

870116

# Universidad Autónoma de Guadalajara

INCORPORADA A LA UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

FACULTAD DE INGENIERIA

ESCUELA DE INGENIERIA EN COMPUTACION

J<sup>2</sup>  
Ej...



TESIS CON  
FALLA DE ORIGEN

TRABAJO DE TESIS

TEXTO PROGRAMADO SOBRE FLIP - FLOP'S

PRESENTADO POR  
 FABIOLA SOTO LEMUS  
 PARA CONFERIRLE EL TITULO DE  
 INGENIERO EN COMPUTACION  
 Guadalajara, Jal., Diciembre de 1989



## **UNAM – Dirección General de Bibliotecas Tesis Digitales Restricciones de uso**

### **DERECHOS RESERVADOS © PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

Todo el material contenido en esta tesis está protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

## CONTENIDO

INTRODUCCION	1
ANTECEDENTES	3
<b>CAPITULO 1. <u>APRENDIENDO SOBRE FLIP-FLOP'S</u></b>	<b>4</b>
1.1 REQUISITOS PREVIOS	6
1.2 FLIP-FLOP'S	10
1.2.1 FLIP-FLOP'S CON COMPUERTAS NOR	11
1.2.2 FLIP-FLOP'S CON COMPUERTAS NAND	14
1.2.3 SEÑALES DE RELOJ	17
1.2.4 FLIP-FLOP'S RS CON RELOJ O TEMPORIZADOS	18
1.2.5 PARAMETROS DE ESTABLECIMIENTO Y MANTENIMIENTO	19
1.2.6 FLIP-FLOP D CON RELOJ O TEMPORIZADO	21
1.2.7 FLIP-FLOP JK CON RELOJ O TEMPORIZADO	22
1.2.8 FLIP-FLOP T O DE CONMUTACION	23
1.2.9 FLIP-FLOP MAESTRO-ESCLAVO	24
1.2.10 ENTRADAS DIRECTAS DC SET Y DC CLEAR	30
1.2.11 TABLAS DE EXCITACION DE LOS FLIP-FLOP'S	31
1.3 VIBRADOR MULTIPLE MONOESTABLE	32
<b>CAPITULO 2. <u>EJERCICIOS CON FLIP-FLOP'S</u></b>	<b>34</b>
<b>CAPITULO 3. <u>APLICACIONES DE LOS FLIP-FLOP'S</u></b>	<b>51</b>
3.1 TRANSFERENCIA	52
3.2 SUMADORES	54
3.3 DIVISOR DE FRECUENCIA	57
3.4 CONTADORES BINARIOS	59
3.5 MEMORIAS	60
3.6 SEPARADOR FIFO (FIRST INPUT FIRST OUTPUT)	62
3.7 REGISTRO SENSIBLE A MULTIPLES ORDENES	64
<b>CAPITULO 4. <u>EJERCICIOS APLICANDO FLIP-FLOP'S</u></b>	<b>66</b>
CONCLUSIONES	79
APENDICES	80
A. NUMEROS BINARIOS	81
B. TEOREMAS BASICOS DEL ALGEBRA DE BOOLE	83
C. COMPUERTAS DIGITALES LOGICAS	84
D. FUNCIONES BOOLEANAS	86
E. CIRCUITOS COMBINACIONALES	88
BIBLIOGRAFIA	89

## CONTENIDO

### INTRODUCCION

### ANTECEDENTES

### CAPITULO 1. APRENDIENDO SOBRE FLIP-FLOP'S

#### 1.1 REQUISITOS PREVIOS

#### 1.2 FLIP-FLOP'S

1.2.1 FLIP-FLOP'S CON COMPUERTAS NOR

1.2.2 FLIP-FLOP'S CON COMPUERTAS NAND

1.2.3 SEÑALES DE RELOJ

1.2.4 FLIP-FLOP'S RS CON RELOJ O TEMPORIZADOS

1.2.5 PARAMETROS DE ESTABLECIMIENTO Y MANTENIMIENTO

1.2.6 FLIP-FLOP D CON RELOJ O TEMPORIZADO

1.2.7 FLIP-FLOP JK CON RELOJ O TEMPORIZADO

1.2.8 FLIP-FLOP T O DE CONMUTACION

1.2.9 FLIP-FLOP MAESTRO-ESCLAVO

1.2.10 ENTRADAS DIRECTAS DC SET Y DC CLEAR

1.2.11 TABLAS DE EXCITACION DE LOS FLIP-FLOP'S

#### 1.3 VIBRADOR MULTIPLE MONOESTABLE

### CAPITULO 2. EJERCICIOS CON FLIP-FLOP'S

### CAPITULO 3. APLICACIONES DE LOS FLIP-FLOP'S

3.1 TRANSFERENCIA

3.2 SUMADORES

3.3 DIVISOR DE FRECUENCIA

3.4 CONTADORES BINARIOS

3.5 MEMORIAS

3.6 SEPARADOR FIFO (FIRST INPUT FIRST OUTPUT)

3.7 REGISTRO SENSIBLE A MULTIPLES ORDENES

### CAPITULO 4. EJERCICIOS APLICANDO FLIP-FLOP'S

### CONCLUSIONES

### APENDICES

A. NUMEROS BINARIOS

B. TEOREMAS BASICOS DEL ALGEBRA DE BOOLE

C. COMPUERTAS DIGITALES LOGICAS

D. FUNCIONES BOOLEANAS

E. CIRCUITOS COMBINACIONALES

### BIBLIOGRAFIA

## INTRODUCCION

A través de la historia de la educación se ha visto el esfuerzo que han hecho diferentes personas para motivar al alumno en el aprovechamiento del material de estudio. Se han desarrollado diferentes y muy variadas técnicas grupales, materiales didácticos, etc. pero se ha visto también que no todas las técnicas empleadas o materiales didácticos usados causan la misma impresión a todos los alumnos. Algunos alumnos captan bastante rápido el tema que se está exponiendo, otros lo harán no muy rápido y en algunos casos, será difícil para el alumno asimilar la clase, por la velocidad a la que ésta se imparte.

Es muy común que los maestros se enfrenten con algunas limitaciones provocadas por razones ajenas a ellos, por ejemplo, el tiempo en que deben cubrir los programas, la poca o nada homogeneidad de los grupos, etc.

Para dar solución, hasta donde es posible, a éstos problemas pensó en la elaboración de un texto programado, el cual ofrece muchas ventajas tanto al estudiante como al profesor.

Las ventajas que podemos encontrar en un texto programado son:

- \* Se hace que el alumno participe activamente en el texto. El alumno "aprende haciendo".
- \* La velocidad de la instrucción es controlada por el alumno. El estudiante progresa en pasos y no puede continuar hasta tener la competencia necesaria en cada ejercicio.
- \* El texto programado, por ser portable, puede estar siempre disponible para el estudiante.
- \* Los profesores pueden tener la garantía virtual de que los alumnos que completen el programa, habiendo seguido las "reglas" del mismo, tendrán dominio del material tratado.

Con lo dicho anteriormente vemos como se resuelven nuestros problemas; esto no quiere decir que siempre deba usarse un texto programado, pero sí que es muy útil cuando tenemos los

problemas mencionados.

El texto programado es recomendable para usuario como material de apoyo, en ninguna manera el texto sustituye al maestro.

## ANTECEDENTES

La forma en que se desarrolló este texto programado fue a través de cuatro capítulos y una serie de apéndices. El material presentado es sencillo y de fácil comprensión.

En el capítulo uno inicio con una sección de requisitos previos en la cual presento al alumno varios problemas con los que pretendo que el alumno tenga un recordatorio de algunos conceptos básicos importantes. En seguida presento una sección teórica en la que expongo qué son los flip-slop's y algunos tipos de éstos. El material es presentado con bastantes dibujos como recurso didáctico para explicar mejor el tema.

En el capítulo dos presento una serie de ejercicios de diferentes grados de dificultad y su respectiva solución para ayudar al alumno a resolver dudas.

En el capítulo tres muestro algunas aplicaciones prácticas de los flip-flop's. No he profundizado en cada aplicación del tema porque no es el objetivo del texto, solo se han presentado con el fin de que el alumno pueda darse cuenta de la manera en que pueden aplicarse y del papel tan importante que juegan estos dispositivos en diferentes circuitos sincrónicos.

En el capítulo cuatro le presento al alumno una serie de ejercicios en los que pretendo que practique lo que ha aprendido a través del texto y que resuelva posibles dudas.

Finalmente incluyo una sección de apéndices con el objetivo de que el alumno pueda revisar algunos conceptos importantes en caso de que los necesite.

# CAPITULO 1

## NOTA PARA EL USUARIO DEL TEXTO PROGRAMADO

Este texto programado está diseñado para hacer el estudio de este tema interesante y sencillo. El texto requiere que el estudiante ponga de su parte y que se sujete a las "reglas" del mismo.

Dentro de las reglas que hay en este texto tenemos:

\* El estudiante ha de resolver los requisitos previos en el primer capítulo para empezar con la teoría.

\* Al resolver cada problema en el texto, no se debe ver la respuesta antes de resolverlo, y además, debe ver únicamente la respuesta del ejercicio resuelto.

\* Para entrar a alguna sección de problemas, se ha de poner la máscara en la hoja que sigue a la de preguntas (la de respuestas) y se ha de ir deslizando conforme se vayan resolviendo los ejercicios.

\* Utilizar una hoja tamaño carta gruesa como máscara de tal forma que no pueda verse a través de ella.

AHORA VAMOS A INICIAR, INSERTA LA MASCARA EN LA HOJA QUE SIGUE A LA HOJA DE PROBLEMAS, AHORA PASA A LA PAGINA SIGUIENTE Y CONTESTA LO QUE SE TE PIDE.

## 1.1 REQUISITOS PREVIOS

### Problema 1.

Convertir el siguiente número decimal a binario:

$$1932 = \underline{\hspace{2cm}}$$

Desliza la máscara hacia abajo a la línea y checa tu respuesta.

---

### Problema 2.

Convertir el siguiente número binario a decimal:

$$10110001110 = \underline{\hspace{2cm}}$$

Desliza la máscara hacia abajo otra vez para checar tu respuesta

---

### Problema 3.

Cuáles son las tablas de verdad de las operaciones, AND, OR, y NOT ?

A	B	A . B	A	B	A + B	A	A'
---	---	-------	---	---	-------	---	----

Respuesta 1.

1932	0	
966	0	
483	1	
241	1	
120	0	1932 = <u>11110001100</u>
60	0	
30	0	
15	1	
7	1	
3	1	
1	1	

DEJA AQUÍ LA MASCARA Y CONTESTA EL PROBLEMA 2

Respuesta 2.

$$\begin{aligned}10110001110 &= 2^{10} + 2^8 + 2^7 + 2^3 + 2^2 + 2^1 \\ &= 1024 + 256 + 128 + 8 + 4 + 2 \\ &= \underline{1422}\end{aligned}$$

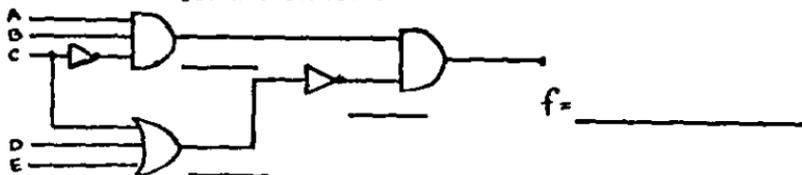
Respuesta 3.

<u>A</u>	<u>B</u>	<u>A . B</u>	<u>A</u>	<u>B</u>	<u>A + B</u>	<u>A</u>	<u>A'</u>
0	0	0	0	0	0	0	1
0	1	0	0	1	1	1	0
1	0	0	1	0	1		
1	1	1	1	1	1		

OTRA VEZ INSERTA LA MASCARA 2 HOJAS ADELANTE DE ESTA, AHORA PASA A LA SIGUIENTE PAGINA.

PROBLEMA 4.

Escribir la expresión algebraica que se produce en cada compuerta en el siguiente circuito:



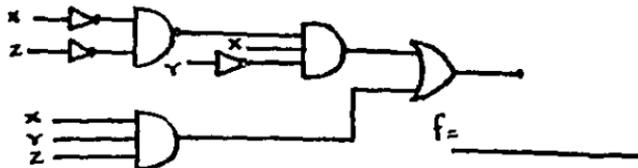
PROBLEMA 5.

(DE)'f

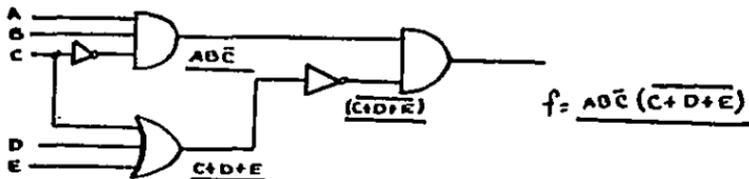
Dibuja el circuito que produce la salida  $f = [(A + B)C +$

PROBLEMA 6.

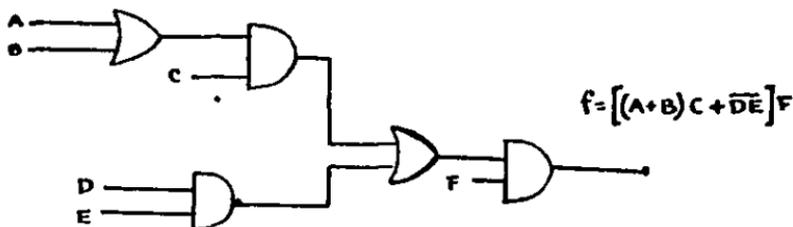
Simplifica el siguiente circuito:



Respuesta 4.



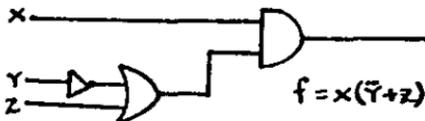
Respuesta 5.



