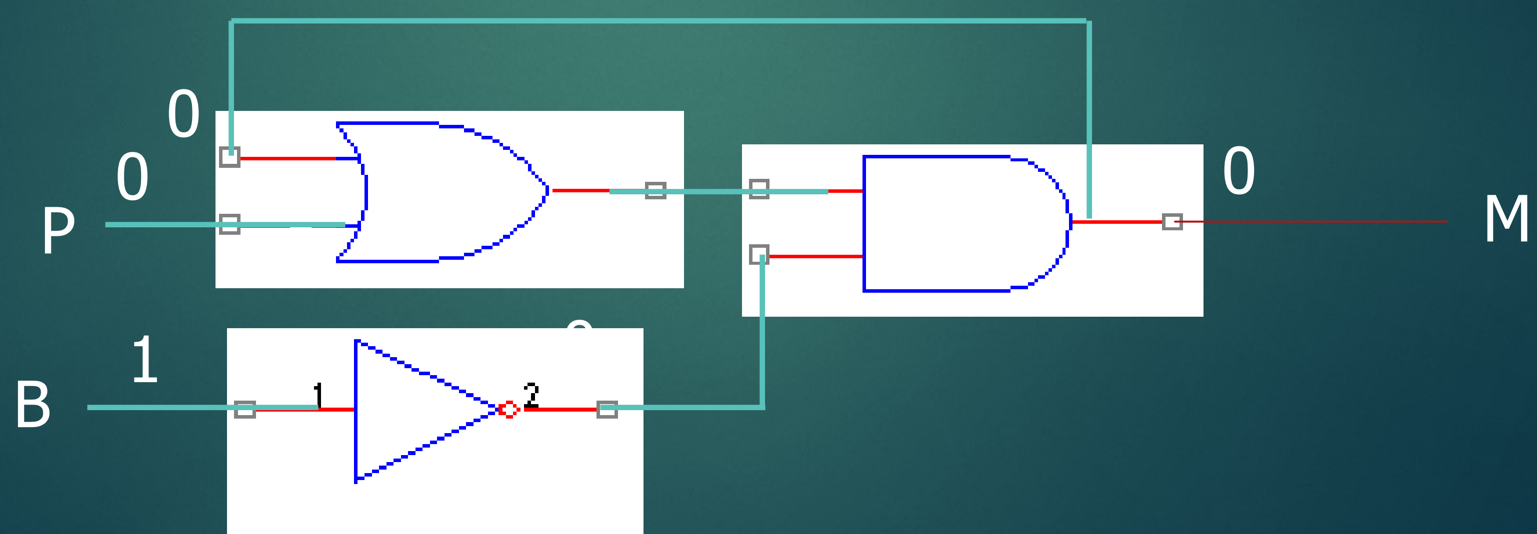
Respuesta del circuito secuencial básico

4to paso: ahora cambiamos la otra variable de entrada B.

Si P=0 y B=1

$$M=(M+P).\overline{B}=(1+0).0=0$$

al cambiar B = 1, M vuelve a 0.



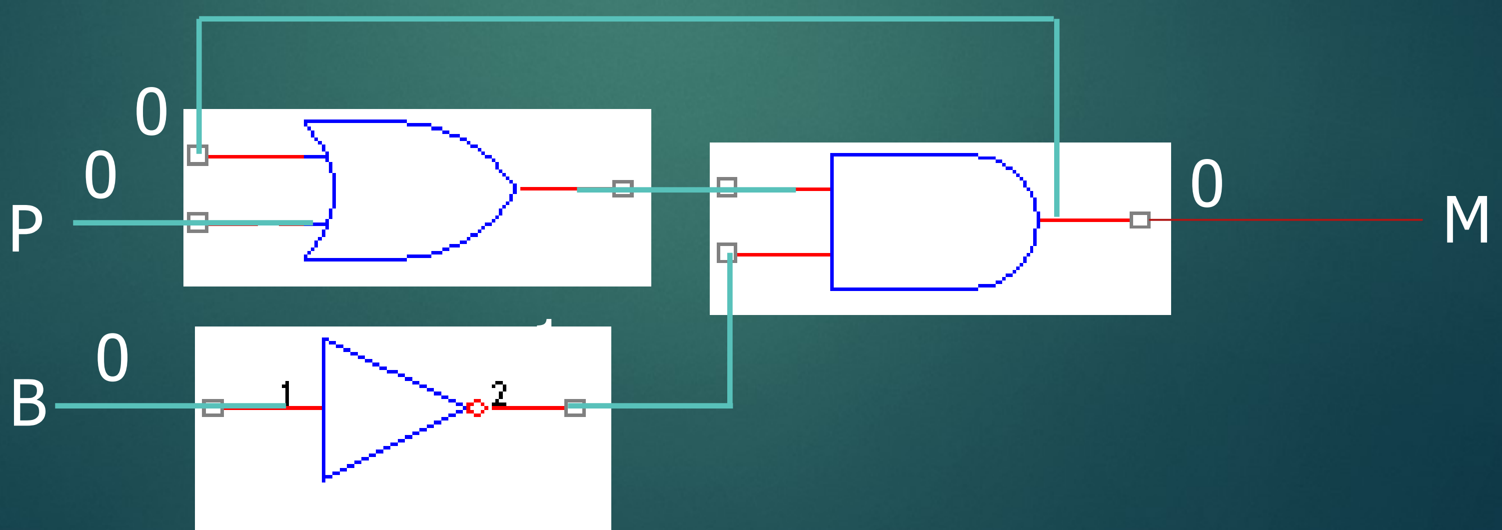
Respuesta del circuito secuencial básico

5to paso: cambiamos nuevamente la variable de entrada B.

Si P=0 y B=0

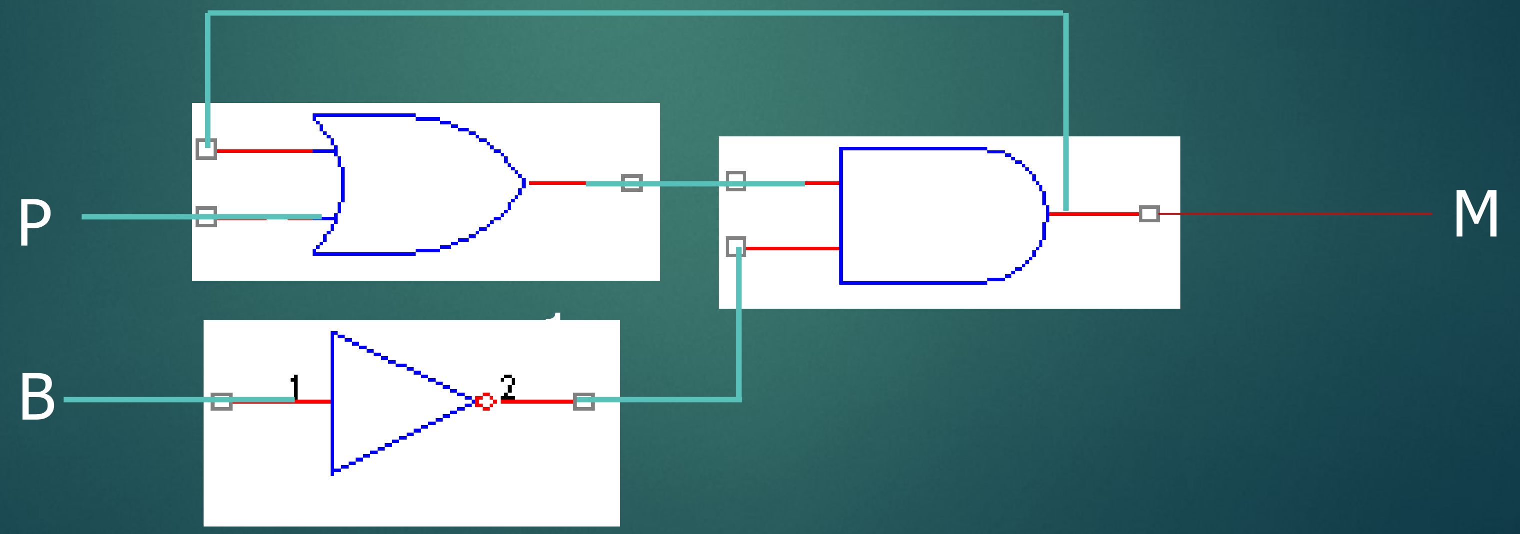
$$M=(M+P).\overline{B}=(0+0).1=0$$

- Conclusión: el circuito quedó con P=0, B=0, M=0, que es el estado inicial.



Circuito secuencial básico con un solo tipo de compuerta

- El circuito secuencial básico que usamos tiene 3 tipos de compuertas: NOT, OR y AND.
- Es mejor si el circuito básico usara un solo tipo de compuerta.

