Práctica 4

Lógica y compuertas (Parte 2): Circuitos Combinacionales y Secuenciales

Objetivos de la práctica: que el alumno domine

- ***Circuitos lógicos y diagramas de compuertas***
- ***Introducción a equivalencias lógicas***
- ***método de sumas de productos.***
- ***Describir el funcionamiento de los distintos tipos de flip flops.***
- ***Comprender el funcionamiento de un circuito secuencial.***

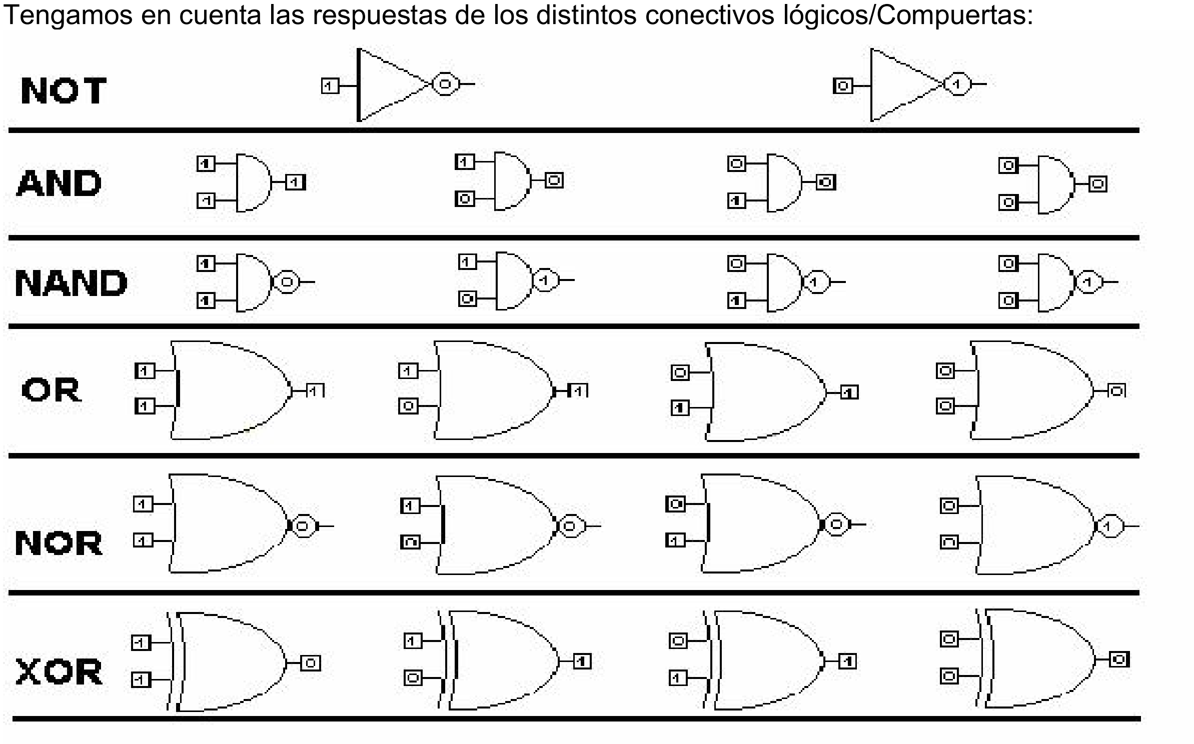
Bibliografía:

- ***"Principios de Arquitectura de Computadoras" de Miles J. Murdocca, apéndice A, pág. 441.***
- ***Apunte 3 de la cátedra, "Sistemas de Numeración: Operaciones Lógicas".***
- ***Apunte 5 de la cátedra, "Circuitos Lógicos Secuenciales".***

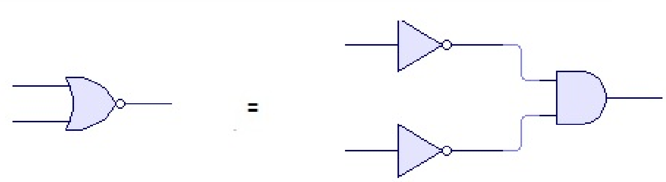
Tener en cuenta para resolución de ejercicios 1 al 5:

Tablas de Verdad: Una tabla de verdad muestra el resultado de una proposición compuesta para cada combinación de valores de verdad que se le puedan asignar a sus componentes de entrada.