Respuesta 6.

$$\begin{aligned}
 f &= (X'Z')' \cdot XY' + XYZ \\
 &= (X''+Z'')XY' + XYZ \\
 &= (X + Z)XY' + XYZ \\
 &= XX'Y' + XY'Z + XYZ \\
 &= XY' + XY'Z + XYZ \\
 &= XY' + XZ(Y + Y') \\
 &= XY' + XZ \\
 &= X(Y' + Z)
 \end{aligned}$$

El circuito correspondiente es:



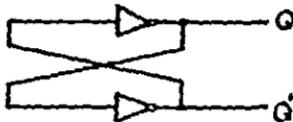
## 1.2 FLIP-FLOP'S

Los flip-flop's (FF's) son celdas binarias con la capacidad de guardar o almacenar un bit de información. Podemos representar el flip-flop en forma general de la siguiente manera:



Como se puede ver, en cada flip-flop tenemos dos salidas, cada una invertida respecto a la otra. En cuanto a los estados posibles de operación, solo tenemos dos en los FF's:  $Q=0$ ,  $Q'=1$  y  $Q=1$ ,  $Q'=0$ . Estos estados pueden mantenerse indefinidamente mientras tengamos potencia en el circuito o hasta que haya un cambio en alguna de las entradas que modifique el estado del FF.

Un FF puede ser construido con inversores o con compuertas NAND o NOR. Si utilizamos inversores nuestro circuito puede ser como el siguiente:



Cualquier nivel establecido se mantendrá sin necesidad alguna de intervención externa. Para establecer un nivel solo basta con conectar temporalmente el terminal  $Q$  a un punto externo que tenga el nivel deseado y  $Q'$  se pondrá en el nivel opuesto.

La construcción de FF's con inversores resulta un poco impráctica debido a que no nos permite tener terminales de control y accesorios adicionales, de ahí que sea más común utilizar compuertas lógicas NAND y NOR.

### 1.2.1 FF CON COMPUERTAS NOR

La construcción básica de un FF con compuertas NOR se muestra en la siguiente figura:

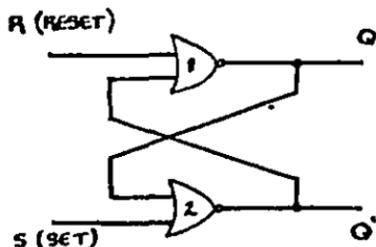
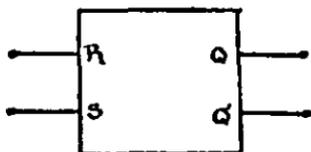


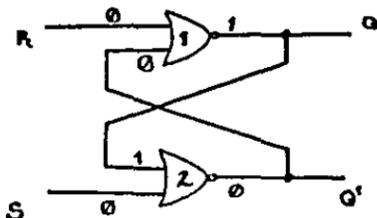
Diagrama lógico



Símbolo lógico

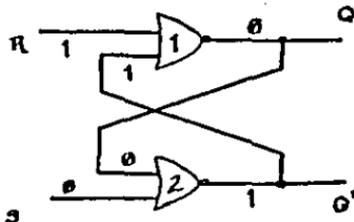
A partir de este circuito básico se pueden obtener FF's más complicados pero antes de verlos veremos como funciona este circuito básico.

Utilizando R y S como terminales de control vamos a ver como funciona con diferentes entradas. Si tenemos que  $R=S=0$ , cada una de las compuertas será activada o no dependiendo del valor de la otra entrada. Con una entrada  $R=S=0$ , se quedará el FF en el estado que estaba. Si tenemos por ejemplo que  $Q=1$  y  $Q'=0$ , en la compuerta 1 tenemos las entradas 0,0 lo cual dará como salida 1. En la compuerta 2 tendremos como entradas 0 y 1, por tanto, la salida será 0.



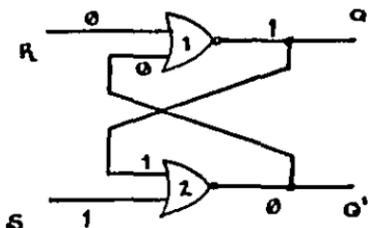
CON  $R=0$  Y  $S=0$  EL FF  
PERMANECE EN EL ESTADO  
ANTERIOR

Si suponemos ahora que  $R=1$  y  $S=0$ , entonces la compuerta 1 quedará inhabilitada y Q tendrá el valor de  $Q=0$  y por tanto  $Q'=1$ ; como se puede ver al poner  $R=1$  el FF se pondrá en RESET. Si el estado anterior era RESET, permanecerá igual, pero si era SET se pondrá en RESET.



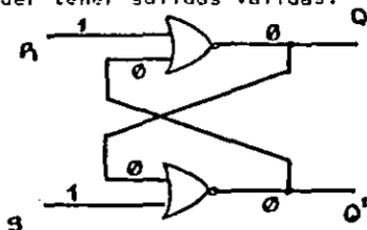
CON  $R=1$  Y  $S=0$  EL FF  
SE PONE EN RESET

Si suponemos ahora que  $S=1$  y  $R=0$ , la compuerta 2 queda inhabilitada y  $Q'$  tomará el valor  $Q'=0$  y por tanto  $Q=1$ ; como se puede observar con  $S=1$  pondrá en estado de SET al FF. Si el estado anterior era SET, permanecerá en este estado, pero si era -- RESET se pondrá en SET.



CON  $S=1$  Y  $R=0$  EL FF  
SE PONE EN SET

Por último consideremos la posibilidad de tener  $R=1$  y  $S=1$ . Esta condición dará como resultado que las compuertas 1 y 2 sean ambas 0, es decir,  $Q=0$  y  $Q'=0$ . Esta condición es ilegal pues deseamos tener  $Q$  opuesta a  $Q'$ . Esta condición no debe usarse para poder tener salidas válidas.



CON  $R=S=1$  EN LAS  
SALIDAS TENEMOS:  
 $Q=Q'=0$   
LO CUAL ES INVALIDO

A continuación tenemos una tabla que nos muestra los posibles estados del FF con compuertas NOR:

SET	RÉSET CLEAR	SALIDA
0	0	Permanece en estado anterior
0	1	$Q=0, Q'=1$
1	0	$Q=1, Q'=0$
1	1	Ambigua (NO VALIDA)

El FF que hemos analizado es conocido como RS debido a sus características de poner el FF en estado de SET o de RESET. Este FF también es conocido como SC (SET y CLEAR).

#### RESUMEN DEL FF CONSTRUIDO CON COMPUERTAS NOR

1.  $SET=0, RESET=0$  no afecta el estado del FF. El estado que tendrá es el estado previo.
2.  $SET=0, RESET=1$  siempre va a hacer que el FF sea aclarado o puesto a 0, esto se hará sin importar el estado previo. Pone en RESET el FF.
3.  $SET=1, RESET=0$  hace que  $Q=1$  sin importar el estado previo

del FF. A esto se le llama poner en SET el FF.

4. SET=1, RESET=1 nos da una salida inválida donde  $Q=Q'$  por tanto no debe usarse.

### 1.2.2 FF CON COMPUERTAS NAND

La construcción básica de un FF con compuertas NAND es muy similar al circuito del FF con compuertas NOR. El circuito es el siguiente:

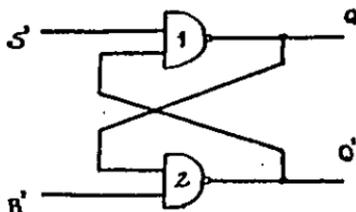
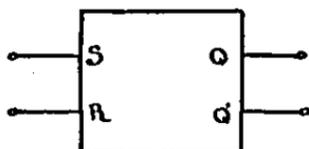


Diagrama lógico



Símbolo lógico

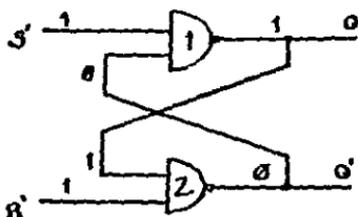
En este circuito la situación es opuesta, los terminales de control son activos en 0, de ahí que haya puesto las entradas como  $S'$  y  $R'$ .

La tabla de verdad para este circuito es:

SET'	RESET' CLEAR'	SALIDAS
0	0	Ambigua (NO VALIDA)
0	1	$Q=1, Q'=0$
1	0	$Q=0, Q'=1$
1	1	Permanece en el estado anterior

Como podemos ver, con las entradas  $S'=R'=1$  el FF permanece en el estado que tenía. Esto es debido a que con  $S'=R'=1$  se

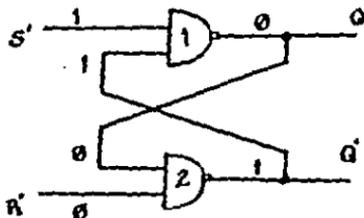
habilitan las compuertas o no dependiendo de la otra entrada. Si tenemos por ejemplo que  $Q=1$  y  $Q'=0$ , entonces en la compuerta 1 -- tenemos las entradas  $S'=1$  y  $Q'=0$ , lo cual nos da como salida  $Q=1$ .



CON  $R'=1$  Y  $S'=1$  EL FF  
PERMANECE EN EL ESTADO  
ANTERIOR

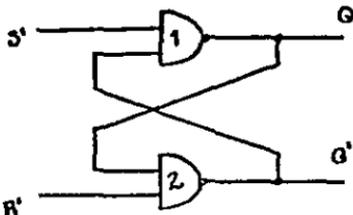
En la compuerta 2 las entradas son  $R'=1$  y  $Q=1$ , lo cual nos da como salida  $Q'=0$ .

Si suponemos ahora que  $S'=1$  y  $R'=0$ , entonces tenemos que la compuerta 2 queda habilitada y  $Q'$  tendrá el valor de  $Q'=1$  y  $Q$  será  $Q=0$ . Con esto estamos poniendo el FF en RESET.



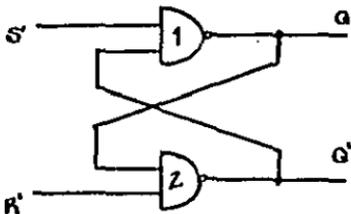
CON  $R'=0$  Y  $S'=1$  EL FF  
SE PONE EN RESET

Si suponemos ahora que  $S'=0$  y  $R'=1$ , entonces tenemos que la compuerta 1 queda habilitada de tal manera que en la salida  $Q$  tendremos  $Q=1$  y por tanto, en  $Q'$  tendremos  $Q'=0$ .



CON  $R'=1$  Y  $S'=0$  EL FF  
SE PONE EN SET

Con las entradas  $S'=R'=0$  obtendremos en cada salida un 1, es decir, en Q tendremos  $Q=1$  y en  $Q'$  tendremos  $Q'=1$ , una vez más encontramos que para tener salidas válidas Q y  $Q'$  deben ser opuestas, por tanto, no debemos usar estas entradas.



CON  $S'=R'=0$ ,  $Q=Q'=1$   
LO CUAL ES INVALIDO

Este FF descrito es también un RS o SC pero funciona con lógica negativa, de ahí que tenga inversores en las entradas en el símbolo lógico.

#### RESUMEN DEL FF CONSTRUIDO CON COMPUERTAS NAND

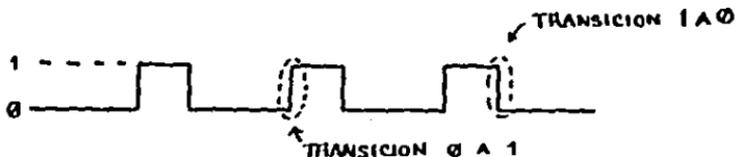
1. SET=0, RESET=0 nos da una salida inválida ya que produce que  $Q=Q'$ , por tanto, no debe usarse esta condición.
2. SET=0, RESET=1 hace que  $Q=1$ , por tanto, pone al FF en estado de SET.
3. SET=1, RESET=0 hace que  $Q=0$ , esto es, pone en RESET al FF.

4. SET=1, RESET=1 no afecta al FF y lo deja en el estado previo.

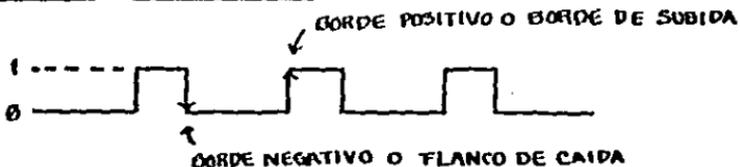
### 1.2.3 SEÑALES DE RELOJ

Muchos de los sistemas digitales operan como sistemas secuenciales síncronos. Esta sincronía en los sistemas se logra manejando señales de reloj. Estas señales de reloj son pulsos periódicos que van a diferentes partes del sistema y hacen que se produzcan ciertos estados a intervalos de tiempo espaciados regularmente.

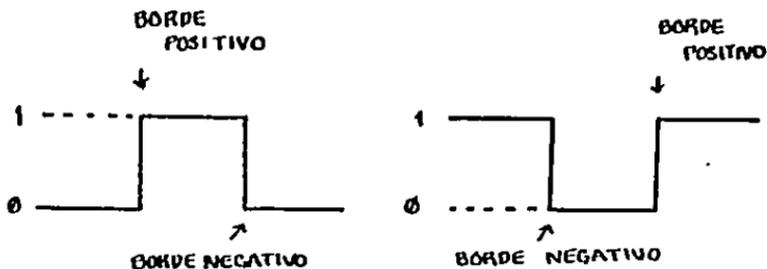
Las operaciones en los sistemas se hacen en el momento que se está haciendo una transición ya sea de 0 a 1 o de 1 a 0. La siguiente figura muestra una señal de reloj:



A las transiciones mostradas se les conoce con un nombre especial. A la transición de 0 a 1 se le llama flanco de subida o borde positivo; a la transición de 1 a 0 se le llama flanco de caída o borde negativo.