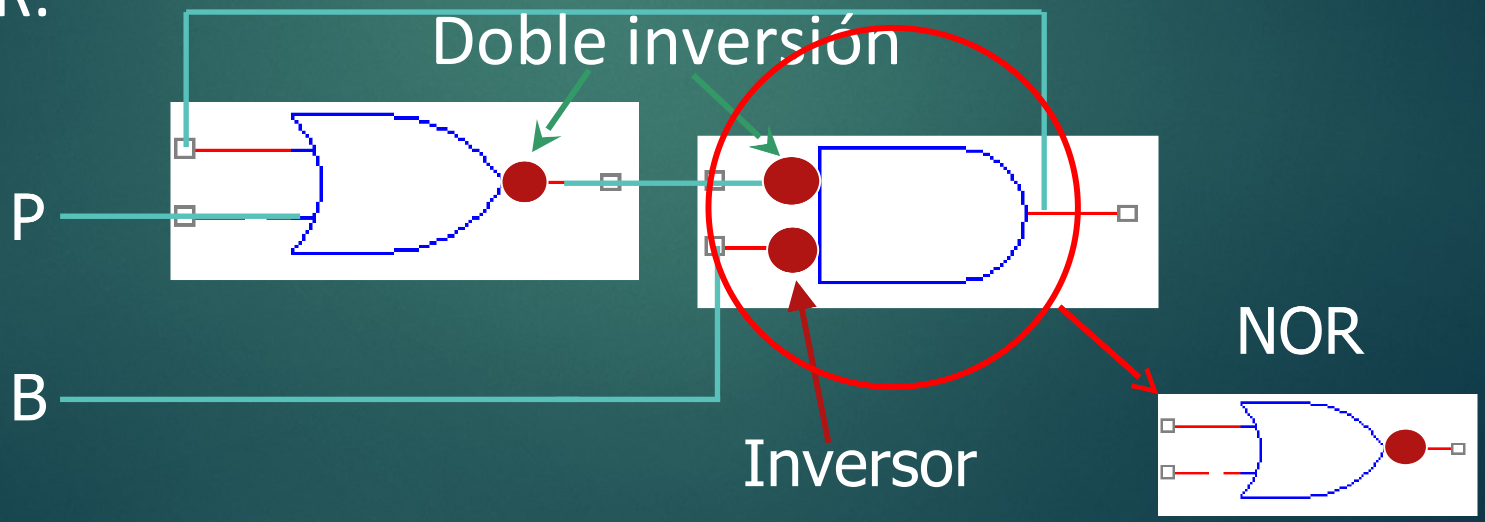


Circuito secuencial básico con un solo tipo de compuerta

Podemos modificar el circuito lógico anterior, sin alterar su funcionamiento, de la siguiente manera:

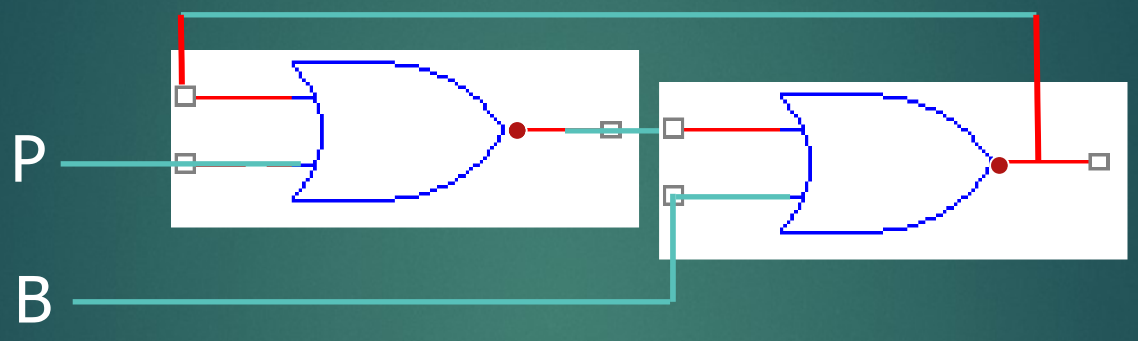
1) Invertimos 2 veces, una a la salida de la OR (queda como NOR) y otra a la entrada de la AND.

2) El inversor lo pasamos a la entrada de la AND, quedando como NOR.

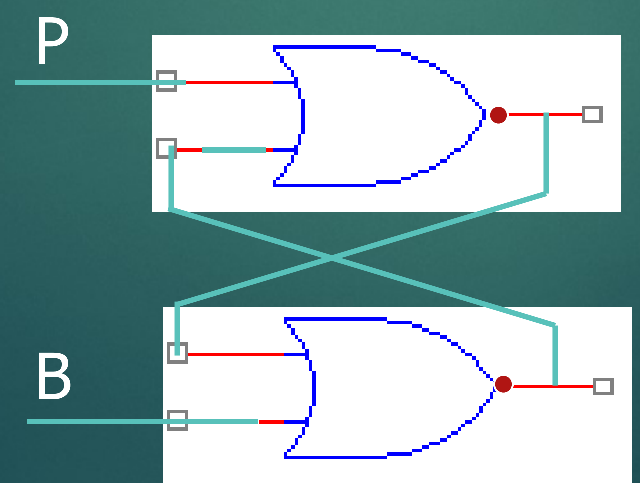


Circuito secuencial básico con un solo tipo de compuerta

El circuito anterior entonces queda:

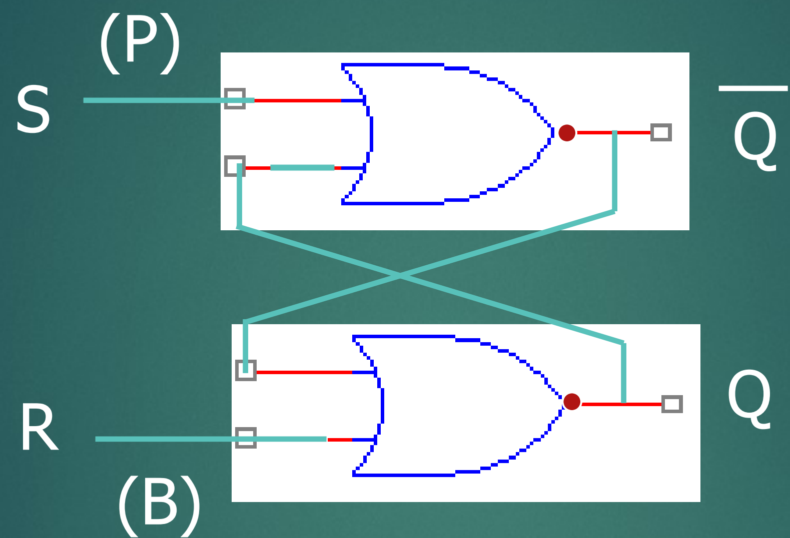


O bien, dibujado de una forma más estándar:



Flip Flop SR (FF SR)

El circuito anterior se conoce como FF tipo SR:

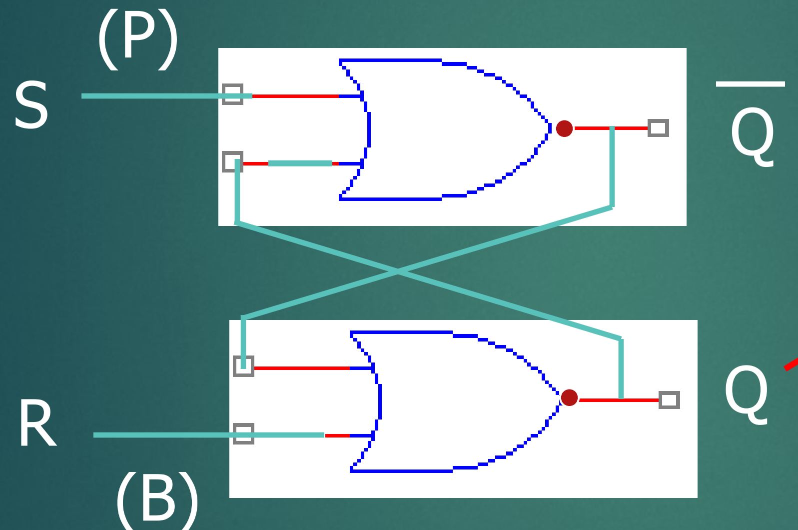


Donde:

- Entrada P = entrada S (Set)
- Entrada B = entrada R (Reset)
- Salida Q = salida normal
- Salida $\overline{Q}$ = salida invertida

Flip Flop SR (FF SR)

- El FF SR tiene una tabla de verdad que describe su comportamiento:



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

$Q = Q_{n+1}$: Valor actual

$Q = Q_n$: Valor anterior

Flip Flop SR (FF SR)

Comentarios sobre el FF SR

- El FF SR tiene 2 salidas Q y $\overline{Q}$, que son complementarias, eso significa que : si Q=0 entonces Q=1, y visceversa.
- La entrada S (Set) pone la salida Q en 1
- La entrada R (Reset) pone la salida Q en 0
- La combinación S=1 y R=1 está prohibida, es decir no se puede hacer (en realidad significa que el FF SR queda en un estado impredecible).
- Salida Q = Q_{n+1} significa próximo estado
- Salida Q = Q_n significa estado anterior, es decir la salida retiene el estado previo.

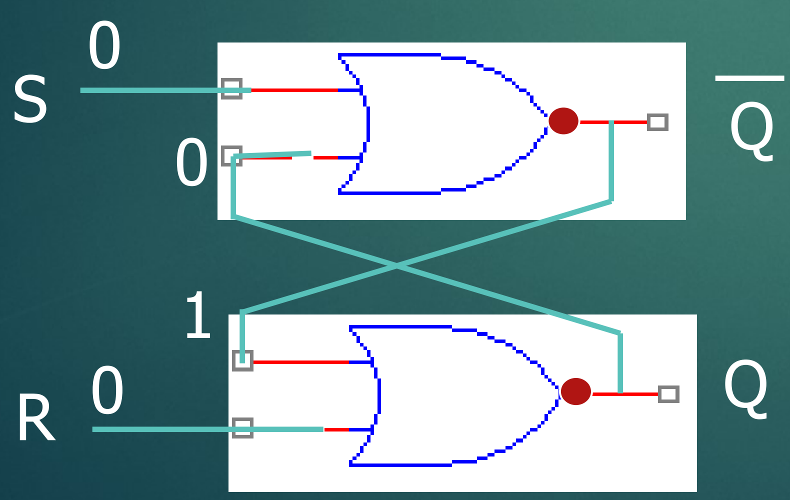
Análisis del Flip Flop SR (FF SR)

Para el análisis del comportamiento del circuito FF SR aplicamos la misma metodología de antes.

<u>1er paso</u>: si suponemos S=0 , R=0

si $Q_n = 0$ (es decir si Q estaba en 0)

$Q_{n+1} = 0$ (es decir Q mantiene el 0)



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

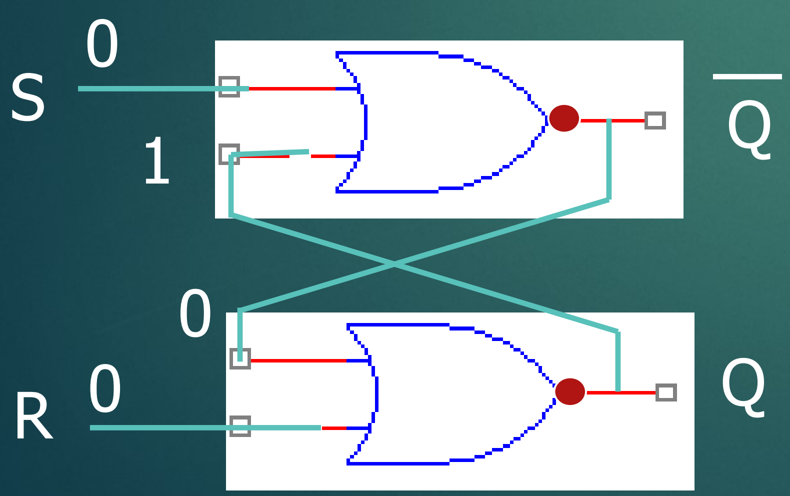
Análisis del Flip Flop SR (FF SR)

1er paso (continuación)

y si $Q_n = 1$ (es decir si Q estaba en 1)

$Q_{n+1} = 1$ (es decir Q mantiene el 1)

- **CONCLUSIÓN**: Si S y R están en 0, la salida Q mantiene ("recuerda") el estado que tenía antes.