Tengamos en cuenta las respuestas de los distintos conectivos lógicos/Compuertas:



Equivalencias lógicas mediante tablas de verdad: Es posible demostrar que dos circuitos son equivalentes si ante iguales entradas responden con el mismo valor de salida. Para llevar a cabo esta demostración, alcanza con construir la tabla de verdad de ambos circuitos y validar que las respuestas coinciden para iguales entradas.
Ejemplo: (La conocida Ley de De Morgan, donde se puede verificar que ante iguales combinaciones de valores de entrada para A y B, la respuesta del circuito es igual en ambos casos)



$\overline{A+B}$

A	B	A+B	$\overline{A+B}$
0	0	0	1
0	1	1	0
1	0	1	0
1	1	1	0

$\overline{A}°\overline{B}$

A	B	$\overline{A}$	$\overline{B}$	$\overline{A}°\overline{B}$
0	0	1	1	1
0	1	1	0	0
1	0	0	1	0
1	1	0	0	0

Otras equivalencias lógicas:

Conjunto cerrado de operaciones lógicas usando sólo compuertas Nand o Nor:
Es posible (su justificación excede el objetivo de este curso) reescribir cualquier expresión lógica compuesta, como una expresión equivalente utilizando EXCLUSIVAMENTE compuertas Nand o Nor. Esto favorece el diseño de circuitos al resolver cualquier lógica con un único tipo de compuertas.

Equivalencias lógicas para representar cualquier conectivo lógico como compuertas Nand:

- $\overline{A} \cong \overline{A} + \overline{A} \cong \overline{A.A}$ (Aplico 2 equivalencias lógicas, la última es la ley de De Morgan).
- $A + B \cong \overline{\overline{A + B}} \cong \overline{\overline{A}.\overline{B}} \cong \overline{\left(\overline{A.A}\right)\left(\overline{B.B}\right)}$ (doble negación, De Morgan, equivalencia anterior para la negación).
- $A.B \cong \overline{\overline{A.B}} \cong \overline{\left(\overline{A.B}\right)\left(\overline{A.B}\right)}$ (doble negación, 1er equivalencia para la negación).
- $A \otimes B \cong \left(A.\overline{B}\right) + \left(\overline{A}.B\right)$..... (definición del or exclusivo, resta aplicar las equivalencias previas para producto, suma y negación para llegar a utilizar sólo compuertas Nand).

El resto de las compuertas pueden reescribirse sólo con compuertas Nand basándose en las equivalencias previas.

---

1. Demostrar mediante tabla de verdad si se cumplen o no las siguientes equivalencias:
→ INCISOS d), e) y f) A RESOLVER

a) $\overline{(A.B)} = \overline{A} + \overline{B}$ (La segunda ley de De Morgan)

A	B	$\overline{(A . B)}$	$\overline{A}$	$\overline{B}$	$\overline{A} + \overline{B}$
0	0	1	1	1	1
0	1	1	1	0	1
1	0	1	0	1	1
1	1	0	0	0	0

SON EQUIVALENTES

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

A	B	C	B . C	A + (B . C)	A + B	A + C	(A + B) + (A + C)
0	0	0	0	0	0	0	0
0	0	1	0	0	0	1	1
0	1	0	0	0	1	0	1
0	1	1	1	1	1	1	1
1	0	0	0	1	1	1	1
1	0	1	0	1	1	1	1
1	1	0	0	1	1	1	1
1	1	1	1	1	1	1	1

NO SON EQUIVALENTES

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

A	B	C	A + B	(A + B) . C	A . B	A . C	(A . B) + (A . C)
0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0
0	1	0	1	0	0	0	0
0	1	1	1	1	0	0	0
1	0	0	1	0	0	0	0
1	0	1	1	1	0	1	1
1	1	0	1	0	1	0	1
1	1	1	1	1	1	1	1

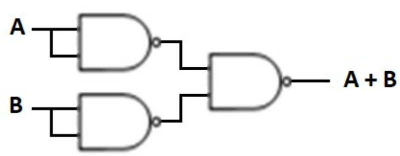
NO SON EQUIVALENTES

d) $A + A + B = A + B + B$
e) $A + B.C = A.C + B$
f) $A \oplus B = \overline{A} \oplus \overline{B}$

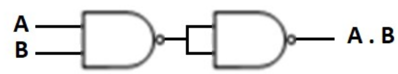
2. Modifique los siguientes circuitos para que sean todas compuertas **NAND**.