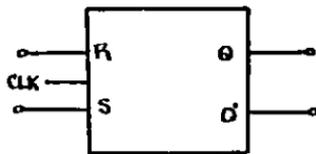
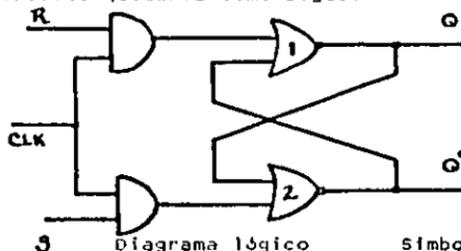


Para pulsos negativos aplicamos la misma idea, el borde negativo va de 1 a 0 y el borde positivo va de 0 a 1. En la siguiente figura podemos identificarlos:



#### 1.2.4 FF RS CON RELOJ O TEMPORIZADO

Los circuitos básicos de FF's que hemos visto son, por sí solos, circuitos secuenciales asíncronos. Para poder hacerlos síncronos podemos agregar 2 compuertas AND a las entradas del circuito básico; con esto podemos hacer que el FF responda a los niveles de entrada durante la ocurrencia de un pulso de reloj. El circuito quedaría como sigue:



3 Diagrama lógico

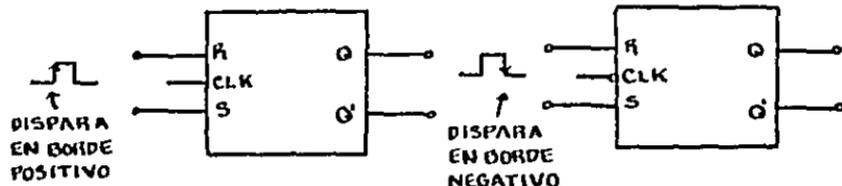
Símbolo lógico

La tabla de verdad puede ser ahora manejada tomando en cuenta el tiempo, en donde  $Q(t)$  está en un tiempo (presente) y la  $Q(t+1)$  está en un tiempo después de que ha ocurrido un pulso de reloj (estado siguiente).

Q(t)	S	R	Q(t+1)
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	Ambigua (NO VALIDA)
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	Ambigua (NO VALIDA)

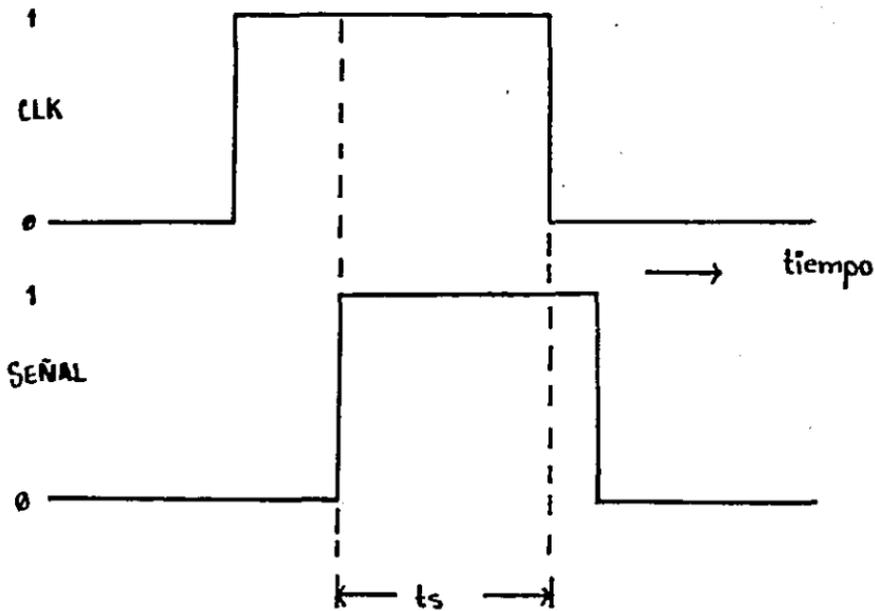
La forma en que las nuevas compuertas van a hacer la sincronía es controlando el tiempo en que serán permitidas las entradas S y R. Las entradas solo van a ser activadas cuando haya un pulso de reloj, esto es debido a que mientras no haya un pulso de reloj, la entrada CLK será 0 y por tanto las AND estarán desactivadas, una vez que haya un pulso, en CLK habrá un 1 y podrán activarse las compuertas AND permitiendo la entrada de S y R.

Hay dos formas de disparar estos FF's ya sea con borde positivo (un cambio de 0 a 1) o con borde negativo (con un cambio de 1 a 0). En ambos casos operan de la misma manera, la diferencia básica es la transición en que se dispara cada uno. El símbolo lógico para cada uno es:

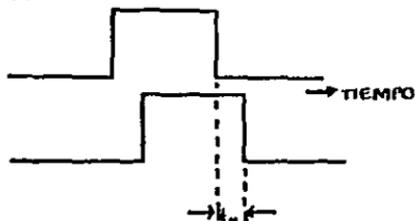


### 1.2.5 PARAMETROS DE ESTABLECIMIENTO Y MANTENIMIENTO

Los FF's disparados por borde tienen dos parámetros importantes: el tiempo de establecimiento y el tiempo de mantenimiento. El tiempo de establecimiento  $t_s$  es el tiempo que deben mantenerse estables las señales que preceden la ocurrencia del borde de disparo en la entrada CLK para que se pueda dar un disparo apropiado. La siguiente figura muestra este tiempo:



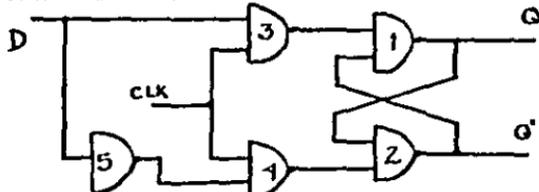
El tiempo de mantenimiento  $t_h$  es la cantidad de tiempo que debe mantenerse estable la señal después de que el borde de disparo del CLK ha ocurrido. Este tiempo es necesario para asegurar que haya un disparo apropiado. La siguiente figura muestra este tiempo:



Los tiempos de establecimiento y de mantenimiento están generalmente en la región de los nanosegundos.  $t_s$  está generalmente entre los 5 y 50 nanosegundos.  $t_h$  es por lo general menor a 10 nanosegundos.

#### 1.2.6 FF D CON RELOJ O TEMPORIZADO

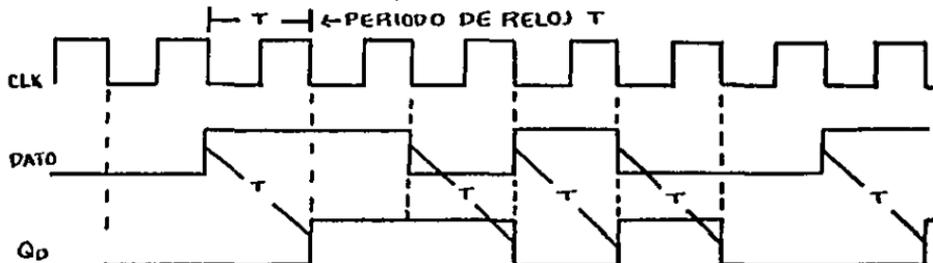
El FF D recibe este nombre debido a la habilidad que tiene para transmitir "datos". Su circuito lógico es una modificación del circuito del FF RE sincronizado. La siguiente figura muestra este circuito:



Como se ve en la figura la entrada D está conectada a través de un inversor a la compuerta 4. Mientras no haya un 1 en el CLK las compuertas 3 y 4 tendrán un 1 en la salida (no importa el valor de las otras entradas). En el momento en que tengamos un 1 en el CLK se tomará en cuenta el valor de la entrada D. Si  $D=1$ , la compuerta 3 será 0 y la compuerta 1 será 1, con esto

$Q=1$ . Como podemos ver  $D=Q$ . Si  $D=0$  y  $CLK=1$ , entonces la compuerta 3 será 1 y la compuerta 1 será 0, con esto  $Q=0$ . Como podemos ver lo que estamos haciendo es transmitiendo datos.

Es muy común que sea necesario retrasar una secuencia de bits, presentados sincronicamente, durante un ciclo de reloj. Con el FF D podemos lograr esto. En la siguiente figura vemos de una manera ilustrativa esta operación de retardo:



En el dibujo, el flanco de disparo de la señal de reloj coincide con el cambio de dato. Esto no es así en realidad ya que de ser así la respuesta del FF sería ambigua. En la práctica, los cambios en los datos se hacen muy poco después de la transición de disparo. El retardo entre la transición de disparo y el instante que cambia el dato sería del orden del retardo de propagación de un FF. Este sería el retraso que se encontraría si los datos se tomaran a la salida de otro FF, gobernado por la misma señal de reloj.

### 1.2.7 FF JK CON RELOJ O TEMPORIZADO

El FF JK es, en cierto sentido, una versión mejorada del RS ya que el circuito básico es similar, esta mejora es debida a que en el FF JK no hay combinaciones de entradas INVALIDAS o no permitidas. El circuito lógico del FF JK es el siguiente:

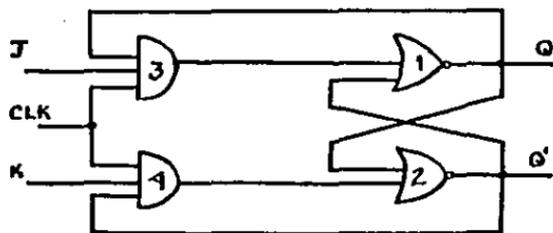
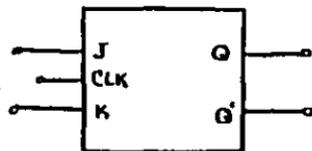


DIAGRAMA LOGICO



SIMBOLO LOGICO

La tabla de verdad para el FF JK es la siguiente:

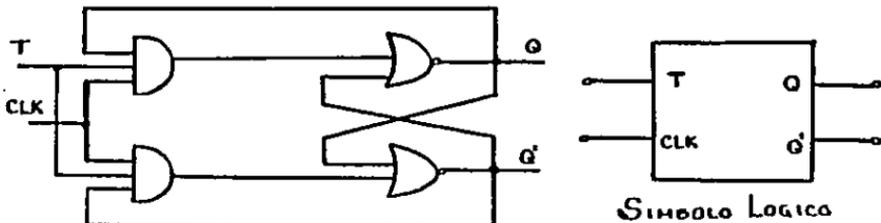
Q(t)	J	K	Q(t+1)
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

Como se puede ver, esta tabla es muy similar a la del FF RS, la diferencia que encontramos aquí es que no hay alguna combinación de entradas inválidas. Cuando  $J=K=1$  lo que vemos en la salida  $Q(t+1)$  es  $Q^t$ , esto es, con esta condición en el FF JK conmuta.

### 1.2.B FF T O DE CONMUTACION

El nombre de este FF se deriva de la habilidad que tie-

ne para conmutar ("TOGGLE"). Este FF es muy similar al JK, la diferencia está en que el FF T tiene conectadas las entradas J y K para formar una sola, la entrada T. El FF va a conmutar cada que reciba un pulso de reloj independientemente del estado interior, la única condición que hay para que lo haga es que  $T=1$ . Mientras  $T=1$  y haya pulsos el FF va a conmutar. El circuito lógico se muestra a continuación:



La tabla de verdad es la siguiente:

$Q_t$	T	$Q(t+1)$
0	0	0
0	1	1
1	0	1
1	1	0

### 1.2.8 FF MAESTRO-ESCLAVO

Los FF's M/E (maestro-esclavo) se construyen a partir de dos FF's RS o JK. Uno de estos dos FF's será maestro y el otro esclavo. La siguiente figura nos muestra un FF M/E construido con FF's RS:

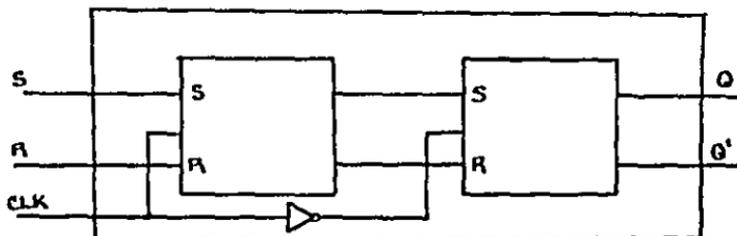
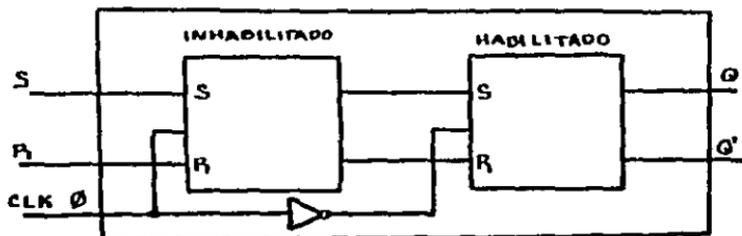


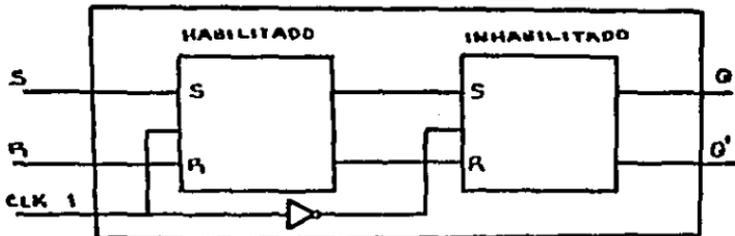
Diagrama lógico de un FF M/E

En la figura anterior tenemos la entrada del CLK del primer FF directa y en el segundo la tenemos invertida, esto hace que no sean activos al mismo tiempo. Si tenemos, por ejemplo, CLK=0, el primer FF está inhabilitado, mientras que el segundo estará habilitado y hará que la salida Q sea igual a la salida del maestro. En el FF maestro no habrá cambios a pesar de que R y S cambien.



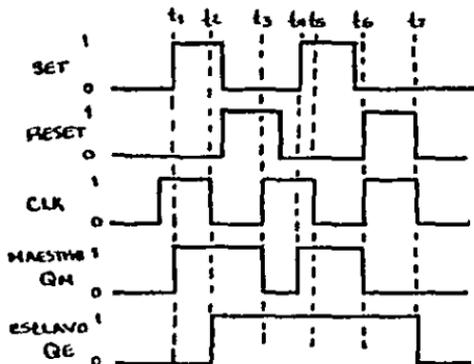
Cuando el CLK=1, el FF maestro es habilitado por lo que las entradas externas, R y S, serán transmitidas al maestro. Du-

rante este tiempo el FF esclavo quedará inhabilitado, con esto, no podrá cambiar de estado.



Para poder comprender mejor como cambia el FF M/E es necesario darse cuenta que el FF esclavo cambia solo cuando el CLK va de 1 a 0, esto es, en flanco negativo y se queda en ese estado mientras CLK esté en cero. Qué implica esto? Que las salidas del FF M/E cambian de estado en el flanco negativo. El FF maestro cambia en el momento que cambien sus entradas mientras el CLK sea 1. Esto nos muestra que el FF maestro puede cambiar más de una vez si cambia R y/o S mientras CLK=1.