S	R	Q_{n+1}
0	0	$\boxed{Q_n}$
0	1	0
1	0	1
1	1	Prohibido

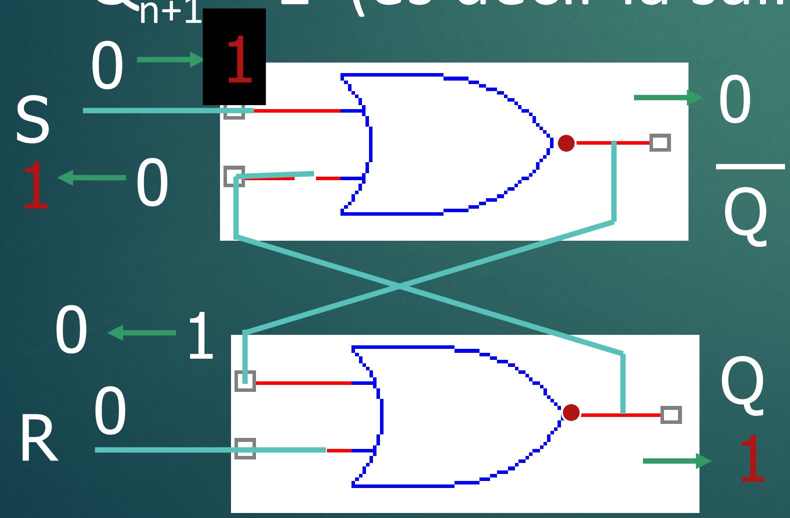
Análisis del Flip Flop SR (FF SR)

2do paso: si suponemos S=1 , R=0

Q_{n+1} = 1 (es decir la salida Q queda en 1, independientemente del estado que estaba antes)

3er paso: si volviéramos al estado inicial S=0 , R=0, Q_n = 1

Q_{n+1} = 1 (es decir la salida Q queda en 1)



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

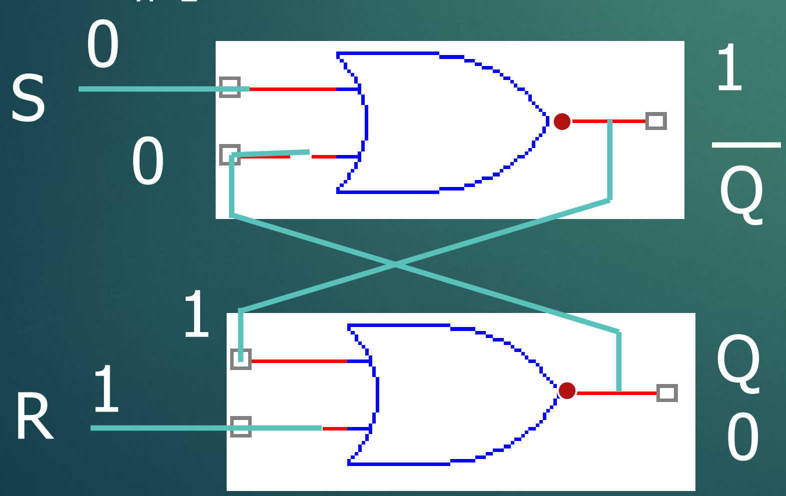
Análisis del Flip Flop SR (FF SR)

4to paso: si suponemos S=0 , R=1

$Q_{n+1} = 0$ (es decir la salida Q queda en 1, independientemente del estado que estaba antes)

5to paso: si volvemos a la condición inicial S=0, R=0, $Q_n = 0$

$Q_{n+1} = 0$ (es decir la salida Q queda en 0)



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

Flip Flop SR (FF SR) como memoria

- El circuito mantiene la salida Q en 1 o en 0 aún cuando las entradas vuelven a 0.
- La salida Q no solo depende de la entrada sino también de Q_n , es decir del estado en que se encontraba anteriormente.
- La combinación S=1 y R=1 está prohibida.

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

Flip Flop como memoria

- Es decir, el circuito “memoriza” el estado en que se encontraba antes (0 o 1).
- Se puede decir que un flip-flop es una “memoria de 1 bit” (hay otros circuitos capaces de memorizar 1 bit).
- Se llama bi-estable porque el circuito posee sólo 2 estados posibles de funcionamiento, se queda en cada uno de ellos, salvo que las entradas provoquen un cambio.

Circuitos secuenciales sincrónicos y asincrónicos

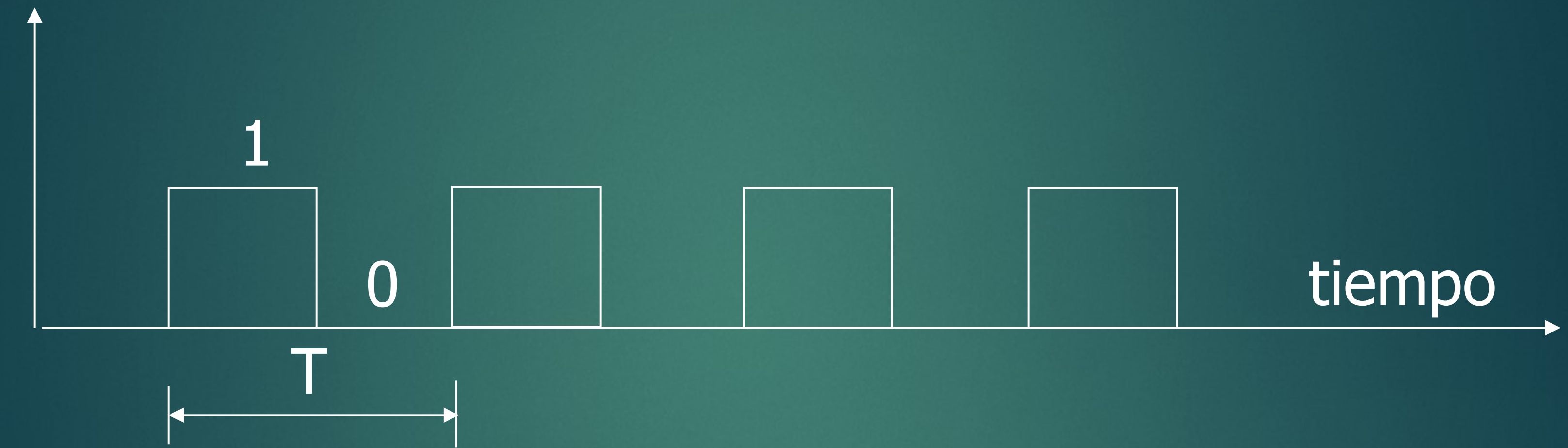
- Los circuitos secuenciales pueden ser:
  - Asincrónicos: las salidas cambian cuando cambian las entradas (si corresponde).
  - Sincrónicos: las salidas cambian cuando cambian las entradas **y** una señal (de sincronización) lo habilita.

Sincronización

- La señal de sincronización es una referencia temporal llamada “reloj”.
- Permite definir el instante en el que ocurren los sucesos.
- En especial si los sucesos deben ocurrir simultáneamente.

Reloj (Clock o CK)

- El reloj es una señal que cambia periódicamente entre 1 y 0 a intervalos regulares de tiempo.



- La señal de reloj se repite cada intervalo de tiempo T.
- T es el período del reloj

  siendo $T=1/f$ y f, frecuencia del reloj.

- Si f= 2GHz => $T = 1/2x10^{9}=0.5x10^{-9}$ = 0.5nseg

Flip-Flop SR sincrónico

- S y R son las mismas entradas de antes.
- Se agrega la entrada de reloj (CK).
- Solo pueden haber cambios si CK =1.
- Si CK=0, entonces las salidas no cambian.

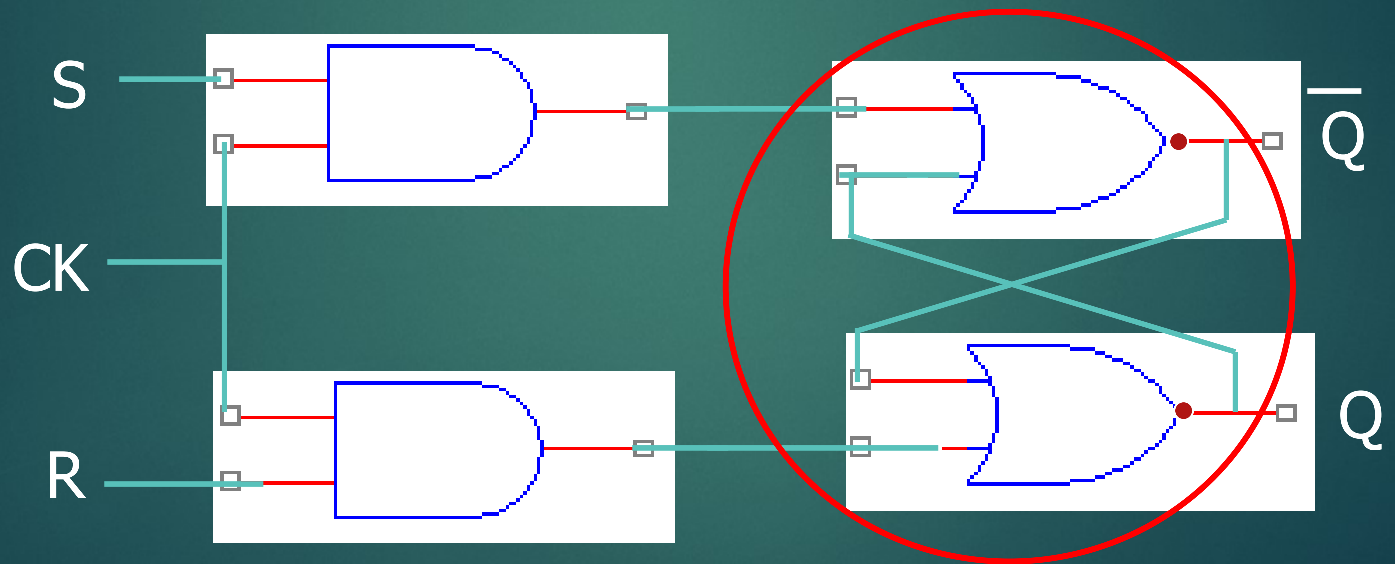


Tabla de la verdad del FF SR sincrónico

- La tabla del FF SR sincrónico es similar a las anteriores con el agregado de la señal CK.
- Si CK=0, no importan los valores de S y R (indicado con x en la tabla) la salida retiene el valor anterior. Cuando CK=1 el FF opera como se vio anteriormente.