NOT: $\overline{A} = \overline{A.A}$

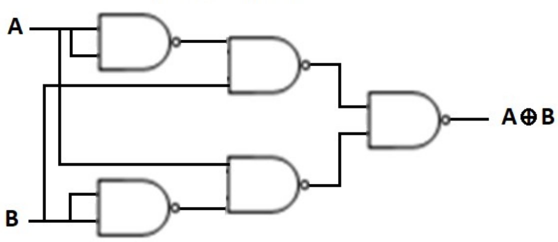
OR: $A + B = \overline{\overline{A + B}} = \overline{\overline{A}.\overline{B}}$



AND: $A.B = \overline{\overline{A.B}}$

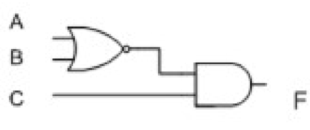


XOR: $A \oplus B = (\overline{A}.B) + (A.\overline{B})$

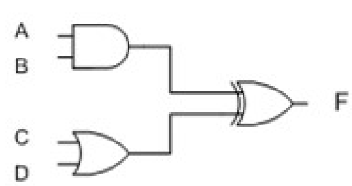


Nota: para lograr este circuito primero se reemplaza con los circuitos con compuertas NAND que corresponden a cada operacion y luego se cancelan las negaciones consecutivas.

1)

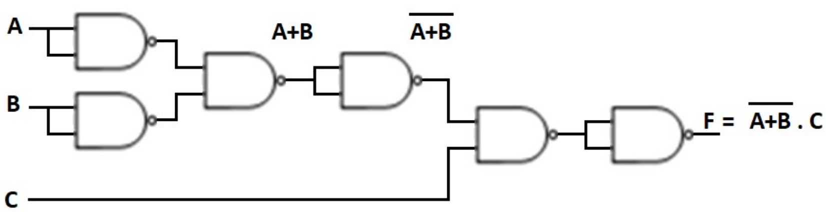


2)

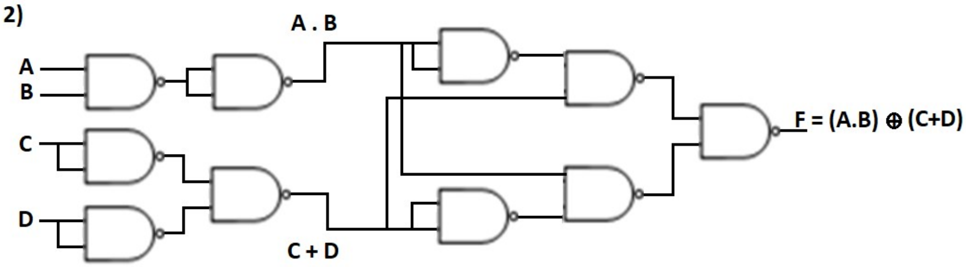


Organización de Computadoras 2020

1)

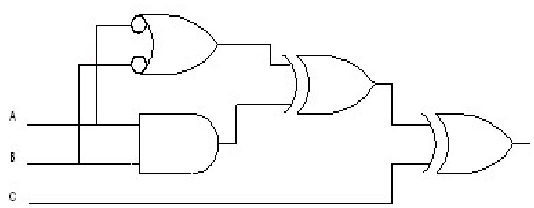


2)



3. Reescriba las compuertas lógicas Not, Or, And y Xor utilizando exclusivamente compuertas **NOR**. (Ver como se resolvió el mismo caso para compuertas Nand, en *Tener en . . .*).
→ EJERCICIO A RESOLVER

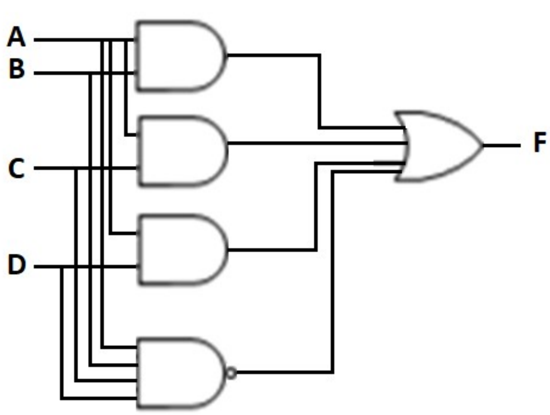
4. Construya la tabla de verdad del siguiente circuito. Analice los valores y basándose en sus conclusiones construya un diagrama más simple que implemente la misma función de salida. Escriba además la ecuación de salida en forma de función.
→ EJERCICIO A RESOLVER



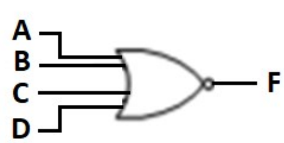
5. Dadas las siguientes relaciones, dibuje los diagramas de compuertas que cumplen con ellas. Modifíquelos utilizando sólo compuertas **NOR**. Modifíquelos utilizando sólo compuertas **NAND**.
→ INCISOS c) y d) A RESOLVER

a) $F = AB + AC + AD + \overline{ABCD}$ b) $F = \overline{A + B + C + D}$ c) $F = \overline{A + B\overline{C}} + C$ d) $F = A\overline{B} + \overline{A}B$

a)