Para darnos cuenta un poco mejor de lo que hemos hablado veamos ahora las siguientes señales:



Qué está pasando en cada uno de los tiempos  $t$  ? Antes de ver que sucede en cada tiempo, nótese que tenemos como estado inicial  $Q_m=Q_e=0$ ,  $R=S=0$  y  $CLK=0$ .

$t_1$ : En este tiempo el  $CLK$  está en 1, lo cual permite que  $Q_m$  cambie si cambian  $R$  y/o  $S$ ; como  $S$  cambia en este tiempo a 1, hace que  $Q_m$  cambie también a 1.

$t_2$ : En este tiempo ocurre un flanco negativo, y como habíamos dicho, este flanco habilita el esclavo. Como el esclavo está aquí en posibilidades de recibir a  $Q_m$ ,  $Q_e$  cambia a 1, que es el estado de  $Q_m$ .

$t_3$ : El  $CLK$  en este tiempo cambia a 1, y por tanto, habilita las entradas del FF maestro, como  $R$  ha tomado el valor de 1,  $Q_m=0$  y  $Q_e$  mantiene el estado que tenía antes pues no ha sido habilitado para poder cambiar.

$t_4$ : En este tiempo no ha cambiado nuevamente el  $CLK$ , aún está en 1, lo cual quiere decir que nuestro FF maestro continúa en posibilidades de cambiar si hay cambio en  $S$  y/o  $R$ . Como en este tiempo cambia  $S$ , el FF maestro recibe esta entrada y  $Q_m$  cambia a 1.

Durante este pulso de reloj  $Q_m$  ha cambiado dos veces.

Por qué es esto posible ? Porque para que el maestro cambie basta con tener un 1 en el  $CLK$  y que cambien  $S$  y/o  $R$ . Para el cambio del maestro no se requiere que haya un flanco.

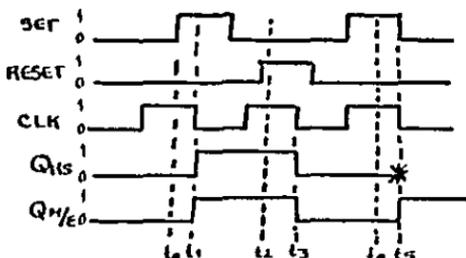
$t_5$ : Aquí el  $CLK$  está en flanco negativo, lo cual permite que el esclavo esté habilitado y pueda recibir las salidas del maestro. Como el maestro en este tiempo es 1 y  $Q_e$  estaba en 1, permanece en el mismo estado.

$t_6$ : El  $CLK$  cambia a 1, como hemos dicho, con esto el maestro se habilita y se dispone a aceptar las entradas  $S$  y  $R$ . En este tiempo tenemos que  $R=1$ , por tanto,  $Q_m=0$ . El maestro en este tiempo se encuentra deshabilitado por lo que se queda en el estado que estaba.

$t_7$ : Aquí ocurre un flanco negativo, por lo tanto, se habilita el esclavo y el maestro queda deshabilitado. El maestro se encuentra ahora en posibilidades de aceptar las salidas del maestro. Como  $Q_m=0$ , el FF esclavo cambia y  $Q_e=0$ .

Ahora vamos a analizar otras señales, vamos a comparar como responde un FF RS disparado en borde negativo y un FF M/E

hecho con FF's RS. Observa detenidamente la siguiente gráfica y trata de decir qué está pasando en cada tiempo:



t1: Como puede verse aquí hay un flanco negativo, por tanto, el FF RS está habilitado y acepta las entradas R y S; como S=1, el FF RS se pondrá en estado de SET, Q<sub>rs</sub>=1. En este momento también se habilita el esclavo y está en posibilidades de aceptar las salidas del maestro. En el tiempo t<sub>0</sub> CLK=1 y SET=1, por tanto, Q<sub>m</sub>=1. Esta Q<sub>m</sub>=1 es la que entra al esclavo y hace que Q<sub>m/e</sub>=1.

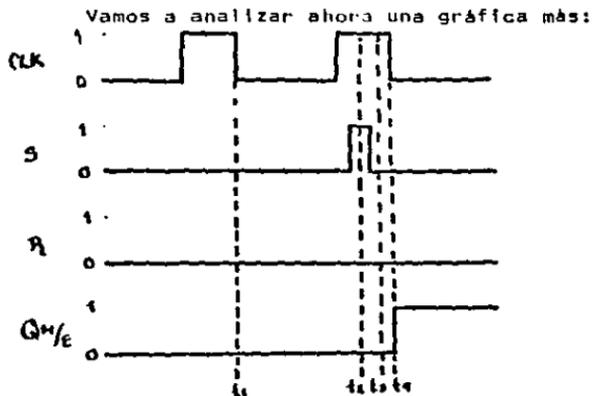
t2: En este tiempo R=1 y como el CLK=1, entonces el maestro puede cambiar y se pone en estado de RESET. El esclavo queda como estaba pues no ha sido habilitado.

t3: En este tiempo hay un flanco negativo de ahí que el esclavo acepte la salida Q<sub>m</sub>=0 del maestro y la salida Q<sub>m/e</sub> sea ahora Q<sub>m/e</sub>=0. Aquí también se activa el FF RS disparado por flanco negativo y Q<sub>rs</sub>=0.

t4: Aquí CLK=1, por tanto, se habilita el FF maestro y está dispuesto a aceptar las entradas R y S, como en este tiempo S=1, entonces Q<sub>m</sub>=1. No sucede nada en el FF RS disparado por flanco. Por qué ?

t5: Este tiempo es muy interesante y es el tiempo en el que deseo poner más atención. Aquí tenemos un flanco negativo en el CLK, por lo cual, se habilita el maestro aceptando la salida Q<sub>m</sub> y cambia a 1 su estado interno; entonces Q<sub>m/e</sub>=1. Ahora Qué es lo que pasa con el FF RS disparado por flanco ? Como vemos, hay un flanco negativo por lo cual se habilitan las entradas del FF, para reconocer el valor de R no hay problema, R=0. Pero, Qué valor tiene S ? Uno o cero ? En estas señales S está cambiando

en el mismo tiempo del flanco. Este detalle es el que hace que el FF RS disparado en flanco esté en cierta forma en desventaja del FF M/E. El FF M/E provee un disparo confiable siempre, no importa que las entradas estén cambiando en el momento en que ocurre un flanco.



t1: Aquí hay un flanco negativo, por tanto, se habilita el esclavo y permite activar Qm/e con el estado que hay en el maestro; como R=S=0, permanece el estado anterior y, como podemos ver es 0, por lo que en Qm/e continúa un 0.

t2: El CLK se encuentra en 1, por lo tanto, se habilita el maestro y acepta las entradas R y S. Como S=1, en Qm habrá un 1. El esclavo sigue sin cambio. Por qué?

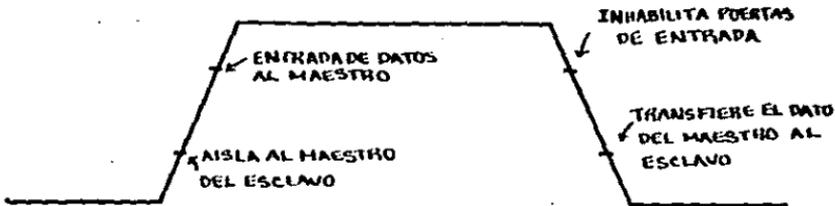
t3: En este tiempo el CLK sigue en 1 y siguen habilitadas las entradas del maestro. Como vemos S volvió a cero y ahora las entradas del FF M/E son R=0, S=0. Con estas entradas permanece el estado anterior y Qm sigue en 1.

t4: Aquí tenemos otra vez un flanco negativo y con esto, tenemos habilitadas las entradas del esclavo. Como Qm=1, el Qm/e cambia a 1.

Como podemos ver, la respuesta del FF está determinada por la última entrada que adopte el 1 lógico mientras las entra-

das estén habilitadas. Esta característica es conocida como la propiedad de captar unos de FF. Esta característica no es aceptada por algunos autores ya que el FF es muy susceptible a los riesgos y a las perturbaciones impradecibles del ruido.

Como resumen de lo que sucede en cada ciclo de reloj tenemos la siguiente figura:

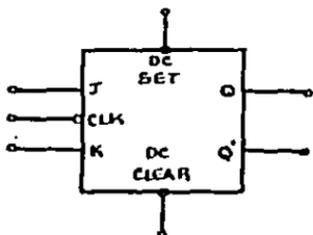


#### 1.2.10 ENTRADAS DIRECTAS DC SET Y DC CLEAR

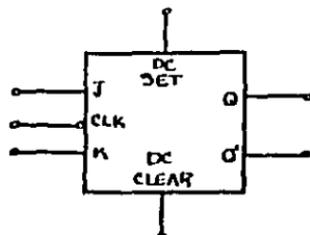
Algunos FF's vienen con unas entradas especiales para poner en 0 el FF o para ponerlo en 1 de manera asincrónica, esto es independientemente del reloj y de las entradas síncronas. Estas entradas son útiles para poner todos los FF's en su estado inicial antes de empezar a usarlos. Esto es importante ya que cuando aplicamos potencia a un circuito, el estado de los FF's es indeterminado.

Es importante que se verifique que al usar estas entradas directas no se mantengan constantes ya que al estar activa cualquiera de estas entradas el FF permanecerá en el estado que genera la entrada directa activa y no responderá a ninguna otra entrada.

Las entradas directas pueden ser activadas ya sea por pulsos altos o bajos. A continuación vemos dos FF's JK con entradas asíncronas, uno activado con 0's y otro con 1's.



ENTRADAS DIRECTAS ACTIVADAS  
EN PULSO NEGATIVO



ENTRADAS DIRECTAS ACTIVADAS  
EN PULSO POSITIVO

### 1.2.11 TABLAS DE EXCITACION DE LOS FF'S

Una tabla de verdad define la propiedad lógica del FF y caracteriza completamente su operación. Es muy útil para el análisis y la definición de la operación del FF. Esta tabla especifica el estado siguiente cuando las entradas y el estado presente se conocen. Las tablas de verdad para cada FF se muestra a continuación:

S	R	Q(t+1)
0	0	Q(t)
0	1	0
1	0	1
1	1	? INDETERMINADO

FF RS

D	Q(t+1)
0	0
1	1

FF D

J	K	Q(t+1)
0	0	Q(t)
0	1	0
1	0	1
1	1	Q(t)'

FF JK

T	Q(t+1)
0	Q(t)
1	Q(t)'

FF T

Durante el proceso de diseño se conoce por lo general la transición del estado presente al siguiente y se desea encontrar las condiciones de entrada del FF que causen la transición requerida. De esto nace la necesidad de tener una tabla que

liste las entradas necesarias para un cambio de estado dado. Estas listas se llaman tablas de excitación. A continuación presento las tablas de Excitación de cada FF de los que mostré su tabla de verdad:

Q(t)	Q(t+1)	S	R
0	0	0	X
0	1	1	0
1	0	0	1
1	1	X	0

FF RS

Q(t)	Q(t+1)	D
0	0	0
0	1	1
1	0	0
1	1	1

FF D

Q(t)	Q(t+1)	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

FF JK

Q(t)	Q(t+1)	T
0	0	0
0	1	1
1	0	1
1	1	0

FF T

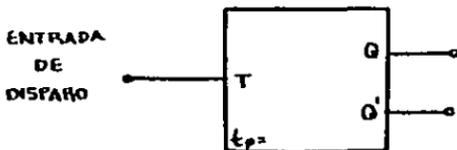
En cada tabla vemos dos columnas, Q(t) y Q(t+1), y una columna para cada entrada para poder mostrar como se logra la transición requerida. Tenemos cuatro transiciones posibles del estado presente al siguiente. El símbolo X representa una condición de no importa, es decir, no importa si la entrada es 1 ó 0, de cualquier manera el FF nos dará el estado esperado.

### 1.3 VIBRADOR MULTIPLE MONOESTABLE

Al flip-flop no solo se le conoce con este nombre sino con otros más, incluyendo multivibrador biestable, cerrojo y binario. Un circuito muy relacionado con el FF es el multivibrador monoestable. El monoestable, como el FF, tiene dos salidas, Q y Q', también una invertida con respecto a la otra. La diferencia que encontramos entre éstos es que el monoestable tiene solo un estado de salida estable (normalmente se tiene Q=0, Q'=1), donde permanece hasta que es disparado por una señal de entrada. Cuando

el monoestable es disparado, las salidas cambian al estado opuesto, con esto  $Q=1$  y  $Q'=0$ . El monoestable permanecerá en este estado un tiempo  $t_p$ , este tiempo está determinado generalmente por una constante de tiempo  $RC$  que forma parte del circuito del monoestable. Una vez que ha transcurrido el tiempo  $t_p$ , las salidas regresan a su estado normal ( $Q=0$ ,  $Q'=1$ ) hasta que se vuelva a disparar.

En la siguiente figura vemos el símbolo lógico del monoestable:



El valor de  $t_p$  se indica, generalmente, en alguna parte del símbolo del monoestable. Los valores más comunes de  $t_p$  están entre los nanosegundos.

Debido a sus características es muy útil como generador de pulsos, circuitos de control de tiempo, elementos de retardo, etc.

Como resumen podemos resaltar los siguientes puntos:

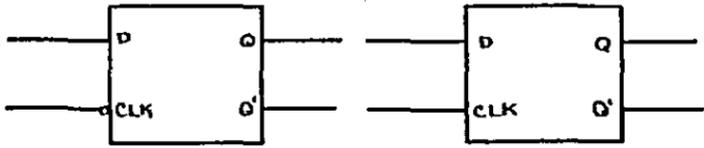
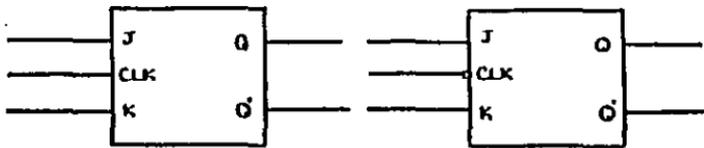
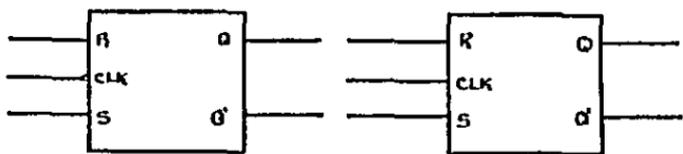
\* El estado normal del monoestable es  $Q=0$ ,  $Q'=1$ , éste cambiará cuando se dispare el monoestable y sus salidas serán  $Q=1$  y  $Q'=0$ . El monoestable durará en este estado un tiempo  $t_p$  y volverá al estado inicial ( $Q=0$ ,  $Q'=1$ ).