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

Tipos de FF

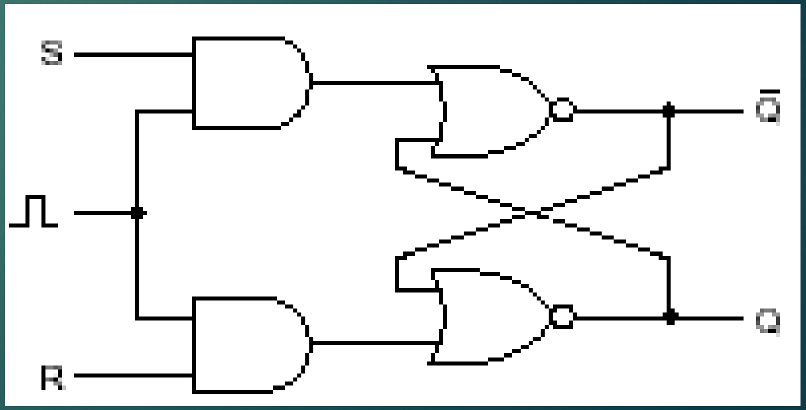
- Existen diferentes tipos de FF, de acuerdo a como están conectadas sus entradas, y como actúan sus salidas.
- Los FF más comunes son:
  - Tipo S-R
  - Tipo D
  - Tipo J-K
  - Tipo T

Flip-Flop S-R

- Es el FF que hemos usado hasta ahora.
- En el FF S-R hay que actuar sobre 2 entradas diferentes (S y R) para cambiar de estado.
- Hay un estado "prohibido" que dificulta su uso.

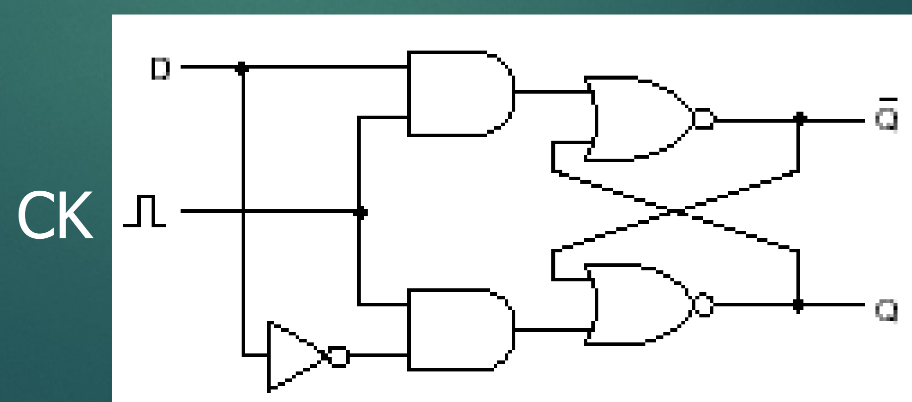
CK	S	R	Q_{n+1}
1	0	0	Q_n
1	0	1	0
1	1	0	1
1	1	1	Prohibido
0	x	x	Q_n

CK



Flip-Flop D

- Se puede modificar el FF SR para simplificar su uso.
- Por ejemplo, el FF tipo D es un FF con una sola entrada de datos.
- Se puede pensar como un FF tipo S-R en el que la segunda entrada es la invertida de la primera.
- La ventaja que tiene es que se requiere una sola señal para cambiar la salida (y por lo tanto para almacenar 1 bit).
- El circuito es:



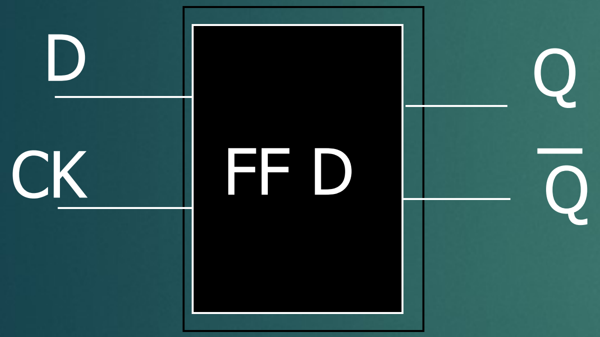
Flip-Flop D

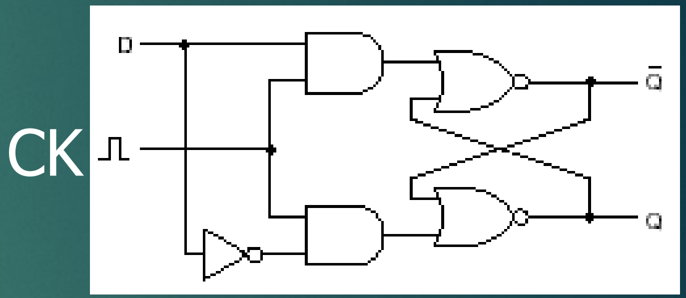
- Como se puede apreciar de la tabla de la verdad del FF D, dado que tiene una sola entrada, no es posible que ocurra un situación de estado prohibido.

D	Qn+1
0	0
1	1

SIMBOLO Tabla de la verdad

- Comparar con la tabla de la verdad del FF S-R.
- El FF tipo D solo usa los S=0 R=1, y S=1 R=0.

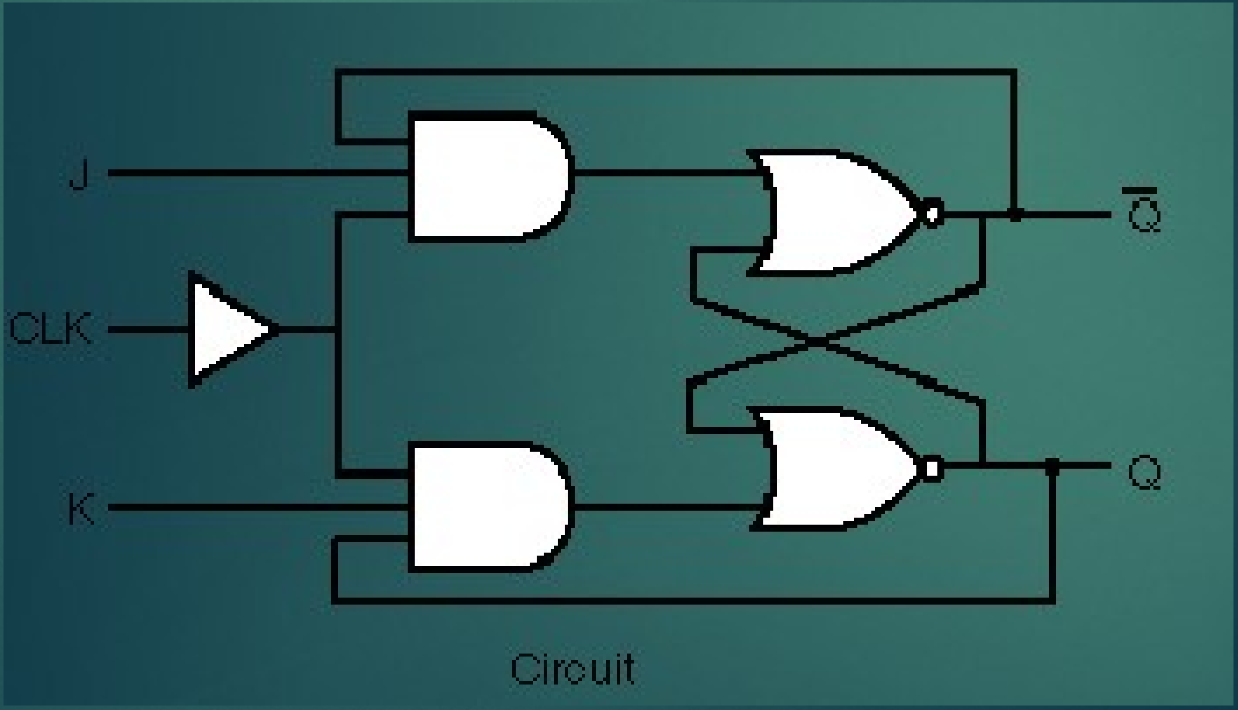




S	R	Q_{n+1}
0	0	Q_n
0	1	0
1	0	1
1	1	Prohibido

Flip Flop J-K

- El FF tipo J-K es un FF con 2 entradas de datos.
- Se puede pensar como un FF tipo S-R en el que se ha eliminado el estado prohibido de las entradas.
- La ventaja que tiene es que no tiene estados prohibidos.
- El circuito es:

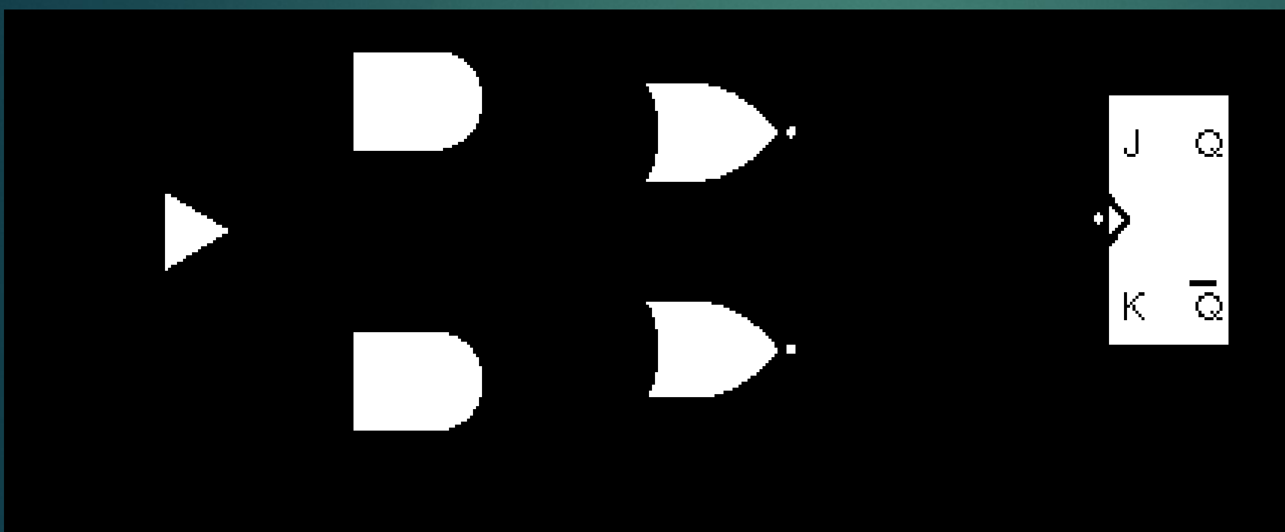


Flip Flop J-K

- Como se puede apreciar de la tabla de la verdad, el estado J=K=1 invierte el estado previo de la salida.