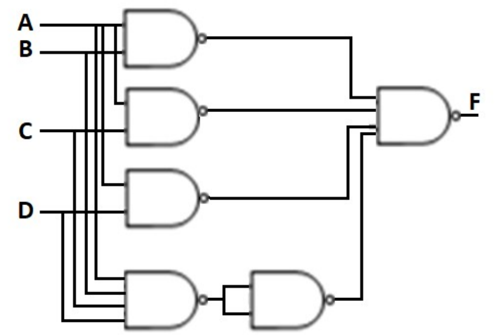


b)

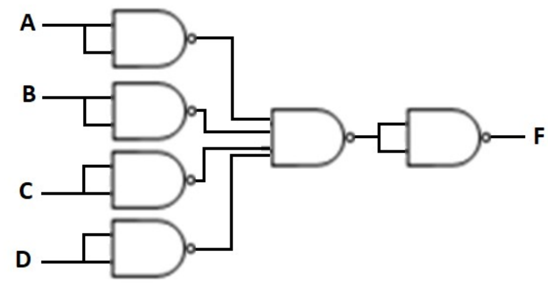


NAND:

a)



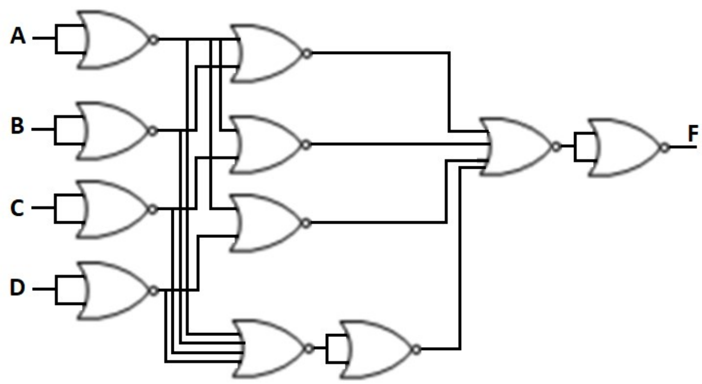
b)



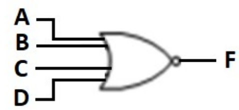
Nota: se cancelaron negaciones consecutivas

NOR:

a)



b)



Nota: queda igual al original

Tener en cuenta para ejercicios 6 al 8:

Suma de Productos: Es posible inferir la fórmula lógica asociada a una función desconocida de la cual sólo se conoce la respuesta ante todas las combinaciones posibles de entradas….
Ejemplo: Supongamos una función que recibe 2 parámetros A y B, si conocemos la respuesta F de la ecuación en base a los posibles valores de A y B mediante la siguiente tabla de verdad:

A	B	F
0	0	**1**
0	1	**1**
1	0	**0**
1	1	**1**

¿En qué casos la salida F será 1? Rta: Cuando las entradas sean A=0 **y** B=0, **o** A=0 **y** B=1, **o** A=1 **y** B=1.
Dicho de otra manera, podemos interpretar como respuesta válida que F será 1 cuando no ocurra A y no ocurra B, o no ocurra A y sí ocurra B, o cuando ocurran A y B.
Esto que es tan simple de entender en lenguaje cotidiano, se traslada con el mismo concepto a la idea de suma de productos, considerando que estamos haciendo una Disyunción/Suma (con la simbología que deseemos: O, Or, ˅, +) de Conjunciones/Productos (simbología: y, And, ˄, .). En conclusión podemos inferir de la anterior tabla de verdad lo siguiente:

$\boxed{F = \overline{A}.\overline{B} + \overline{A}.B + A.B}$ (Por convención y de manera análoga a las operaciones aritméticas conocidas entendemos que ante la ausencia de paréntesis se calculan primero los productos y luego las sumas con los resultados intermedios de cada producto).

Para validar la veracidad de lo expuesto, se debe armar la tabla de verdad de la proposición compuesta y comprobar que coinciden las salidas para todas las combinaciones posibles de la tabla original.

Imaginemos ahora una función que recibe 4 variables A,B,C,D que representan los 4 dígitos de un número binario (Siendo D el menos significativo hasta A como más significativo)….Respondamos ahora la siguiente pregunta:
¿Cuándo viene representado el número 5? (Sabemos que el 5 se representa en binario como 0101)

Rta: cuando viene A=0 **y** B=1 **y** C=0 **y** D=1. O dicho de otra manera, cuando NO ocurra A y SI ocurra B y NO ocurra C y SI ocurra D.