\* La duración del pulso con el que se dispara no importa ya que el monoestable se dispara con el flanco del CLK.

\* Una vez que el monoestable ha sido disparado, no tomará en cuenta ningún otro flanco, lo hará hasta que haya pasado el tiempo  $t_p$ .

# CAPITULO 2

1. Escribir con qué señal se dispara cada uno de los siguientes FF's:



2. Dibuja una señal de reloj en la que muestres un borde de subida o flanco positivo y un borde de bajada o flanco negativo.

1.

FLANCO NEGATIVO  
O  
BORDE DE BAJADA

FLANCO POSITIVO  
O  
BORDE DE SUBIDA

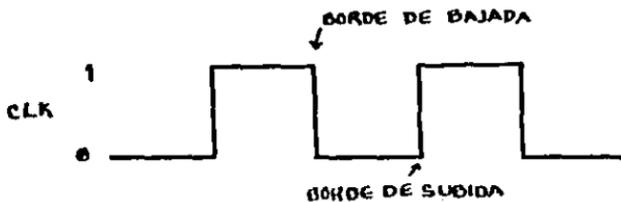
FLANCO POSITIVO  
O  
BORDE DE SUBIDA

FLANCO NEGATIVO  
O  
BORDE DE BAJADA

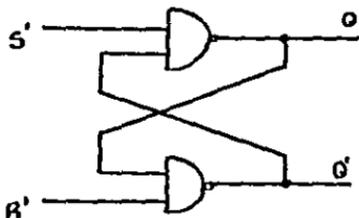
FLANCO NEGATIVO  
O  
BORDE DE BAJADA

FLANCO POSITIVO  
O  
BORDE DE SUBIDA

2.



3. Hacer una tabla de verdad en la que muestres cómo trabaja el FF construido con compuertas NAND que te presento a continuación:



---

4. Cuál es la diferencia entre un FF construido con compuertas NAND y uno construido con compuertas NOR ?

---

5. Cómo modificarías un FF RS para que funcione como un FF D ?

---

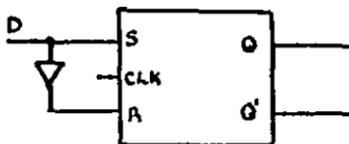
6. Qué arreglos harías a un FF JK para que se convierta en un FF de conmutación (T) ?

3.

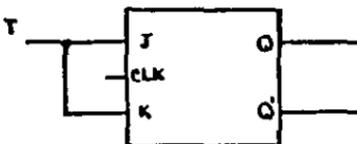
SET	RESET	Q
1	1	No cambia
1	0	Q=0
0	1	Q=1
0	0	Inválida

4. El FF construido con compuertas NOR responde a niveles altos en R y S, en cambio, el FF construido con compuertas NAND responde a niveles bajos en sus entradas.

5.



6.



7. En el siguiente pulso indica en donde se encuentra el borde negativo y el borde positivo:



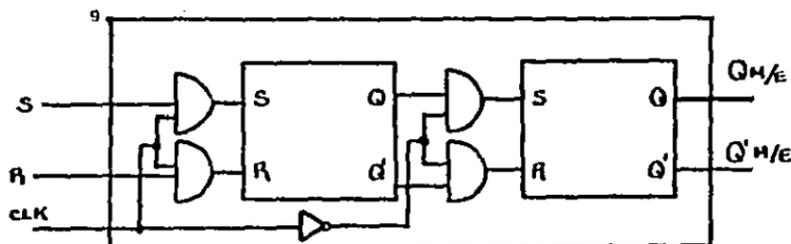
8. A qué se debe que en el tiempo en el que en un FF M/E está habilitado el FF maestro, el esclavo no está habilitado ?

9. Hacer un dibujo en el que muestres la construcción de un FF M/E a partir de FF's RS.

10. Explica como funciona un FF JK a través de una tabla de verdad.



8. A que el FF maestro está habilitado mientras el CLK = 1, y el FF esclavo solo se habilita en flancos negativos de reloj. Además las entradas CLK están invertidas una respecto a la otra.

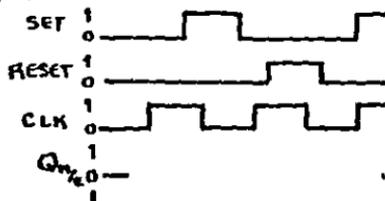


10.

J K Q

0	0	Permanezca en el estado anterior
0	1	Q=0
1	0	Q=1
1	1	Conmuta

11. Dibuja la señal de salida  $Q_{m/e}$  que se obtiene de acuerdo a las señales dadas a continuación si estamos utilizando un FF M/E.



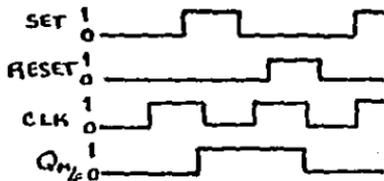
---

12. Explica las diferencias que hay en cuanto a funcionamiento, entre el FF RS y el JK.

---

13. Explica las diferencias que hay entre el FF T y el FF D.

11.



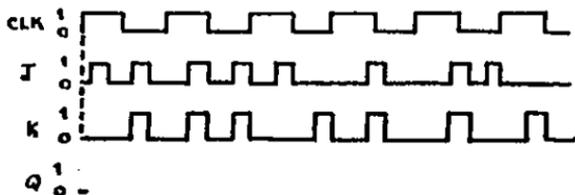
---

12. El FF JK responde adecuadamente bajo cualquier combinación de entradas y no encontramos un estado inválido, en cambio, en el RS tenemos que cuando  $R=S=1$  se produce una salida inválida, de ahí que no se deban usar éstas entradas.

---

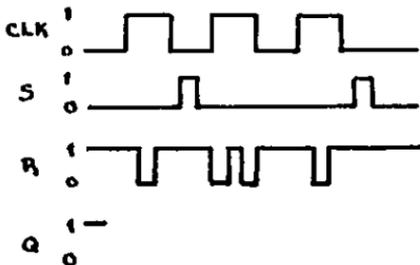
13. El FF T o de conmutación, como su nombre lo dice, tiene la función de conmutar cada que recibe un pulso, esto es, cada vez tendremos en la salida  $Q(t+1)=Q(t)'$ . El FF D solo se encarga de guardar información un tiempo y cuando recibe un pulso de reloj, permite la salida de la información sin ninguna modificación, la salida siempre es  $Q(t+1)=Q(t)$ .

14. Suponiendo que tienes un FF JK que se dispara en borde negativo y que las entradas J y K y el CLK son las siguientes:



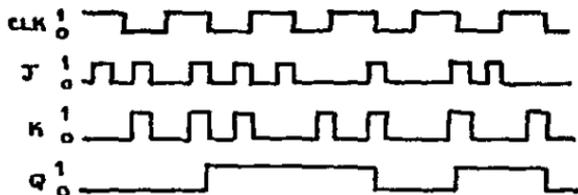
Tomando en cuenta que Q inicia en 0, dibuja la señal de salida Q que tendrías al aplicar las entradas mostradas.

15. Suponiendo que tienes un FF RS disparable en borde positivo y que las entradas R y S y la señal de reloj son las siguientes:

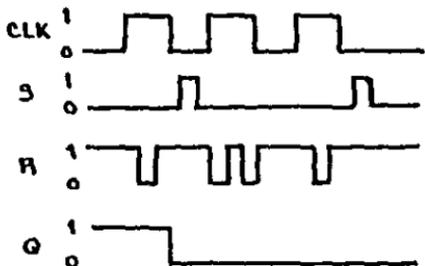


Tomando Q=1, haz un dibujo de la señal Q de salida del FF que tendrás al aplicar las señales anteriores.

14.



15.



16. Qué diferencia encuentras entre un FF disparado por borde y un FF M/E ?

---

17. Si tengo un FF M/E al cual entra una señal CLK=1 durante un tiempo t, Cuántas veces puede cambiar el estado del maestro ? y Cuántas veces puede cambiar el estado del esclavo?

16. Que el FF M/E provee un disparo confiable aún si las entradas de control están cambiando en el tiempo en que hay una transición de reloj. En los FF's disparados por borde no siempre hay este disparo confiable ya que cuando el CLK y alguna de las entradas van cambiando al mismo tiempo, el FF responderá impredeciblemente.

---

17. El estado del maestro puede cambiar tantas veces como cambios haya en las entradas mientras CLK=1.

El estado del esclavo solo cambiará en la transición del reloj, esto es en el flanco.

18. Describir los tiempos de establecimiento y de mantenimiento.

---

19. Para qué son útiles las entradas directas DC SET y DC CLEAR ?

---

20. Al aplicar potencia a un circuito, cuál es el estado de los FF's ?

---

21. Qué es lo que muestran las tablas de excitación ?

18. El tiempo de establecimiento es el tiempo que deben mantenerse estables las señales que preceden la ocurrencia del borde de disparo en la entrada del CLK para que pueda dar un disparo apropiado.

---

El tiempo de mantenimiento es la cantidad de tiempo que debe mantenerse estable la señal después de que el borde de disparo del CLK ha ocurrido.

---

19. Para poner los FF's en su estado inicial antes de empezar a usarlos

---

20. El estado de los FF's es indeterminado

---

21. Las condiciones de entrada del FF que causan la transición requerida o cambio de estado dado.

---

22. Hacer una tabla de excitación para un FF JK

---

23. Con qué otros nombres se le conoce el FF ?

---

24. Cuáles son las características de un vibrador multiple monoestable ?

---

25. Debido a sus características en qué es útil el mono estable ?

22.

Q(t)	Q(t+1)	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

---

23. Multivibrador biestable, carrojo, y binario

---

24. Al igual que el FF tiene dos salidas, Q y Q'. El monoestable tiene un solo estado de salida estable, normalmente se tiene Q=0 y Q'=1. Cuando es disparado, las salidas cambian al estado opuesto en donde permanecerá un tiempo determinado. Una vez transcurrido este tiempo las salidas regresarán a su estado normal. El monoestable es disparado con el flanco del CLK.

---

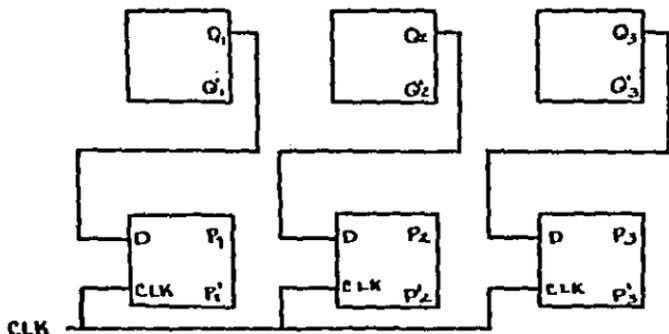
25. Como generador de pulsos, en circuitos de control de tiempo, en elementos de retardo, etc.

# CAPITULO 3

### 3.1 TRANSFERENCIA

La transferencia es una operación muy usual en sistemas digitales, podemos transmitir de un FF a otro o de un grupo de FF's a otro grupo de FF's. En sí la transferencia es llevar un dato o datos a otro lugar.

La mayor parte de las operaciones en una computadora o en algún procesador digital son transferencias de datos de un registro a otro. Los registros son arreglos de FF's como los que se muestran a continuación:



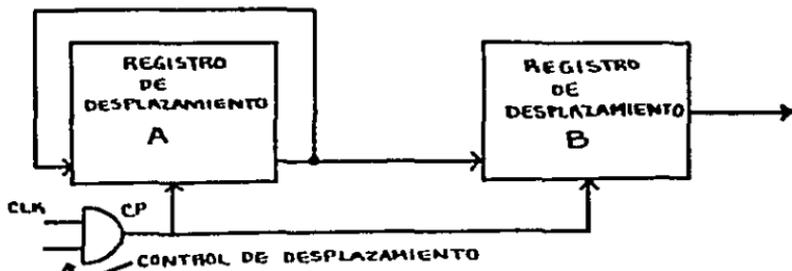
Podemos llamarle al primer registro Q, éste está compuesto de tres FF's por lo que podrá almacenar datos o palabras de tres bits. El segundo registro es el P, también compuesto de tres FF's. La transmisión se realiza mandando datos del registro Q al P cuando ocurre un flanco positivo. El dato que hay en Q1 pasará a P1, el dato en Q2 a P2 y el de Q3 a P3. La transmisión que se está efectuando aquí es en paralelo ya que se transmiten los tres bits al mismo tiempo.

Es muy importante que al estar utilizando los FF's para transferir datos, nos demos cuenta de la forma en que se disparan los FF's que estamos usando. Si por ejemplo, construimos nuestro registro con FF's D, debemos revisar que estos sean disparados en la transición de pulso y no en la duración del pulso. Por qué? Porque en el diseño de circuitos secuenciales temporizados se debe usar FF's disparados en flancos o bordes. A los registros que

responden a la duración de un pulso, se les conoce como compuertas retenedoras.

La transferencia de datos se puede efectuar en serie transfiriendo de bit a bit, esto es, manipulando un bit en cada tiempo. El contenido de un registro se transfiere a otro desplazando los bits de un registro al siguiente. La información se transfiere bit a bit, una cada vez desplazando los bits del registro fuente hacia el registro destino.