CIRCUITO SIMBOLO TABLA DE LA VERDAD

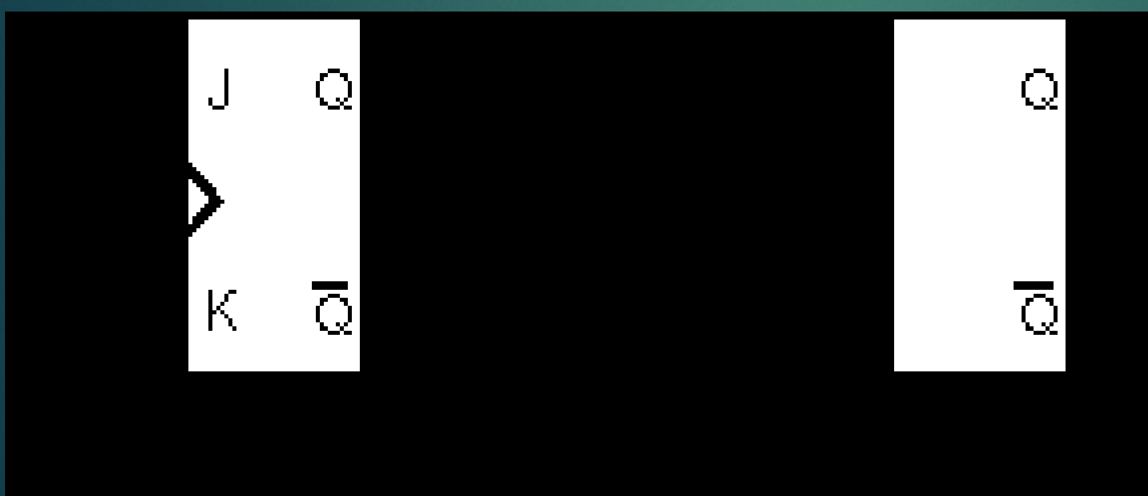
- Comparar circuito y tabla con el FF S-R.



J	K	Q_{n+1}
0	0	Q_n
0	1	0
1	0	1
1	1	Q_n

Flip Flop T

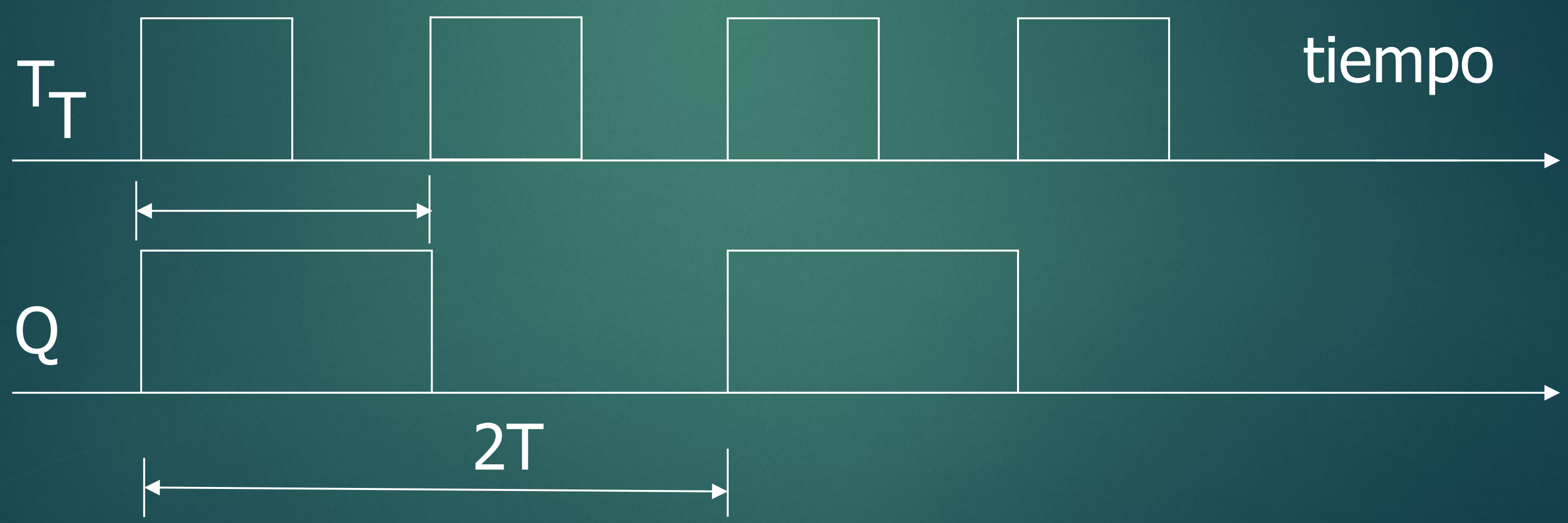
- Es un FF sin entradas de datos.
- Se puede pensar como un FF J-K con las 2 entradas conectadas a 1.
- La salida Q cambiará de 0 a 1, y de 1 a 0, en cada cambio de 0 a 1 de la entrada T (entrada de reloj).



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

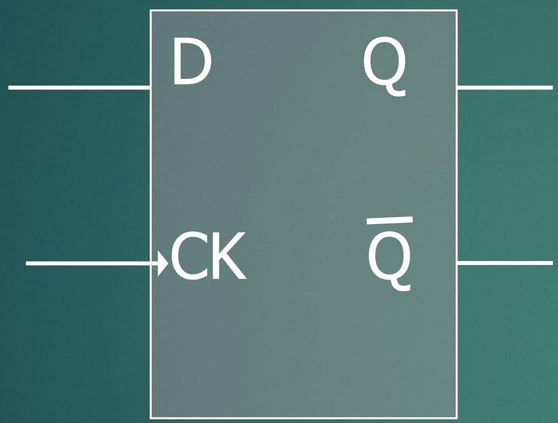
Flip Flop T

- La salida Q cambiará de 0 a 1 y de 1 a 0 en cada pulso de la entrada T.
- El nombre de FF tipo T proviene de que la salida cambia (“Toggle”) alternativamente entre 1 y 0.



Registro de 1 bit básico (“Memoria” de 1 bit)

- Consideremos un FF D:

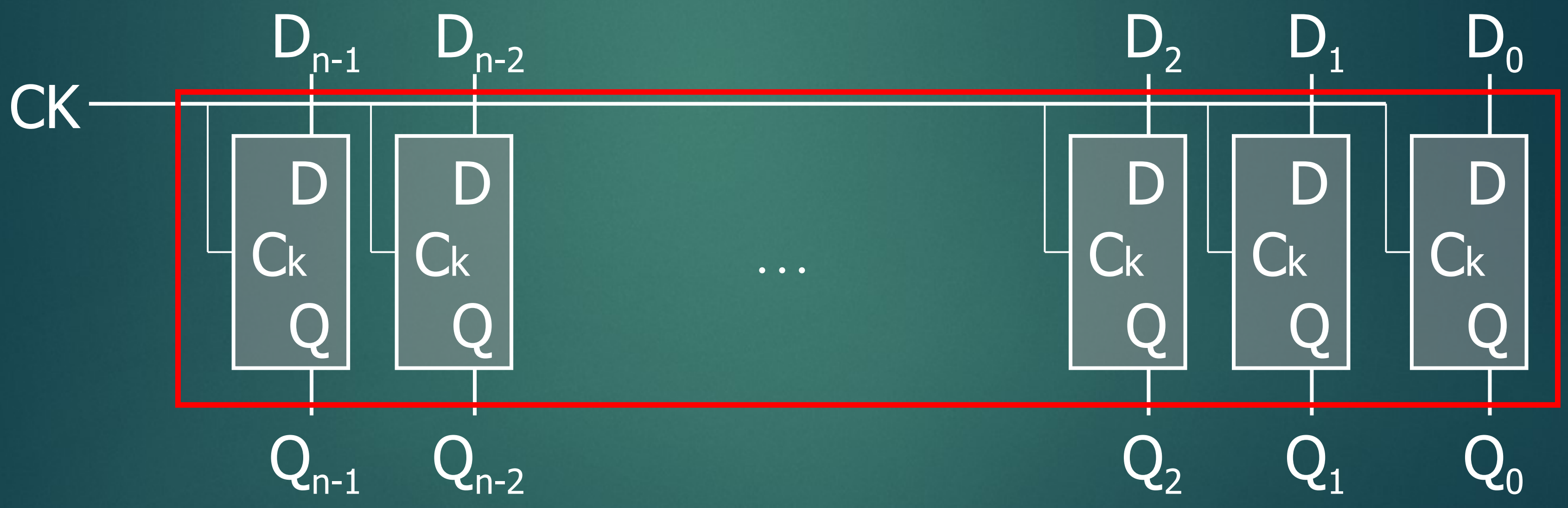


CK	D	Q
0	0	q
0	1	q
1	0	0
1	1	1

- Con la señal CK=1, la salida Q “copia” el valor D de la entrada. Con la señal CK=0, la salida “retiene” el valor que tenía previamente. Es decir, “memoriza” el valor de la entrada.
- Se puede decir que un FF D es un <u>registro (o memoria) de 1 bit</u>, porque puede almacenar solo 1 bit.