Conclusión: Se puede representar una ecuación que retorne 1 cuando en las cuatro entradas reciba el número 5, de la siguiente manera: $F_5 = \overline{A}.B.\overline{C}.D$ (Notar que la salida F_5 tomará valor 1 exclusivamente cuando las entradas ABCD sean 0101)

Ahora estamos preparados para determinar una ecuación que, por ejemplo, retorne 1 cuando el número representado en las cuatro entradas sea 5 **o** 7 **o** 9 (es decir 0101 **o** 0111 **o** 1001)

$F = \overline{A}.B.\overline{C}.D + \overline{A}.B.C.D + A.\overline{B}.\overline{C}.D$ (Notar que la salida F tomará el valor 1 exclusivamente cuando las entradas sean alguna de las 3 definidas, en cualquier otra combinación de entrada, la ecuación retornará 0).

---

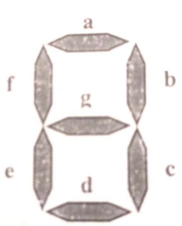
6. Para la siguiente tabla de verdad encuentre una fórmula lógica correspondiente (utilizando suma de productos).

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

$$F = (\overline{A}\,.\,\overline{B}\,.\,C) + (\overline{A}\,.\,B\,.\,\overline{C}) + (A\,.\,\overline{B}\,.\,C)$$

7. Diseñe un circuito que tenga como entrada código BCD empaquetado (4 entradas) y 7 salidas para controlar los 7 segmentos de un display numérico, siendo la salida para los segmentos '0' para apagado y '1' para prendido. Construya la tabla de verdad y la ecuación de la salida correspondiente a los segmentos **a, b, c, d, e, f** y **g**.

→ COMPLETAR LA TABLA DE VERDAD Y EL RESTO DE LAS ECUACIONES

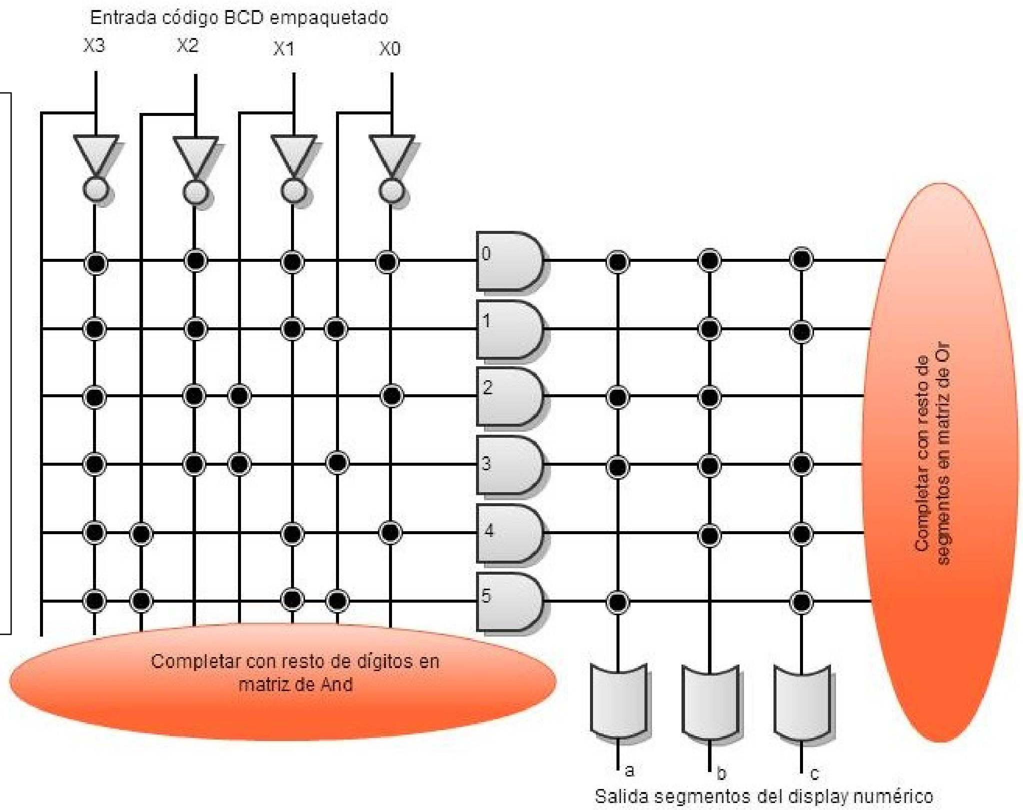


Ayuda 1: Cada segmento se considera como una salida distinta, y cada uno se debe activar (poner en 1) dependiendo del número recibido en las entradas que representan los 4 bits de un BCD empaquetado.
Ejemplo: El segmento **b** se debe activar cuando se recibe un 1 (0001), o un 2 (0010), o un 3 (00110, o un 4 (0100), o un 7 (0111), o un 8 (1000), o un 9 (1001). Se aplica la misma idea con el resto de las salidas.