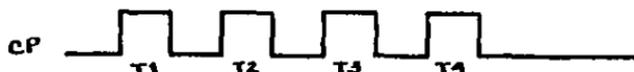
En la siguiente figura tenemos un diagrama a bloques de la transferencia serie del registro A al registro B:



Supongamos que en la figura tenemos cuatro bits en cada registro de desplazamiento. El tiempo de la señal de control de desplazamiento debe ser lo suficientemente grande para que se puedan transmitir los cuatro bits. Las señales que necesitamos para lograr la transferencia en CLK y control de desplazamiento son las siguientes:



Estas señales al pasar por la compuerta AND producen una señal como la que vemos a continuación:



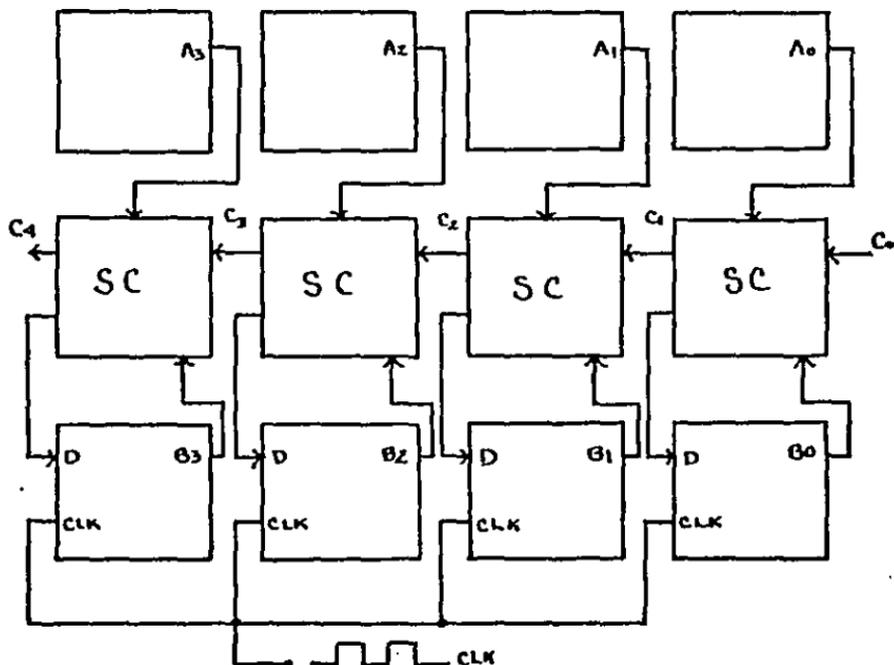
Con ésta señal se está haciendo que solo se transmita un bit cuando haya un pulso de reloj y además esté habilitada la señal de control de desplazamiento.

Como puedes ver, en el diagrama a bloques presentado tenemos una señal que sale del registro A y entra al registro B y al maestro y al mismo tiempo esta señal vuelve a entrar al registro A. Con qué fin tenemos esto? Con el fin de que con cada desplazamiento que se haga, el registro continúe teniendo 4 bits y que al final de la transmisión tenga los datos iniciales.

Qué diferencias tenemos entre los modos de operación en serie y paralelo? En el modo paralelo todos los bits son disponibles al mismo tiempo y pueden ser transferidos simultáneamente durante un pulso de reloj. En el modo en serie la información se transfiere bit a bit, con cada pulso de reloj se transfiere un bit mientras que los registros se van desplazando en la misma dirección. Con esto podemos darnos cuenta de la diferencia que tenemos en tiempo entre una forma de transmisión y otra. Las operaciones en serie son más lentas debido al tiempo que tarda en transferir la información hacia el interior y exterior de los registros de desplazamiento, y además requieren tantos pulsos de reloj como bits haya en el registro, en cambio, en la transmisión en paralelo se transmite todo en un pulso de reloj independientemente del número de bits a transmitir. Otra diferencia es la cantidad de material necesario para poder hacer la transmisión; en la transmisión en paralelo se requiere más material que en la transmisión serie.

### 3.2 SUMADORES

En una computadora, los números que van a ser sumados se almacenan en registros de FF's. En la siguiente figura vemos el esquema completo para la suma de dos números binarios de cuatro bits cada uno:



SUMADOR PARALELO

Los datos de cada registro van al SUMADOR COMPLETO (SC) y son sumados, una vez hecha la suma se va el resultado al registro B a través de cada salida Sn de cada SC. La salida Cn de cada SC es el acarreo. C4 es el acarreo de salida desde la última etapa y puede usarse como acarreo de entrada a una quinta etapa que nos indique que la suma excede 1111. Este sumador presentado es un sumador en paralelo ya que suma los 4 bits de cada registro con el mismo disparo de reloj, esto es, los suma al mismo tiempo.

Existen algunas ventajas y desventajas en cuanto al uso de sumadores en paralelo:

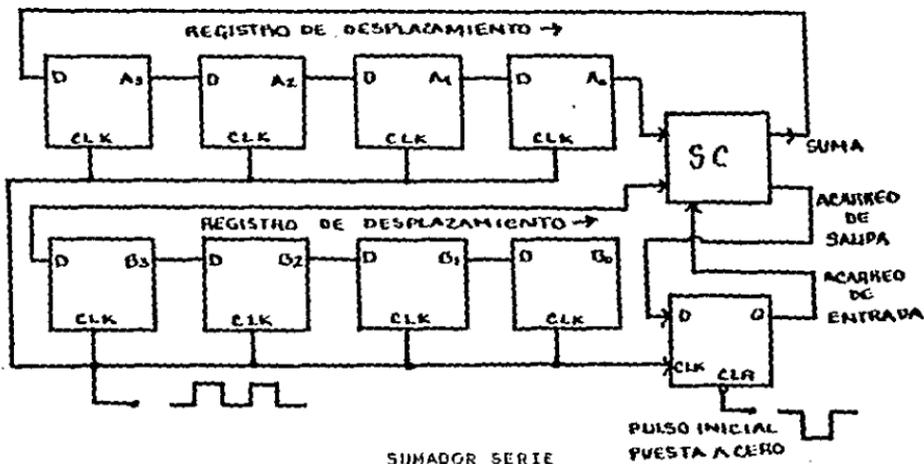
**VENTAJAS**

- La suma es rápida

**DESVENTAJAS**

- Usa una gran cantidad de circuitería lógica la cual aumenta en proporción directa al número de bits.

En la suma serie, la circuitería es más simple pero su velocidad de operación es menor. Lo que se usa en los sumadores en serie son registros de desplazamiento en los cuales sus valores binarios se desplazan de la izquierda a la derecha cada que se aplica un pulso de reloj. La siguiente figura nos muestra como podemos construir un sumador serie de 4 bits.



SUMADOR SERIE

El sumador completo tiene tres entradas y dos salidas, dos de las entradas, A0 y B0, son los bits a sumar y la otra en-

trada es el bit de acarreo de salida que va al FF de acarreo y la otra salida es la suma de los bits A0 y B0, esta salida va al FF A0 para que con esto la suma quede almacenada en el registro A.

Como podemos ver en el FF B0 tenemos una salida que se ramifica, una línea va al sumador completo y la otra va al FF B3, esto es con el fin de que en el registro B se mantenga el sumando.

Con cada pulso de reloj hay un desplazamiento de izquierda a derecha de un bit en cada registro, de tal manera que todos los bits llegarán a A0 y B0 para pasar al sumador completo y ser sumados.

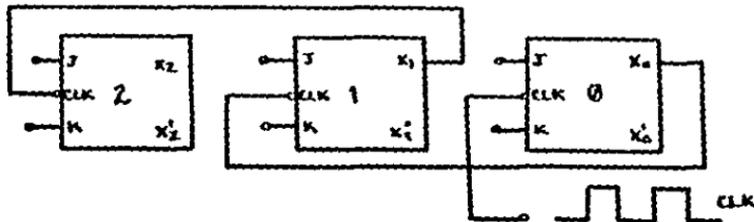
Si deseamos que nuestro sumador serie pueda manejar más bits por cada registro, lo que se debe hacer es agregar un par de FF's D por cada bit que queramos añadir.

Antes de iniciar la suma es necesario dar un pulso en la entrada CLK del FF de acarreo para que nuestro carry (acarreo) inicial sea cero.

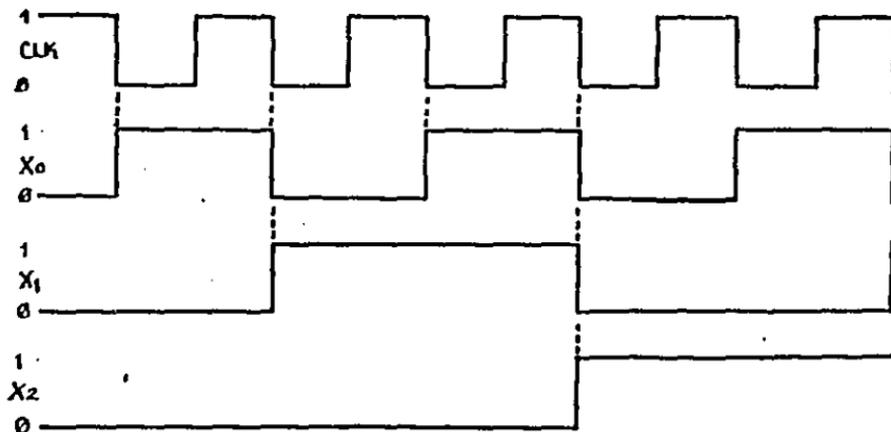
Podemos realizar cualquier otro circuito, ya sea para restar, dividir, multiplicar, etc. en donde utilicemos FF's como registros de almacenamiento en donde se tendrán guardados los operandos para poder realizar la operación. Los resultados, como vimos, también son almacenados en registros de almacenamiento.

### 3.3 DIVISOR DE FRECUENCIA

Para poder ver de qué manera podemos hacer un divisor de frecuencia con FF's vamos a analizar el siguiente circuito:



En cada entrada tenemos  $J=K=1$  y, por lo que hemos visto, con estas señales el FF va a conmutar cada que reciba un pulso de reloj. Como podemos ver el FF0 tiene la señal del CLK conectada directamente en su entrada CLK, mientras que el FF1 recibe en la señal CLK la señal que produce X0, y en el FF2 tenemos la señal X1 entrando en el CLK. Qué sucede con esto? Vamos a analizar las señales de salida de cada FF:



El FF0 conmuta en el flanco negativo de cada pulso de reloj, de tal manera que la frecuencia en X0 es la mitad de la que hay en el CLK.

El FF1 conmuta en el flanco negativo de los pulsos de la señal X0, de ahí que la frecuencia en X1 es la mitad de la frecuencia en X0 y un cuarto de la frecuencia del reloj.

El FF2 conmuta con los flancos negativos de la señal que tiene en X1, de ahí que la frecuencia en X2 sea la mitad de la frecuencia en X1 y 1/8 de la frecuencia de reloj.

Como puedes ver, en cada FF divide la frecuencia de su

entrada por dos. Usando n FF's tenemos una frecuencia de  $1/2^n$  de la frecuencia de entrada.

### 3.4 CONTADORES BINARIOS

La idea básica es similar a la forma en que se usaron los FF's como divisores de frecuencia. Si observamos los estados de cada FF después de los cambios de reloj, vemos como obtenemos un conteo binario. Vamos a revisar las señales que salen de los FF's :

X2	X1	X0	CLK	Secuencia de estados	Equivalente decimal
0	0	0	1	0 0 0	0
0	0	1	1	0 0 1	1
0	1	0	1	0 1 0	2
0	1	1	1	0 1 1	3
1	0	0	1	1 0 0	4
1	0	1	1	1 0 1	5
1	1	0	1	1 1 0	6
1	1	1	1	1 1 1	7
0	0	0	1	0 0 0	0
0	0	1	1	0 0 1	1
0	1	0	1	0 1 0	2
⋮	⋮	⋮	⋮		
⋮	⋮	⋮	⋮		

Los primeros 8 estados X2, X1, X0, nos muestran la secuencia binaria del conteo desde 000 hasta 111.

Para hacer contadores más grandes solo necesitamos agregar más FF's. Como se cuántos FF's tengo que usar para contar hasta el número que quiero contar? Esto es fácil pensando en que con n FF's tendré un número máximo de  $(2^n - 1)$ .

En el circuito divisor de frecuencia se usan tres FF's por lo tanto tenemos un número máximo de  $(2^3 - 1) = (8 - 1) = 7$  por tanto, nuestro número máximo (decimal) es 7; después de contar hasta el 7 (111 binario) regresa a cero (000).

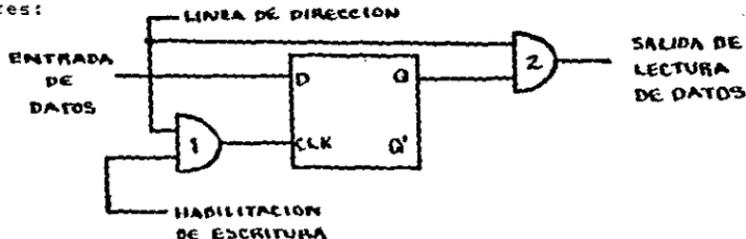
### 3.5 MEMORIAS

#### MEMORIAS DE LECTURA/ESCRITURA DE SELECCION LINEAL

Basicamente encontramos dos tipos de celdas de memoria usadas en memorias semiconductoras estáticas:

- a) SELECCION LINEAL
- b) SELECCION COINCIDENTE

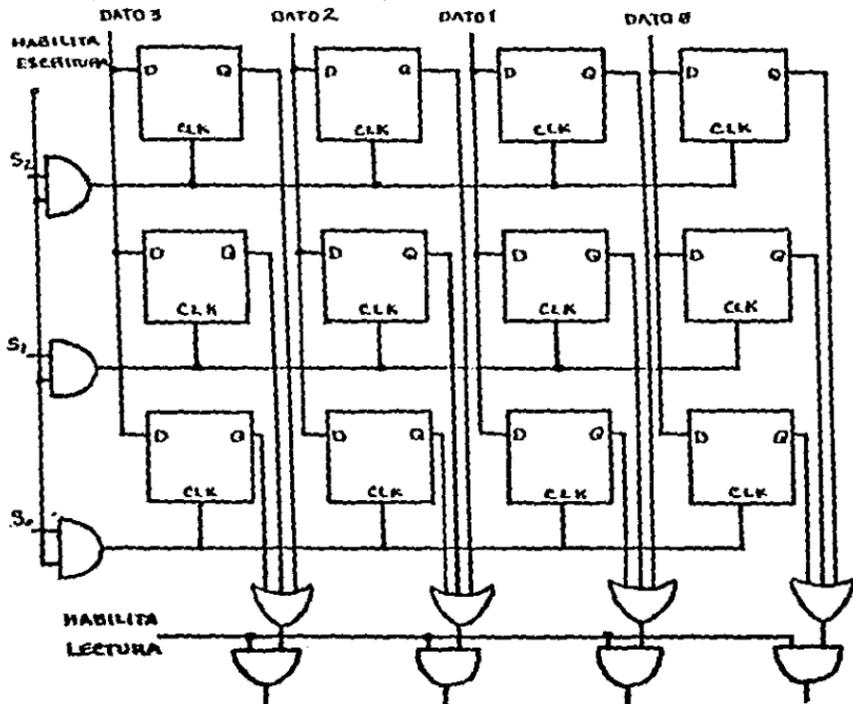
En este ejemplo veremos una aplicación de FF's en las memorias de selección lineal. Para este tipo de memoria usaremos un FF D. Los elementos de una celda de memoria son los siguientes:



La información se escribe en la celda poniendo la línea de DIRECCION en 1 lógico, aplicando los datos o información a la entrada DATOS y aplicando luego un 1 a la entrada de HABILITACION DE ESCRITURA. El estado de la celda se lee colocando la línea de DIRECCION en 1 y observando la salida de la compuerta 2.