Registro de n bits básico

- Consideremos un arreglo de n FF tipo D como los del ejemplo anterior, cada uno con su entrada D_i y su salida Q_i, con una sola entrada de reloj CK actuando sobre los n FF tipo D simultáneamente.
- Este arreglo es un registro de n bits

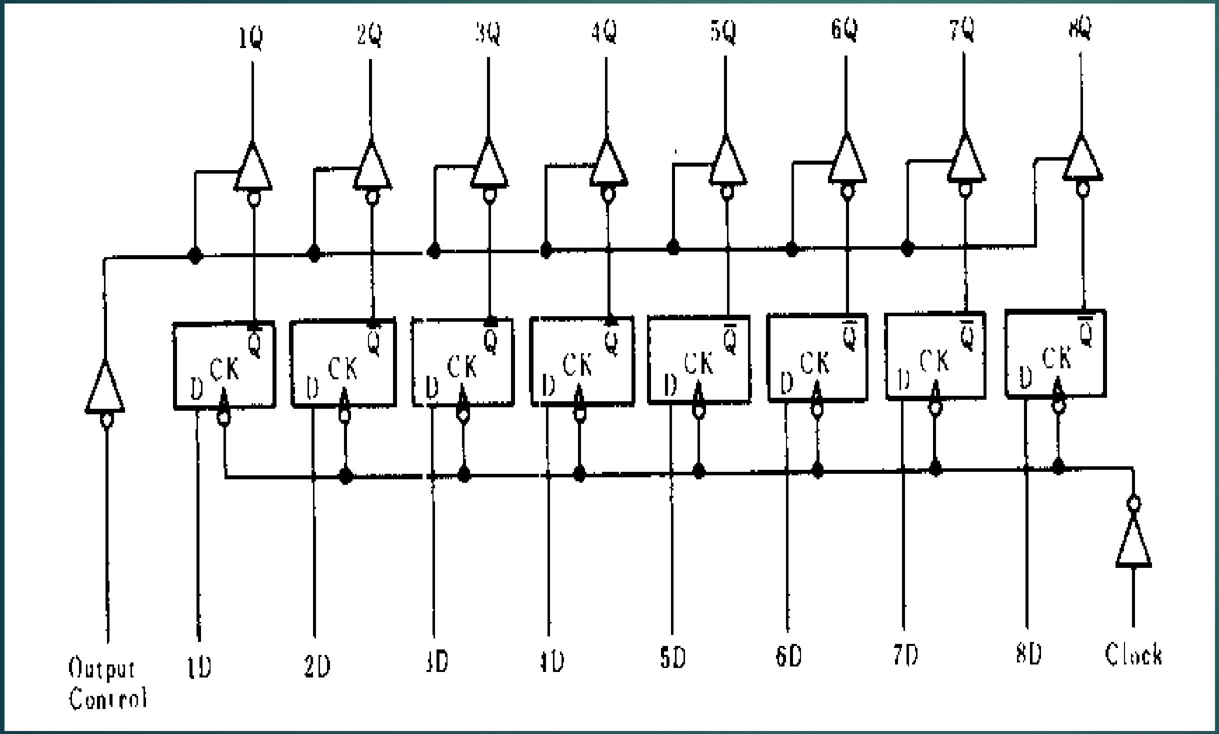


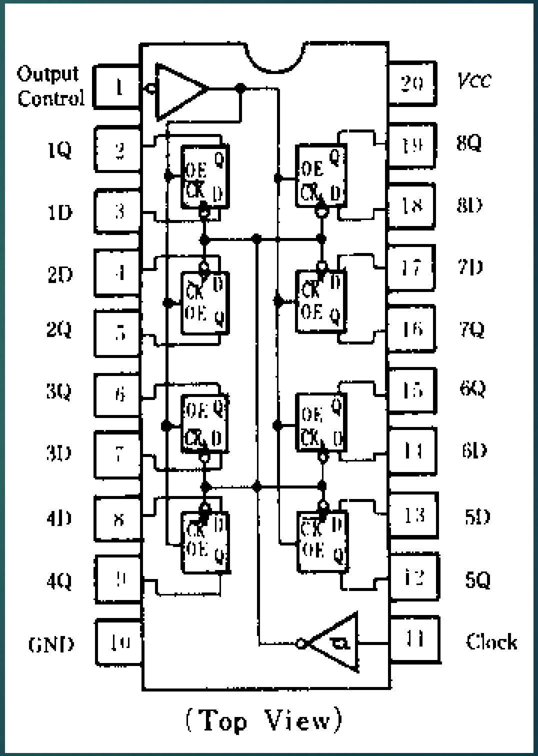
Ejemplo de “registro de 8 bits” - Chip con 8 FF-D (74374)

- En las imágenes se muestra un circuito integrado (de uso comercial identificado con la numeración 74374), de un registro de 8 bits, compuesto de 8 FF tipo D.

Diagrama funcional

Esquema eléctrico





Registro con entrada de selección

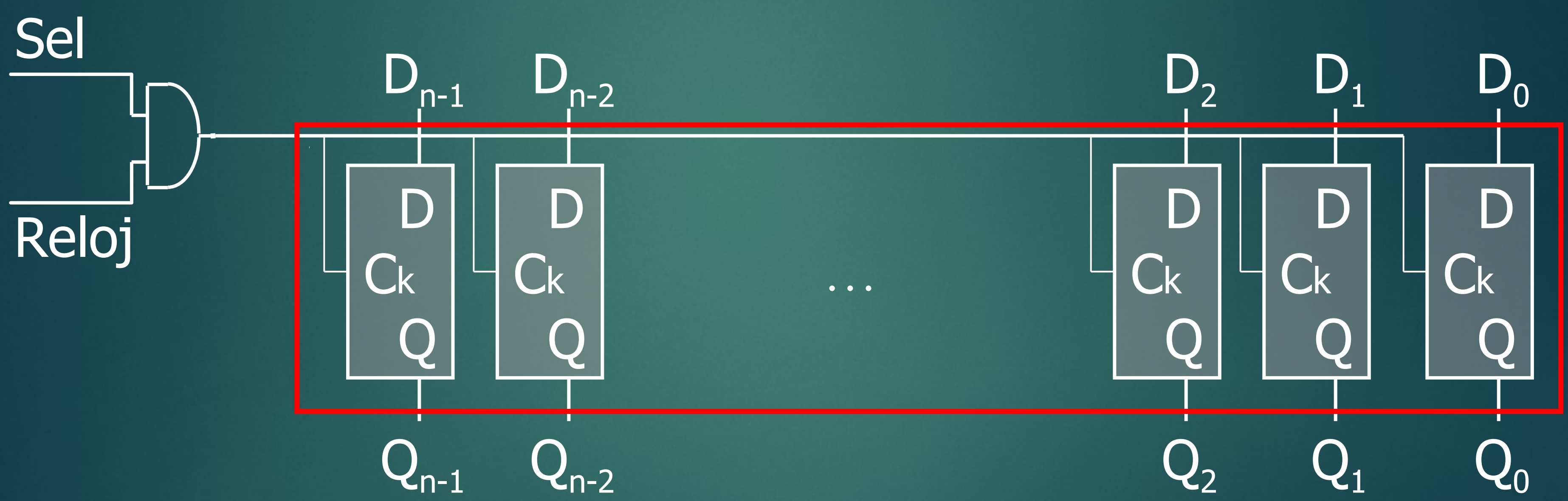
Problema del registro básico

- En los registros básicos analizados, la señal CK está conectada a un reloj que siempre cambia de 0 a 1 y de 1 a 0 (con un período T, como se vió).
- Por la tabla de la verdad del FF D, cada vez que la señal CK cambia, la salida copia la entrada, “refrescando” el valor almacenado en cada FF D.
- Eso significa que si cambia alguna entrada, cambia la salida.
- Sería más deseable que cambie sólo cuando se quiere modificar su contenido, es decir cuando se lo quiere “escribir”.
- Para resolver esto se requieren 1 señal adicional, que permita controlar si se modifica el registro o no.

Registro de n bits con entrada de selección

- Se puede modificar el circuito del registro de n bits, agregando a la entrada de Reloj una compuerta AND y una señal extra llamada Sel ("selección").

- La señal de reloj se propaga a las entradas Ck de los FF D si la entrada Sel=1, si Sel=0 los FF D no reciben los cambios de la señal Reloj.