	A	B	C	D	a	b	c	d	e	f	g
0	0	0	0	0	1	1	1	1	1	1	0
1	0	0	0	1	0	1	1	0	0	0	0
2	0	0	1	0	1	1	0	1	1	0	1
3	0	0	1	1					0		
4	0	1	0	0					0		
5	0	1	0	1					0		
6	0	1	1	0					1		
7	0	1	1	1					1		
8	1	0	0	0					1		
9	1	0	0	1					0		

$$e = \bar{A}.\bar{B}.\bar{C}.\bar{D} + \bar{A}.\bar{B}.C.\bar{D} + \bar{A}.B.C.\bar{D} + \bar{A}.B.C.D + A.\bar{B}.\bar{C}.\bar{D}$$

Ayuda 2: Gráficamente, el circuito con las 4 entradas y las 7 salidas conviene diseñarlo como una matriz de compuertas And, seguida de la matriz de compuertas Or (basarse en la siguiente gráfica parcial del circuito:



8. Un controlador de proceso industrial recibe como entrada tres señales de temperatura T1, T2, T3 (T1<T2<T3) que adoptan el valor lógico ‘1’ cuando la temperatura es mayor que t1, t2 y t3 respectivamente. Diseñar un circuito que genere una señal F cuando la temperatura esté comprendida entre t1 y t2 o cuando la temperatura sea mayor que t3.
→ EJERCICIO A RESOLVER

Tener en cuenta para ejercicios 9 al 13:

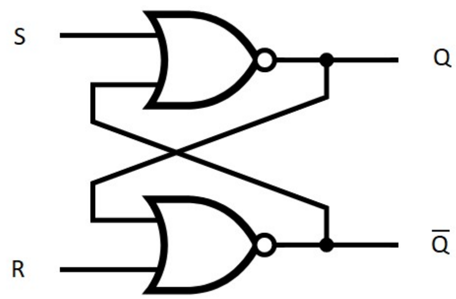
Circuitos Secuenciales: (repasar apuntes de la cátedra y teoría)

- Flip flop S-R asincrónico:
  - Problemas de sincronismo ante cambios de entrada durante el cálculo.
  - Reacción frente a doble entrada de 1’s.
- Flip flop S-R sincrónico:
  - Resuelve problema de sincronismo, pero mantiene problema ante doble entrada de 1’s.
- Flip flop D:
  - Pequeña variante del S-R que resuelve el problema de la doble entrada de 1’s.
- Flip flop J-K:
  - Incorpora posibilidad de alterar el valor previo (complemento lógico).
- Flip flop T:
  - Pequeña variante del J-K, que sólo se dedica a invertir su valor ante cada orden del clock.

---

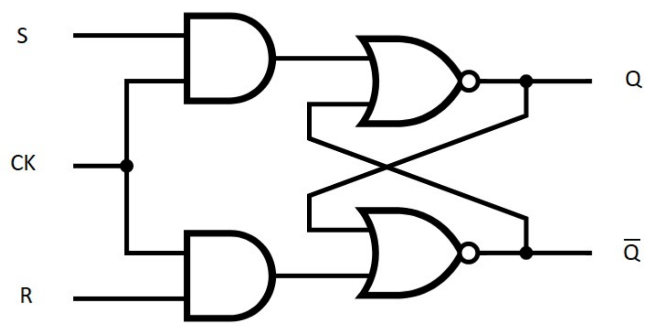
9. Dibuje el esquema de compuertas que componen un flip flop S-R. Describa a través de una tabla los estados en función de las entradas. Modifique el esquema anterior para hacerlo sincrónico. Describa gráficamente su respuesta temporal.

SR



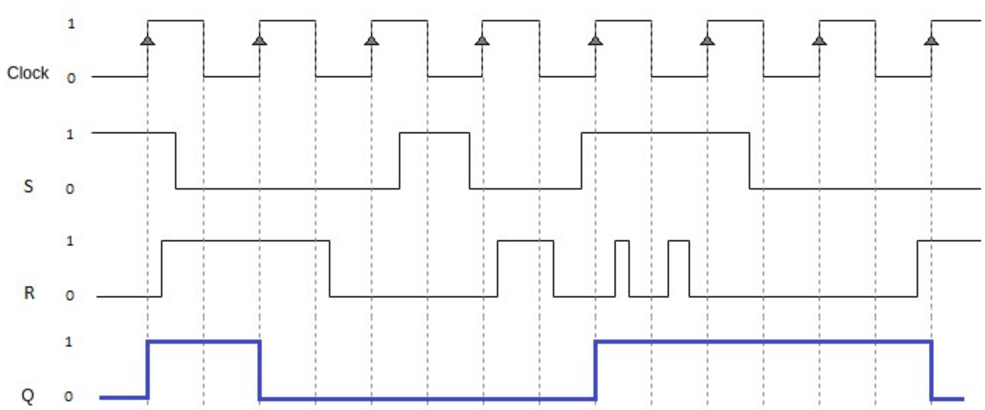
S	R	Q_{N+1}
0	0	Q_N
0	1	0
1	0	1
1	1	PROHIBIDO