Las celdas se pueden combinar en arreglos para almacenar un número dado de palabras de tantos bits por palabra como se desee.

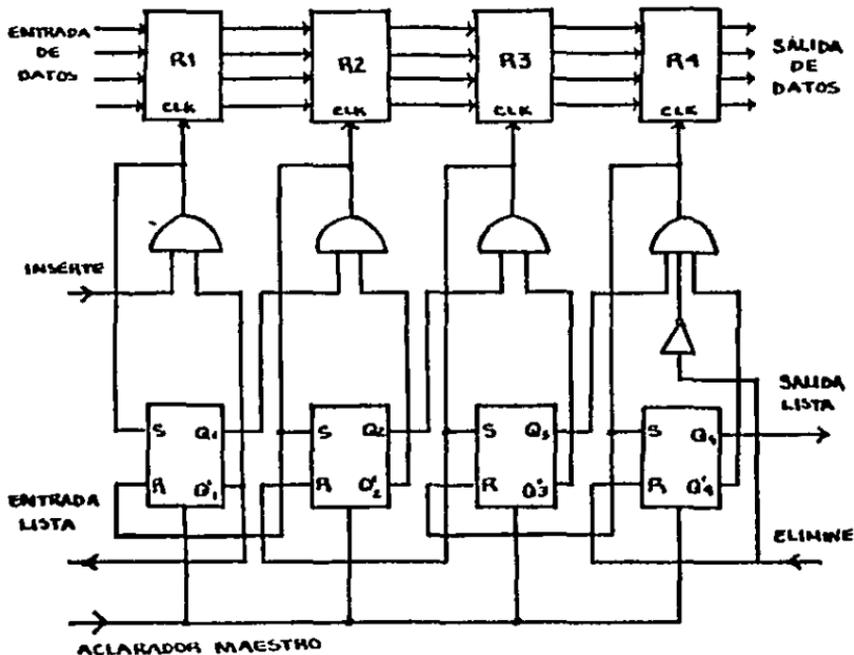
En la siguiente figura vemos un arreglo que almacena tres palabras de cuatro bits cada una:



### 3.6 SEPARADOR FIFO (FIRST INPUT FIRST OUTPUT)

Es muy común entre los computadores y sus demás dispositivos (por ejemplo entre un computador y una impresora) haya diferentes velocidades de transmisión de datos, de ahí que surja la necesidad de una unidad de almacenamiento de información que pueda recibir datos a una velocidad y transmitirlos en otra, transmitiéndolos además en el orden en que llegaron, esto es, el primero que llegue que sea el primero en salir. Un separador FIFO cumple con estos requisitos y es bastante útil. Los separadores FIFO son muy útiles en transferencia asincrónica.

Vamos a ver en un ejemplo sencillo como funcionan y como podemos construir uno. La siguiente figura nos muestra un separador FIFO de 4\*4 (almacenamiento de 4 palabras de 4 bits):



Como podemos ver tenemos 4 registros de 4 bits cada uno, con esto se podrá almacenar 4 palabras de 4 bits cada una. Cada uno de los registros tiene un flip-flop de control, éste controlará indicando si tiene o no una palabra almacenada en el registro que controla. Si el FF esta en 1 indica que hay una palabra almacenada en su correspondiente registro y un 0 indicará que no contiene datos válidos. Lo que en realidad está haciendo el FF de control es dirigiendo el movimiento de datos a través de los registros. Cada que el FFi es colocado en 1 y el FFi+1 es puesto en 0 se genera un pulso de reloj haciendo que el registro Ri+1 acepte los datos del registro Ri. Al efectuarse la transmisión de reloj se pone el FFi+1 en 1 y el FFi en 0. Los datos en los registros se mueven en el separador FIFO hacia la salida si las localidades que hay adelante, en el siguiente registro, estan vacías. Los datos se seguirán moviendo hasta que en un registro se encuentren que el siguiente FFi+1 esté en 1, o en el último registro R4.

Los datos son insertados en el separador FIFO unicamente cuando la señal Listo para entrar es habilitada. Esto sucede cuando el primer FF es restaurado indicando que el primer registro está vacío. Los datos se cargan en las líneas de entrada habilitando el reloj en R1 a través de la línea de control insertar. El reloj pone en 1 F1, el cual deshabilita el control entrada lista con esto indica que el FIFO esta ocupado y es incapaz de aceptar mas datos. Los datos son aceptados en el FIFO mientras R2 esté vacío. Los datos en R1 son transferidos a R2 y F1 es puesto en 0. Esto hace que se habilite la línea entrada lista, con esto tenemos disponible el FIFO para la entrada de otra palabra.

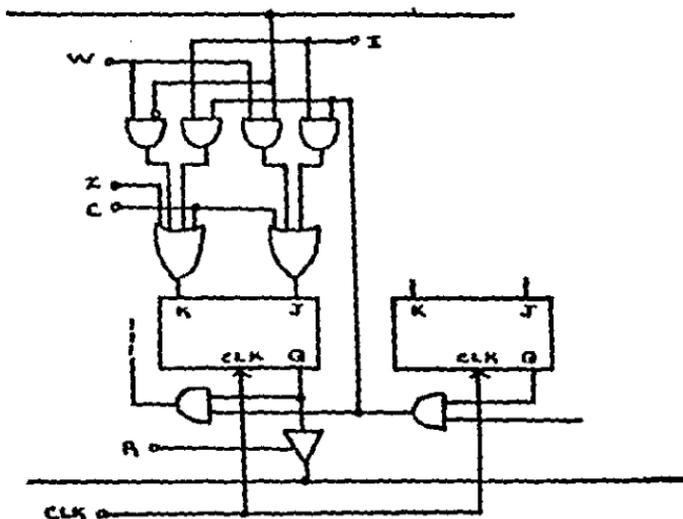
Mientras el FIFO está lleno F1 permanece en 1 y la línea entrada lista en 0. Los datos que entran en los registros se dejan en el terminal de salida. La línea de control salida lista se habilita cuando el último flip-flop F4 es puesto en 1, lo cual indica que hay datos válidos en R4. Una vez que está lleno el FIFO y que los datos son transmitidos a su destino, se habilita la señal de control suprimir, con esto el F4 se pone en 0 haciendo que salida lista se deshabilite. Solo después de que suprimir vuelve a 0, los datos de R3 se pueden transmitir a R4.

### 3.7 REGISTRO SENSIBLE A MULTIPLE ORDENES

Ordenes a las cuales responde el registro:

ORDEN	SIMBOLO
1. Escribe en el registro la palabra del bus	W
2. Lee en el bus la palabra del registro	R
3. Incrementa el registro	I
4. Complementa el registro	C
5. Borra el registro (toda Q=0)	Z

En la siguiente figura tenemos una de las etapas de un registro sensible a múltiples órdenes:



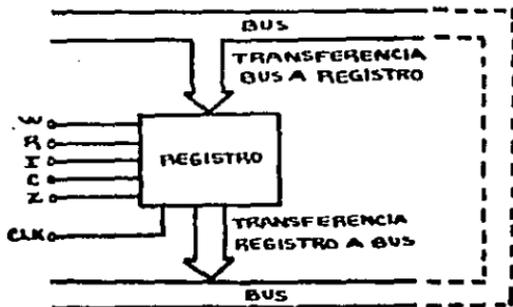
En la figura anterior tenemos 5 líneas de control: W, R, I, C y Z las cuales son activadas en un flanco de reloj. Solo una señal puede ser activada a la vez. Cuando, por ejemplo,  $W=1$ , al ocurrir un pulso de reloj se va a escribir en el registro la palabra que se tenga en el bus. Cuando  $R=1$ , se habilitan los buffers que hay en la salida de cada FF y permite que los datos que hay en el registro sean leídos.

Si queremos que se complemente la información que hay en el registro, ponemos  $C=1$ , con esto tenemos una entrada  $J=1$  y  $K=1$  con lo que cada FF conmuta complementándose así la información.

Si se desea borrar el registro, ponemos  $Z=1$ , con esto en K habrá un 1. El valor de J será 0 ya que en I hay un 0 y en W también, de ahí que en ambas compuertas AND salga un 0 y de la OR también. Como hemos visto, con entradas  $J=0$  y  $K=1$  el FF JK se pone en 0.

Para incrementar el registro ponemos  $I=1$ , y como se tienen conectados los FF's como contadores, se incrementa el registro al recibir el pulso de reloj.

Una representación a bloque del registro sensible a múltiples órdenes se muestra en la siguiente figura:



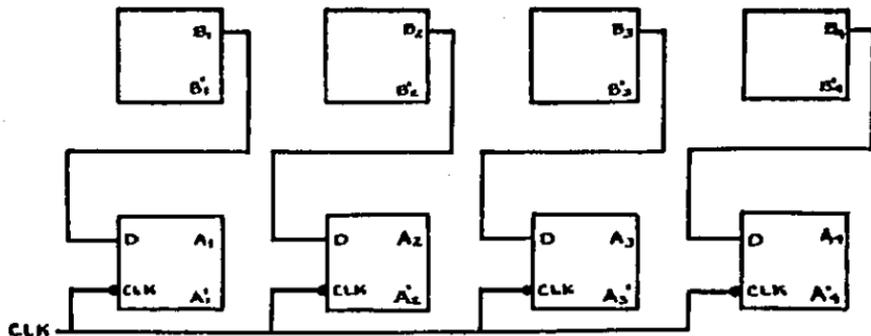
# CAPITULO 4

1. Hacer un circuito con dos registros A y B, en el cual se transfiera la información del B al A en paralelo. La información deberá ser transferida en el borde negativo. Tomar en cuenta que cada registro ha de almacenar 4 bits.

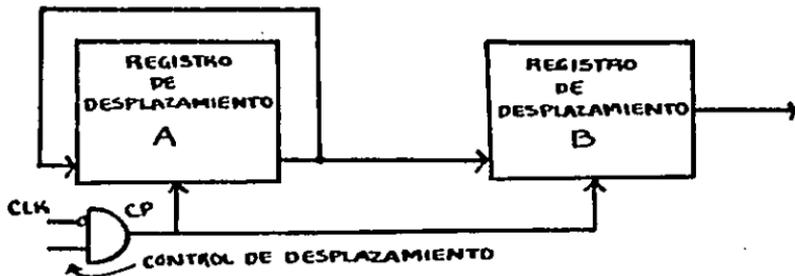
---

2. Hacer un circuito a bloques de 2 registros en los cuales se haga una transferencia serie de 5 bits del registro A al B en el borde negativo de reloj. Cuidar que el registro A no quede sin información.

1.



2.



3. Hacer un dibujo en el que muestres las señales tanto de reloj como de control de desplazamiento del circuito elaborado en el problema anterior

---

4. Menciona algunas diferencias entre la transferencia en paralelo y la transferencia serie.

---

5. Menciona los tipos de sumadores que hay y las diferencias que hay entre éstos.

3.



---

4. En la transferencia en serie tenemos que se transmite bit a bit, de tal manera que no se hace la transmisión al mismo tiempo como en la transmisión en paralelo. Otra diferencia que encontramos está en la cantidad de material que requiere cada una, en la transmisión en paralelo se requiere más material que en la transmisión serie.

---

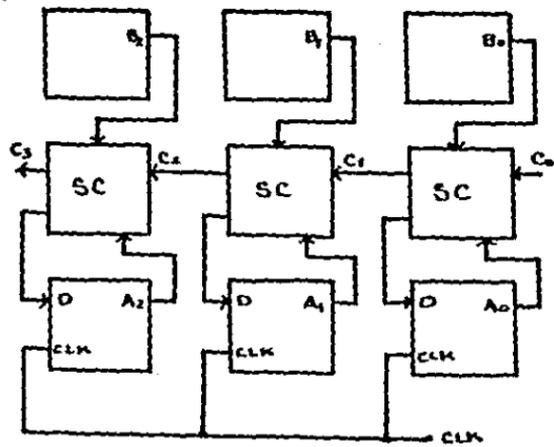
5. Básicamente tenemos el sumador paralelo y el sumador serie. En el sumador serie la suma se hará bit a bit a través de registros de desplazamiento. En el sumador en paralelo la suma de los bits se hace al mismo tiempo mandando éstos a un sumador completo. La suma paralela es más rápida pero requiere de más circuitería

6. Hacer un sumador paralelo de tres bits. Utilizar 2 registros A y B y guardar la suma en el registro A.

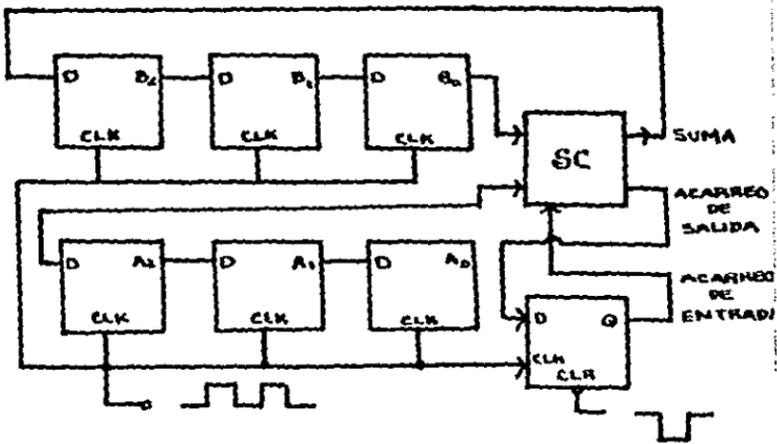
---

7. Hacer un sumador serie de tres bits. Utilizar dos registros A y B y guardar la suma en el registro B.

6.