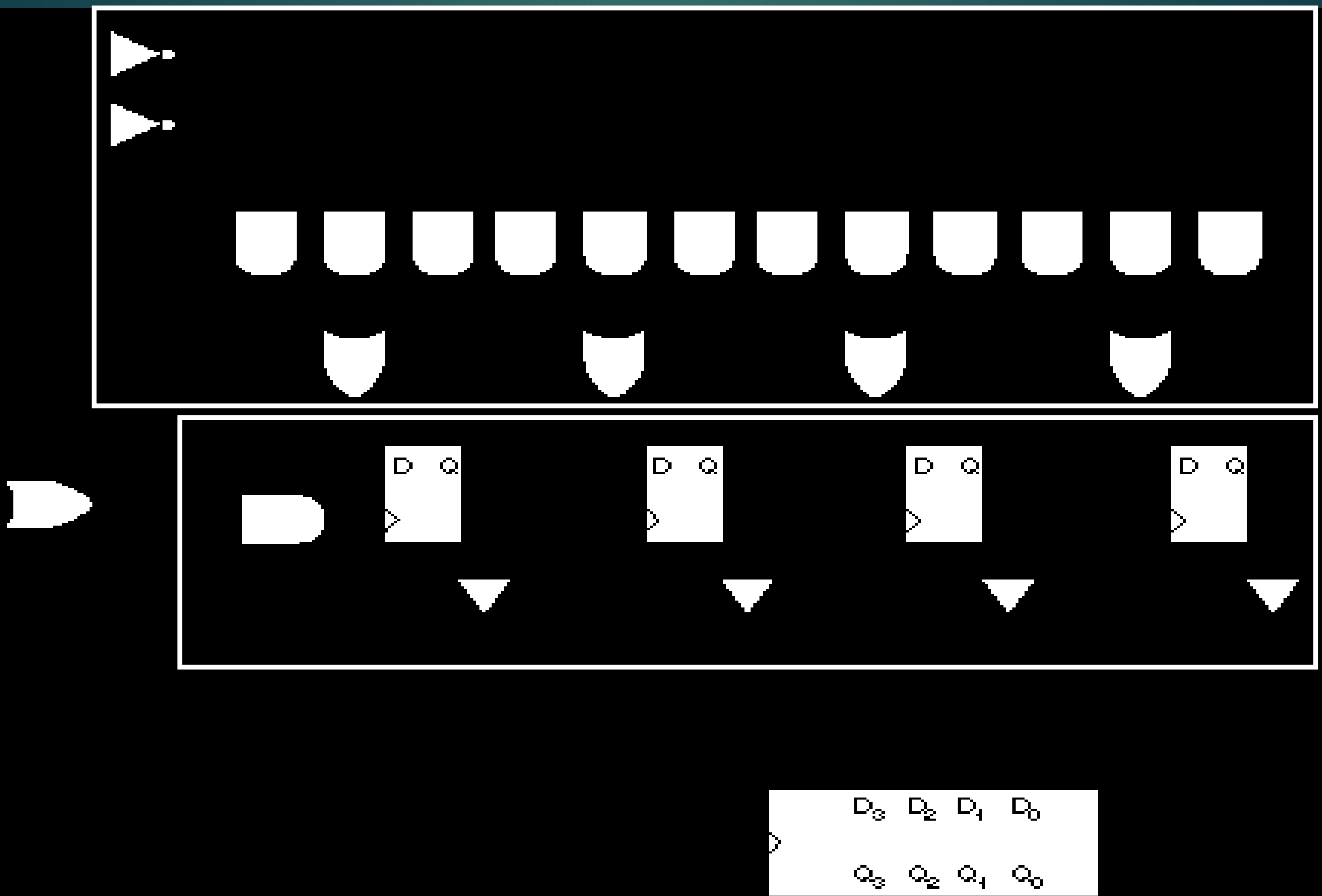
Ejemplo circuito secuencial 1: Registro con desplazamiento

- Es un circuito compuesto por 2 bloques funcionales:
  - El circuito de desplazamiento (parte superior)
  - El circuito de registro (parte inferior)
- En la parte inferior izquierda hay una tabla que indica las funciones del circuito, de acuerdo a las entradas de control.

C1	C0	Función
0	0	sin cambios
0	1	carga de registro con desplazamiento a izquierda
1	0	carga de registro con desplazamiento a derecha
1	1	carga de registro sin desplazamiento

- En la parte inferior derecha de la filmina está el símbolo del circuito.

Ejemplo circuito secuencial 1: Registro con desplazamiento



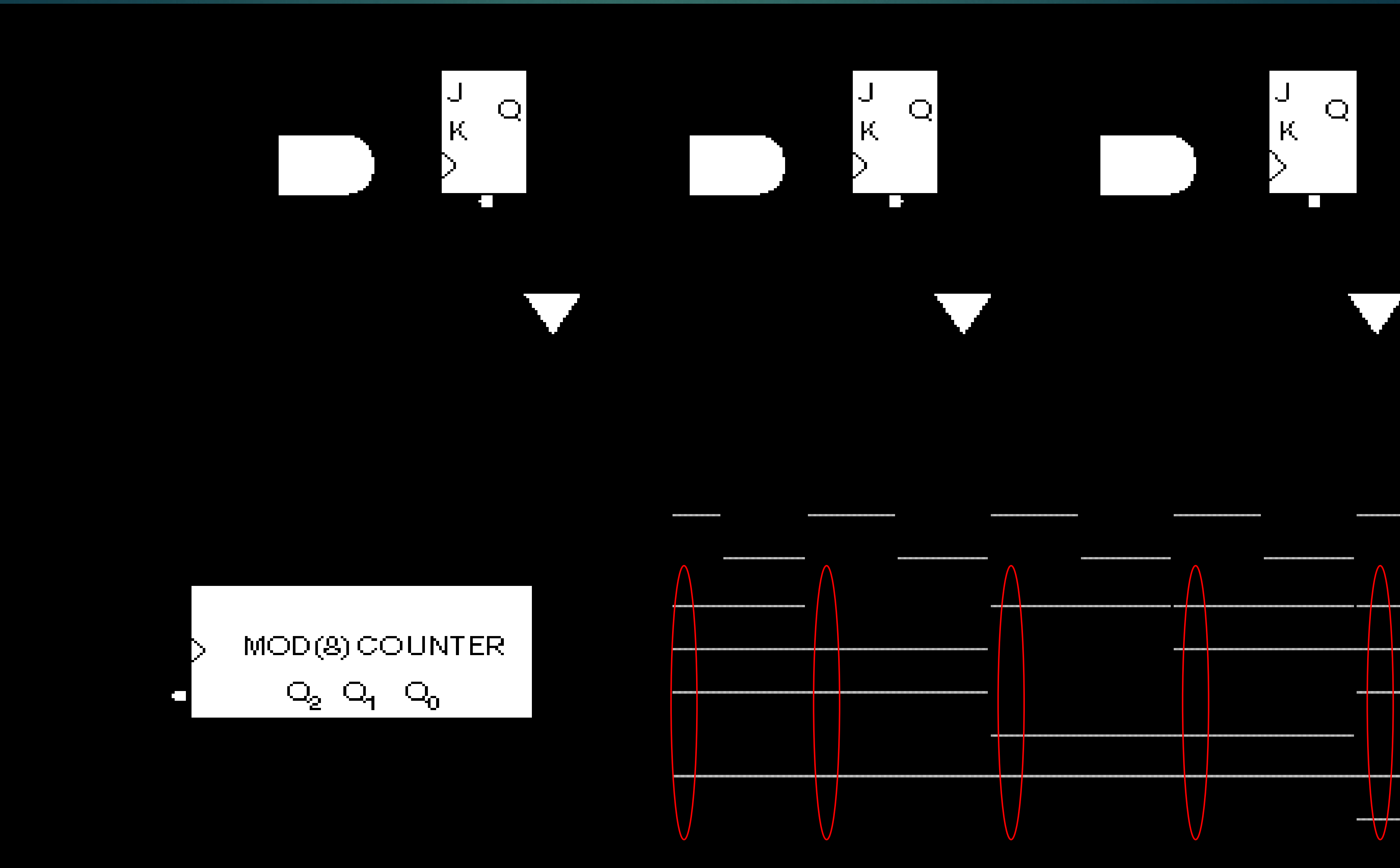
Ejemplo circuito secuencial 2: Contador módulo 8

- Es un circuito secuencial que cuenta en 3 bits (módulo 8), en binario, desde 0 (000) hasta 7 (111).

Q2	Q1	Q0
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

- El contador avanza cada vez que el reloj cambia de 1 a 0.
- La entrada de Reset pone el contador en 0 (000).

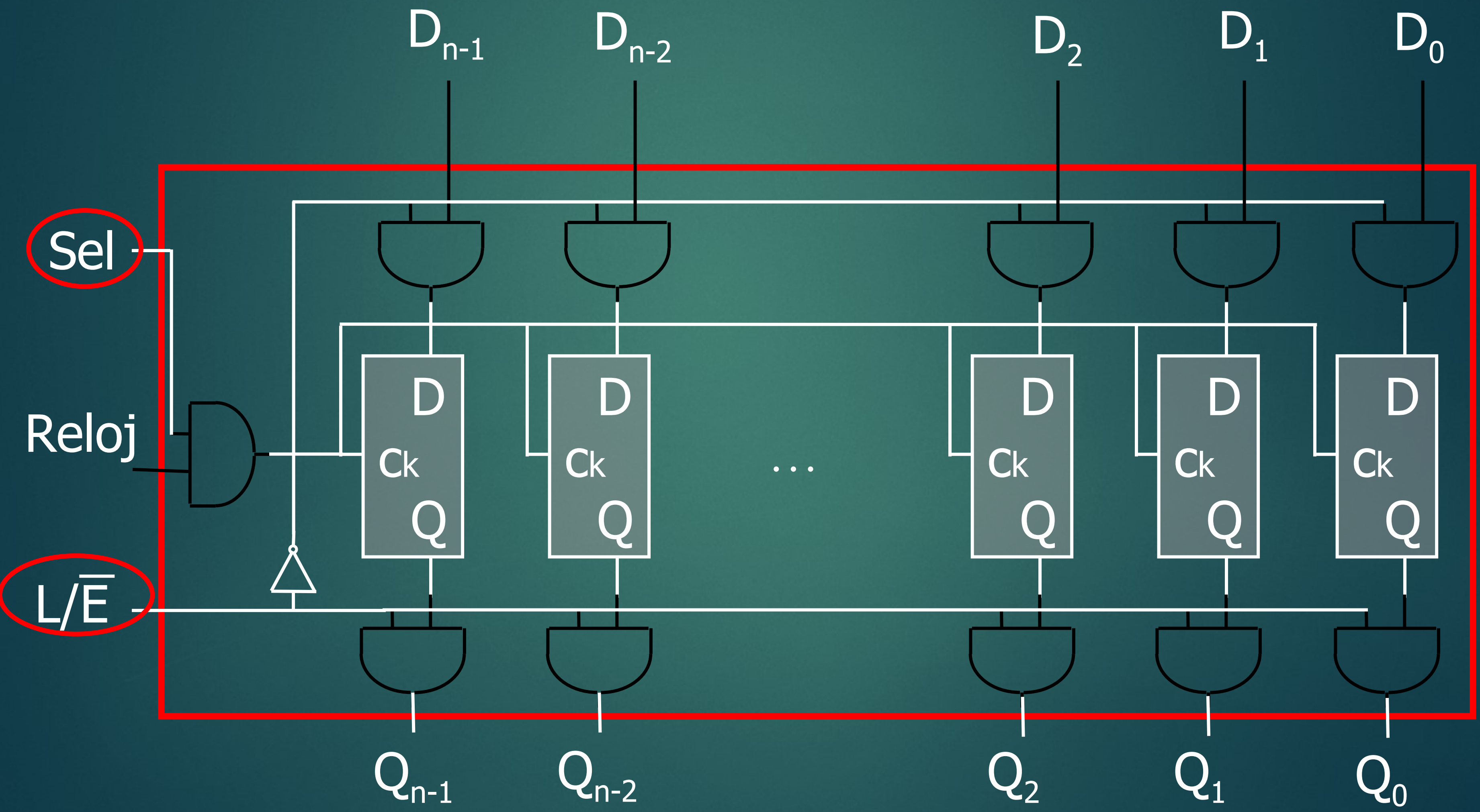
Ejemplo circuito secuencial 2: Contador módulo 8