SR SINCRONICO:

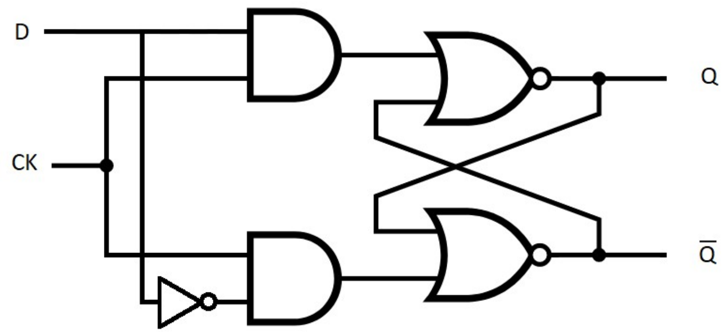


Clock	S	R	Q_{N+1}
1↑	0	0	Q_N
1↑	0	1	0
1↑	1	0	1
1↑	1	1	PROHIBIDO
0	-	-	Q_N

SR SINCRONICO: RESPUESTA TEMPORAL.
ACTIVADO POR FLANCO ASCENDENTE (CLOCK ↑)

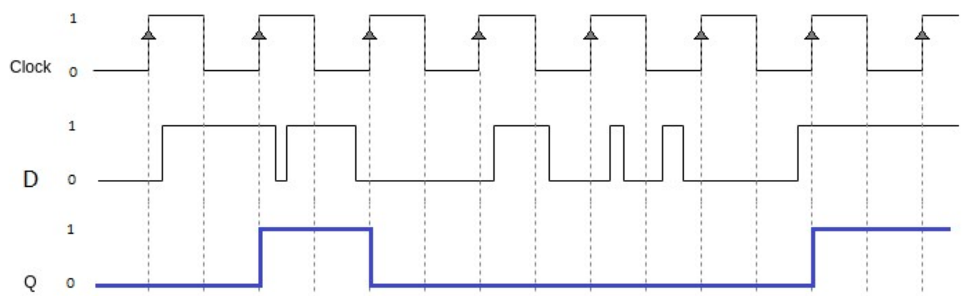


10. Dibuje el esquema de un flip flop D. Detalle en su respuesta temporal como resuelve el problema de la doble entrada de 1's que se presentaba en el S-R.

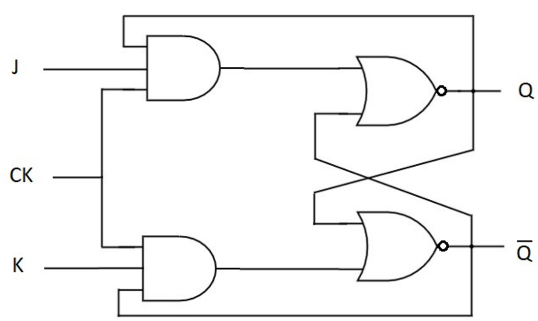


Clock	D	Q_{N+1}
1↑	0	0
1↑	1	1
0	-	Q_N

RESPUESTA TEMPORAL. ACTIVADO POR FLANCO ASCENDENTE (CLOCK ↑)

