

**Tema 3**  
**Circuitos Secuenciales**  
**Análisis FF**

FCHE 2011-2

Parte 1

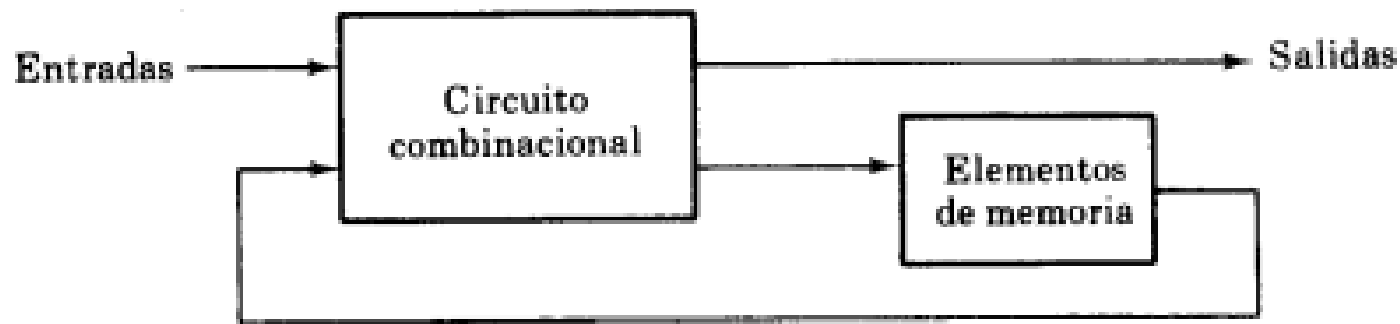
# Circuitos Secuenciales

## Contenido:

- 3.1 Estructura y modelo de la máquina secuencial Mealy y Moore. Estado presente y estado siguiente, decodificación del estado siguiente, decodificación de salidas, elementos de memoria
- 3.2 Elementos de memoria Latches y Flip – Flops tipos RS, D, T, JK. Flip – Flop maestro – esclavo. Tablas características, tablas de excitación, ecuaciones características, diagramas de tiempo. Descripción estructural y por comportamiento usando lenguajes HDL.
- 3.3 Sistemas Secuenciales Síncronos vs. Sistemas Secuenciales Asíncronos
- 3.4 Descripción estructural y por comportamiento de los bloques secuenciales básicos: Registros de “n” bits entrada serial – salida serial, entrada serial – salida paralela, entrada paralela – salida serial, entrada paralela – salida paralela, registro universal. Contadores de “n” bits ascendente, descendente, anillo, módulo “m”, Johnson en versión síncrona y asíncrona