Registro con entradas de selección y de control del tipo de operación

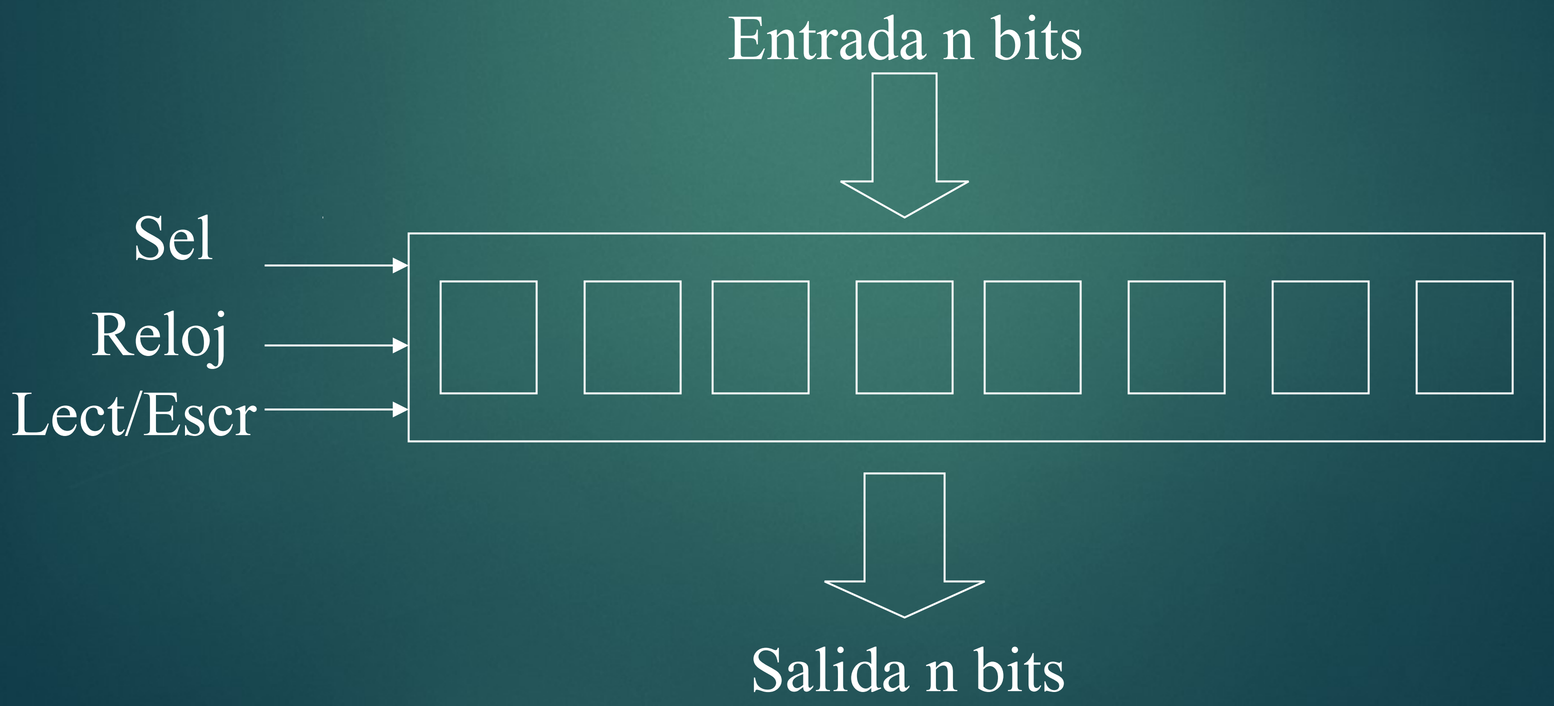
- En general en un registro se pueden realizar 2 acciones:
  - Lectura: leer los n bits del registro (leer todas las salidas Qi simultáneamente).
  - Escritura: escribir los n bits del registros (cargar, en los n registros D simultáneamente, a través de sus entradas Di)
- Para resolver estas 2 acciones se requiere una señal adicional, que indique el tipo de operación a realizar: lectura (L) cuando está en 1, o escritura ($\overline{E}$) cuando está en 0.
- Al registro con entrada de selección le agregamos una entrada de control de operación L/$\overline{E}$.

Registro con entradas de selección y de control del tipo de operación



Modelo elemental de registro de n bits

- El modelo de un registro de n bits se representa como un bloque que contiene n elementos de almacenamiento (n FF D), n entradas de datos, n salidas de datos, una entrada de reloj, una entrada de selección, y una entrada de función.

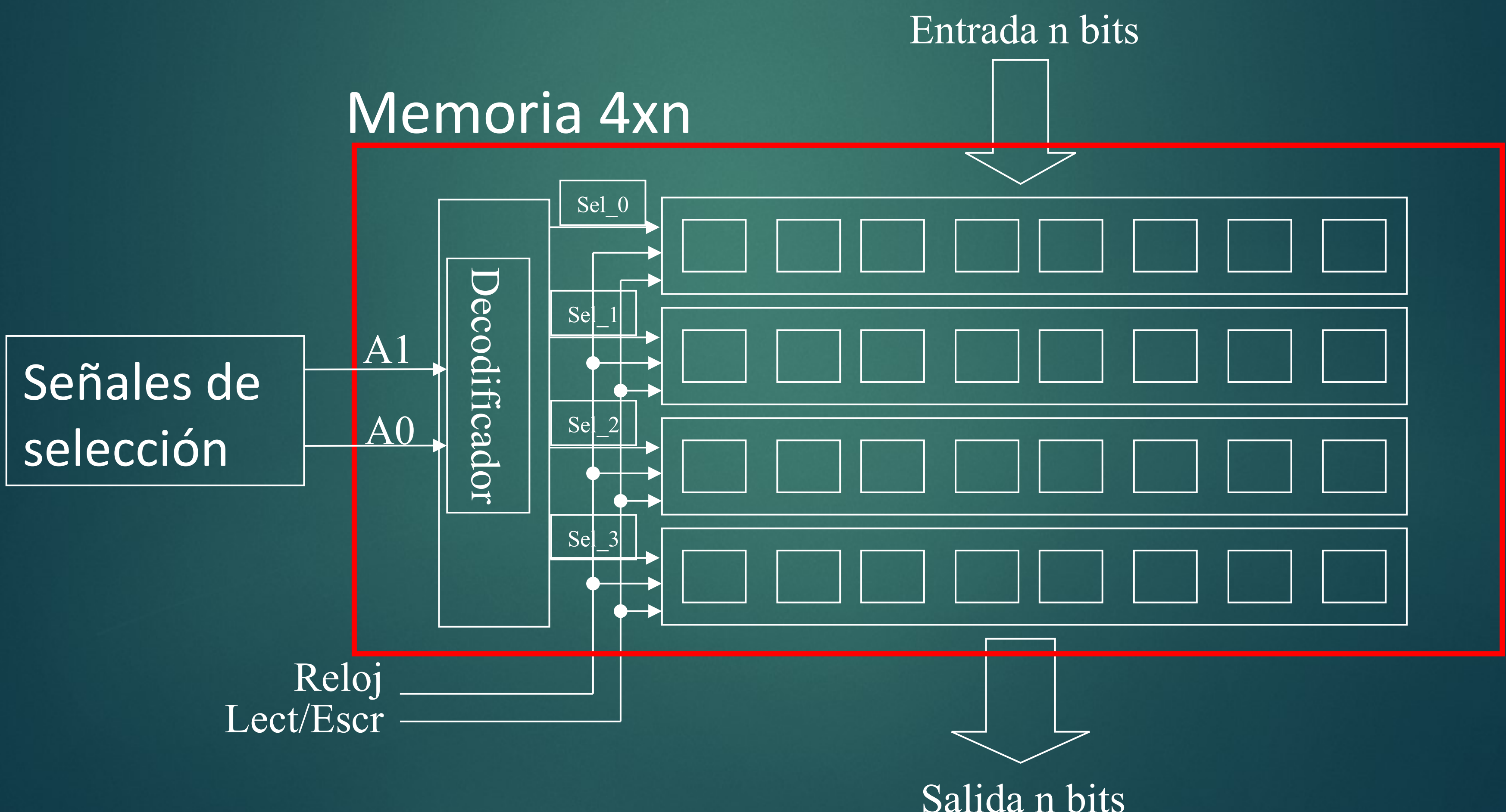


Modelo elemental de memoria de 4 palabras de n bits por palabra

- Una memoria de 4 palabras de n bits por palabra se puede pensar como un arreglo bidimensional de 4 registros de n bits, al que se puede acceder de a uno a la vez.
- Cada palabra es un registro de n bits.
- Las 4 palabras tienen:
  - n entradas de datos comunes a las 4 palabras.
  - n salidas de datos comunes a las 4 palabras.
  - 1 reloj común a las 4 palabras.
  - 1 entrada de función (Lectura/escritura) común a las 4 palabras.
  - 2 entradas para la selección de 1 de las 4 palabras.

Modelo elemental de memoria de 4 palabras de n bits por palabra

- Una memoria de 4 palabras de n bits por palabra (es decir 4xn) se puede representar de la siguiente manera:



Modelo elemental de memoria de 4 palabras de n bits por palabra

- Observar que se proveen 2 señales A0 y A1 para la selección de 1 de los 4 registros (o palabras).
- Con 2 señales de selección se tienen 4 posibles combinaciones:

A1	A0	Sel_0	Sel_1	Sel_2	Sel_3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

- Se usa un <u>decodificador 2 a 4</u> para decodificar las 2 señales de selección A0 y A1 en las 4 Sel_0, Sel_1, Sel_2 y Sel_3 requeridas para seleccionar 1 de los 4 registros.

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