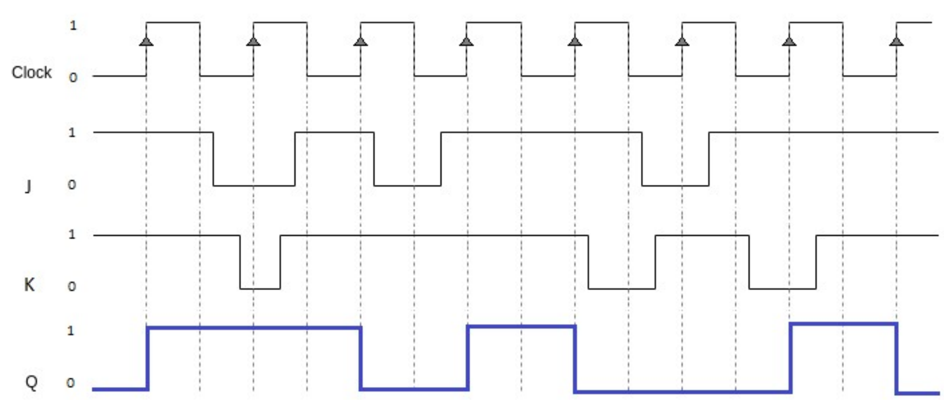


11. Dibuje el esquema de un flip flop J-K, describiendo su respuesta temporal.

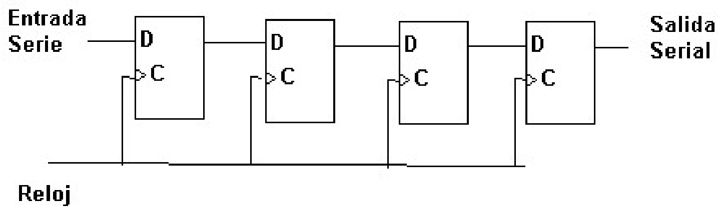


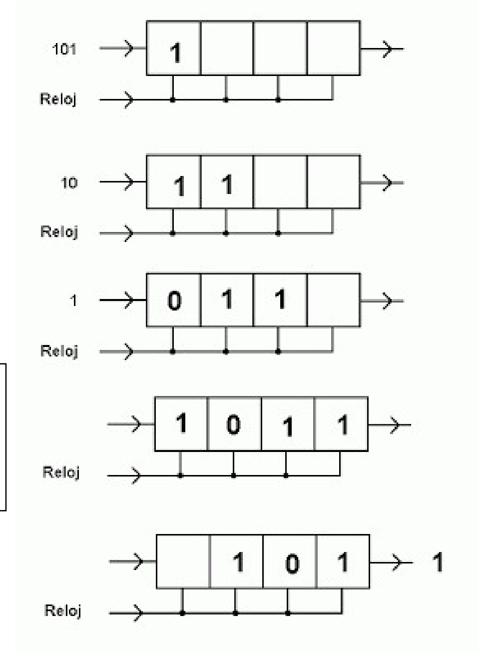
Clock	J	K	Q_{N+1}
1↑	0	0	Q_N
1↑	0	1	0
1↑	1	0	1
1↑	1	1	$\overline{Q_N}$
0	-	-	Q_N

RESPUESTA TEMPORAL. ACTIVADO POR FLANCO ASCENDENTE (CLOCK ↑)



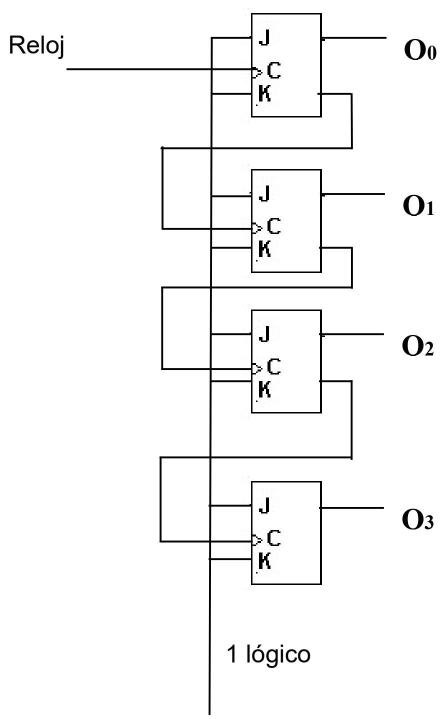
12. Dibuje el diagrama de tiempos del registro de la figura, implementado con flip flops D. Modifíquelo para desplazamiento izquierda derecha y derecha izquierda. **→ EJERCICIO A RESOLVER**



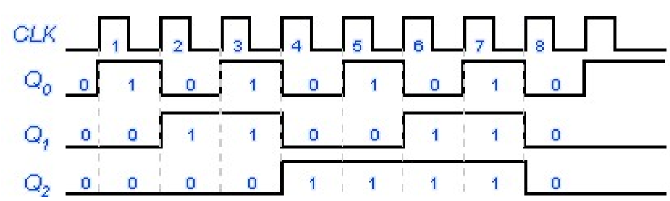


Ayuda: Ejemplo de respuesta temporal para interpretar como responde el registro previo ante la entrada serial del número binario 1011:

13. Describa gráficamente la respuesta temporal de cada flip flop ante una señal de unos y ceros entrando por Reloj. **→ EJERCICIO A RESOLVER**



Ayuda: El diagrama correspondiente considerando sólo los primeros 3 flip-Flops es el siguiente:



Se observa como la respuesta de cada flip-flop emite una onda a la mitad de frecuencia que su clock de entrada.