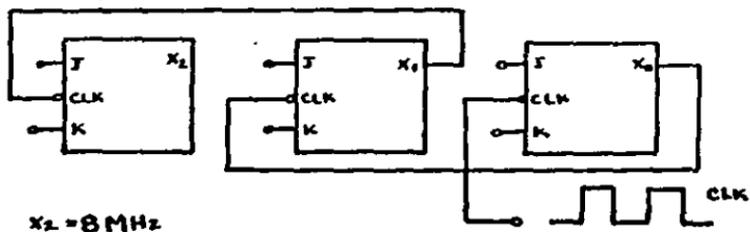
7.



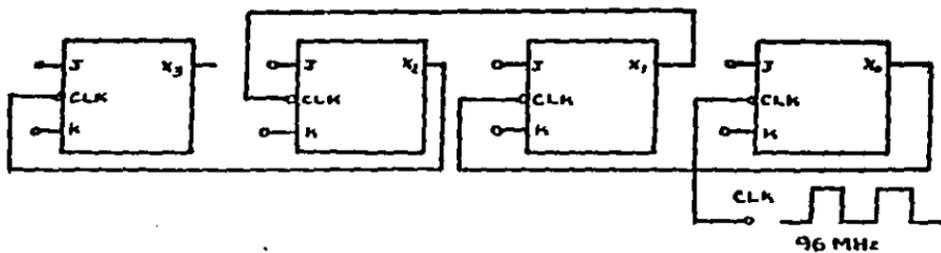
8. Construir un circuito divisor de frecuencia para obtener una frecuencia de 6 MHz si tenemos un reloj con una frecuencia de 96 MHz.

---

9. Cuál será la frecuencia del reloj en el siguiente circuito ?



8.



9.

64 MHz

10. Cuántos FF's necesito para hacer un contador binario que cuente de 00000 a 11111 ? \_\_\_\_\_. Hacer el circuito lógico de este contador.

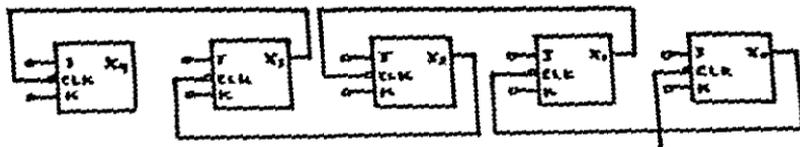
---

11. Cuál es la frecuencia de la señal que sale del último FF.

---

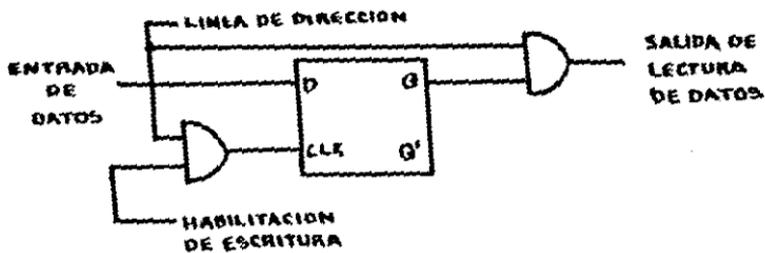
12. Hacer un diagrama de una celda de memoria especificando cada uno de sus elementos.

10.



11.  $\frac{1}{32}$  DE LA FRECUENCIA DEL RELOJ

12.



13. Construir un circuito que almacene una palabra de tres bits.

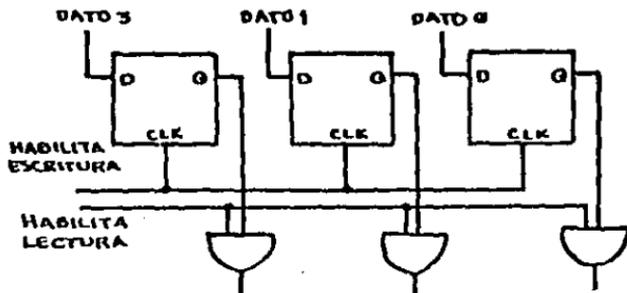
---

14. Qué es un separador FIFO ?

---

15. Cuándo es útil usar un separador FIFO ?

13.



---

14. Es una unidad de almacenamiento de información que puede recibir datos a una velocidad y transmitirlos a otra. El orden en que transmite los datos es: el primero que llega es el primero que sale (first input first output).

---

15. Es útil utilizarlo cuando tenemos dispositivos conectados entre sí y éstos operan a diferentes velocidades de transmisión.

## CONCLUSIONES

El texto programado es un material de apoyo útil tanto para el maestro como para el alumno.

En un texto programado el alumno participa activamente en el texto.

En la utilización de un texto programado como material de apoyo, el profesor puede tener la garantía virtual de que los alumnos que lo hayan seguido, tendrán la competencia necesaria en el tema.

Con el uso de un texto programado se logra tener un poco más de homogeneidad, en cuanto a aprendizaje, en los grupos ya que la velocidad de la instrucción es controlada por el alumno.

El texto programado es bastante práctico ya que el alumno puede disponer de éste en cualquier momento.

## A. NUMEROS BINARIOS

En el sistema binario encontramos solamente dos valores posibles para los dígitos, 0 y 1. Con este sistema podemos representar cualquier número decimal utilizando unos y ceros, en lugar de números del 0 al 9, como en el sistema decimal.

En el sistema binario de conteo, el valor de cada dígito se basa en el 2, en las potencias de dos de la siguiente manera:

$$\begin{array}{cccccccc} 2^8 & 2^7 & 2^6 & 2^5 & 2^4 & 2^3 & 2^2 & 2^1 & 2^0 \\ 256 & 128 & 64 & 32 & 16 & 8 & 4 & 2 & 1 \end{array}$$

Aquí, si el valor de un dígito es cero, su valor es de 0. Si el dígito es 1, su valor se determinará por su posición empezando a contar del lado derecho. Por ejemplo, el número binario 100110 equivale a:

$$\begin{array}{cccccc} 2^5 & 2^4 & 2^3 & 2^2 & 2^1 & 2^0 \\ 32 & 16 & 8 & 4 & 2 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 \\ 32 + 0 + 0 + 4 + 2 + 0 = 38 \end{array}$$

### \*Conversión de números decimales a binarios

La conversión se logra fácilmente siguiendo los siguientes pasos:

1. Dividir N por 2 (donde N es nuestro número decimal)
2. Separar el residuo, sea 0 ó 1, y dejar la parte entera como próximo número a dividir.
3. Continuar con los pasos 1 y 2 hasta que nuestra N sea 0.

4. Tomar los residuos para formar el número binario, el dígito menos significativo será el primer residuo y el último será el más significativo.

EJEMPLO 1: Encontrar la representación binaria de 17.

$$\begin{array}{r} N=17 \\ 8 \\ 4 \\ 2 \\ 1 \\ 0 \end{array} \begin{array}{l} 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{array} \rightarrow 10001 = 17$$

EJEMPLO 2: Encontrar la representación binaria de 28.

$$\begin{array}{r} N=28 \\ 16 \\ 8 \\ 4 \\ 2 \\ 1 \\ 0 \end{array} \begin{array}{l} 0 \\ 0 \\ 1 \\ 1 \\ 1 \\ 1 \\ 0 \end{array} \rightarrow 11100 = 28$$

## B. TEOREMAS BASICOS DEL ALGEBRA DE BOOLE

\* Teoremas que involucran una sola variable

1)  $x \cdot 0 = 0$

2)  $x \cdot 1 = x$

3)  $x \cdot x = x$

4)  $x \cdot x' = 0$

5)  $x + 0 = x$

6)  $x + 1 = 1$

7)  $x + x = x$

8)  $x + x' = 1$

\* Teoremas que involucran más de una variable

1)  $x + y = y + x$

LEYES CONMUTATIVAS

2)  $x \cdot y = y \cdot x$

3)  $x + (y + z) = (x + y) + z = x + y + z$

LEYES ASOCIATIVAS

4)  $x(yz) = (xy)z = xyz$

5)  $x(y + z) = xy + xz$  LEY DISTRIBUTIVA

6)  $x + xy = x$

7)  $x + x'y = x + y$

8)  $(x + y)' = x'y'$

LEYES DE MORGAN

9)  $x'y' = x' + y'$

### C. COMPUERTAS LOGICAS DIGITALES

NOMBRE COMPUERTA	SIMBOLO GRAFICO	FUNCION ALGEBRAICA	TABLA DE VERDAD															
AND		$F = xy$	<table border="1"> <thead> <tr> <th>x</th> <th>y</th> <th>F</th> </tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </tbody> </table>	x	y	F	0	0	0	0	1	0	1	0	0	1	1	1
x	y	F																
0	0	0																
0	1	0																
1	0	0																
1	1	1																
OR		$F = x + y$	<table border="1"> <thead> <tr> <th>x</th> <th>y</th> <th>F</th> </tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </tbody> </table>	x	y	F	0	0	0	0	1	1	1	0	1	1	1	1
x	y	F																
0	0	0																
0	1	1																
1	0	1																
1	1	1																
INVERSOR		$F = x'$	<table border="1"> <thead> <tr> <th>x</th> <th>F</th> </tr> </thead> <tbody> <tr><td>0</td><td>1</td></tr> <tr><td>1</td><td>0</td></tr> </tbody> </table>	x	F	0	1	1	0									
x	F																	
0	1																	
1	0																	
SEPARADOR		$F = x$	<table border="1"> <thead> <tr> <th>x</th> <th>F</th> </tr> </thead> <tbody> <tr><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td></tr> </tbody> </table>	x	F	0	0	1	1									
x	F																	
0	0																	
1	1																	
NAND		$F = (xy)'$	<table border="1"> <thead> <tr> <th>x</th> <th>y</th> <th>F</th> </tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>0</td></tr> </tbody> </table>	x	y	F	0	0	1	0	1	1	1	0	1	1	1	0
x	y	F																
0	0	1																
0	1	1																
1	0	1																
1	1	0																
NOR		$F = (x + y)'$	<table border="1"> <thead> <tr> <th>x</th> <th>y</th> <th>F</th> </tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>0</td></tr> </tbody> </table>	x	y	F	0	0	1	0	1	0	1	0	0	1	1	0
x	y	F																
0	0	1																
0	1	0																
1	0	0																
1	1	0																

HOMBRE  
COMPUERTA

SIMBOLO  
GRAFICO

FUNCION  
ALGEBRAICA

TABLA DE  
VERDAD

XOR



$$F = xy' + x'y$$
$$F = x \oplus y$$

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

XNOR



$$F = xy + x'y'$$
$$F = x \odot y$$

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

#### D. FUNCIONES BOOLEANAS

Una función de Boole es una expresión formada con variables binarias, operadores como AND, NOT, OR, paréntesis y el signo igual. Una vez que se ha formado la función booleana y se han elegido valores para cada variable, el valor de la función puede ser 1 ó 0.

Supongamos que tenemos la siguiente función:

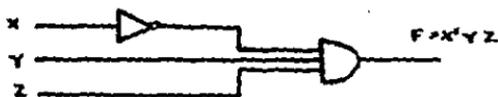
$$F = X'YZ$$

esta función será 1 si  $x=0$ ,  $y=1$ , y  $z=1$ ; con cualquier otro conjunto de valores la función será igual a 0.

Las funciones booleanas pueden representarse por medio de tablas de verdad. Para representarlás las necesitamos tener una lista de  $2^n$  combinaciones de unos y ceros y una columna que muestre el valor de la función para cada combinación en la tabla. Tomando la función que se usó anteriormente se hará una representación en una tabla;

X	Y	Z	$F = X'YZ$
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

Una función de Boole puede transformarse de una expresión algebraica a un diagrama lógico compuesto de compuertas AND, OR y NOT. Si queremos hacer el diagrama lógico de la expresión algebraica anterior tenemos:



Para hacer el diagrama lógico se pone una compuerta AND por cada término que haya y una compuerta OR para combinar dos o más términos. Es recomendable que se simplifique la función antes de que se haga el diagrama, para esto, es necesario que se conozca cómo aplicar los teoremas del álgebra de Boole a las funciones.

Ejemplo: Supongamos que tenemos  $F3 = X'Y'Z + X'YZ + XY'$  vamos a simplificar la función:

$$F3 = X'Y'Z + X'YZ + XY'$$

$$= X'Z(Y' + Y) + XY'$$

$$\text{como } Y' + Y = 1$$

$$F3 = X'Z(1) + XY'$$

$$= X'Z + XY'$$

$$\underline{F3 = X'Z + XY'}$$

## E. CIRCUITOS COMBINACIONALES

En los sistemas digitales encontramos básicamente dos tipos de circuitos: los secuenciales y los combinacionales. Los circuitos combinacionales están compuestos de compuertas lógicas cuyas salidas están determinadas en cualquier momento por la combinación que hay en las entradas en ese momento sin tomar en cuenta las entradas que se haya tenido anteriormente. Los circuitos combinacionales realizan sus operaciones por medio de compuertas lógicas, no requieren elementos de memoria. Los circuitos secuenciales utilizan elementos o celdas de memoria, además de compuertas lógicas. Sus salidas dependen de las entradas en ese momento y del estado que tienen los elementos de memoria los cuales a su vez dependen de las entradas previas.

Un circuito combinacional se forma de variables de entrada, compuertas y variables de salida.



Para  $n$  variables de entrada hay  $2^n$  combinaciones de valores posibles de entrada binaria. Para cada combinación de entrada se obtiene una y solo una combinación posible en la salida.

Es muy común ver que las fuentes y destinos de variables que usan los circuitos combinacionales sean registros de acumuladores, estos registros no deben tener influencia en el comportamiento del circuito combinacional, de tener alguna influencia se convertiría en un circuito secuencial.

## BIBLIOGRAFIA

Morris M.M. Lógica Digital y Diseño de Computadores. México, Distrito Federal: Prentice-Hall Hispanoamericana, 1979.

Taub H. Circuitos Digitales y Microprocesadores. México, Distrito Federal: Mc Graw Hill, 1982.

Morris M.M. Arquitectura de Computadores México, Distrito Federal: Prentice-Hall Hispanoamericana, 1982.

Tocci R.J. Sistemas Digitales: Principios y Aplicaciones. México, Distrito Federal: Prentice-Hall Hispanoamericana, 1986.