# Introducción

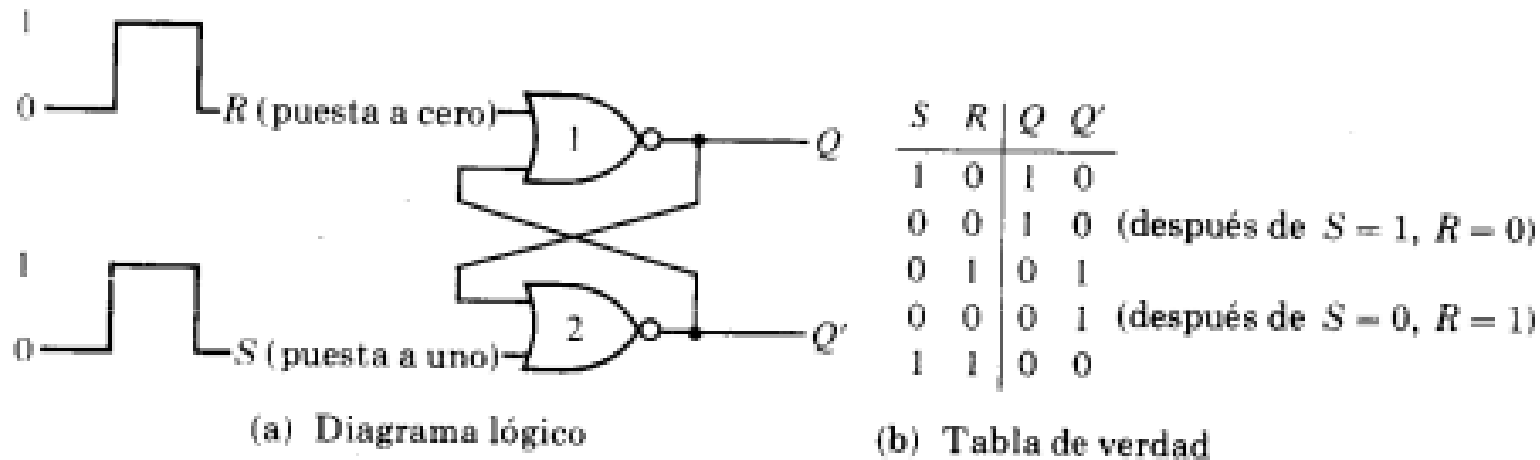
Los circuitos digitales hasta ahora considerados han sido combinacionales, es decir, las salidas en un instante dado de tiempo son enteramente dependientes de las entradas presentes en ese mismo tiempo. Aunque cada sistema digital debe tener circuitos combinacionales, la mayoría de los sistemas encontrados en la práctica incluyen también elementos de memoria, los cuales requieren que el sistema se describa en términos de la *lógica secuencial*.



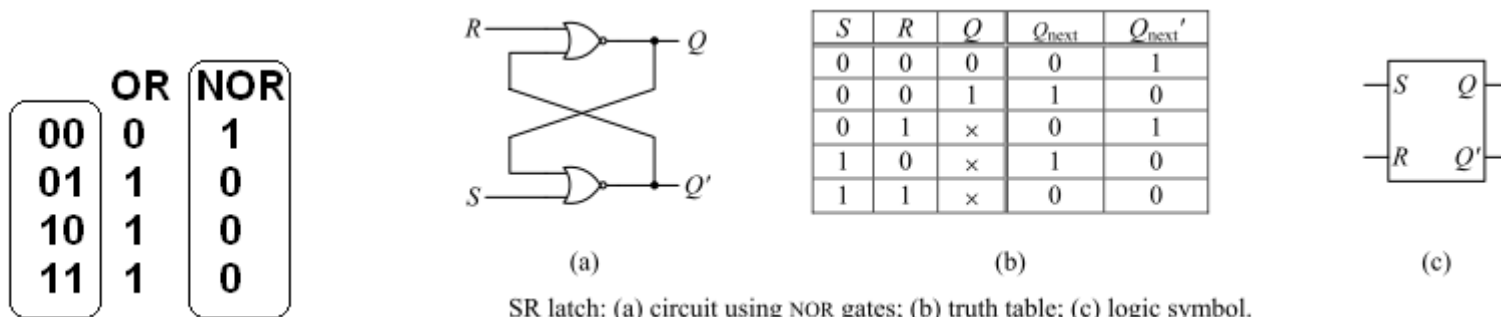
un circuito secuencial asincrónico puede tomarse como un circuito combinacional con realimentación.

# FLIP-FLOPS

Un circuito flip-flop puede mantener un estado binario indefinidamente (siempre y cuando se esté suministrando potencia al circuito) hasta que se cambie por una señal de entrada para cambiar estados. La principal diferencia entre varios tipos de flip-flops es el número de entradas que poseen y la manera en la cual las entradas afectan el estado binario.

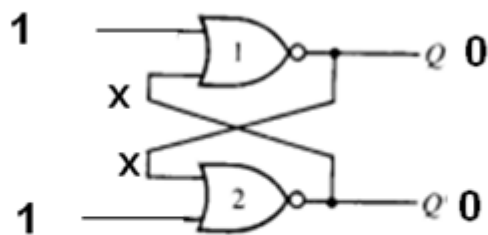


## Circuito flip-flop básico con compuertas NOR



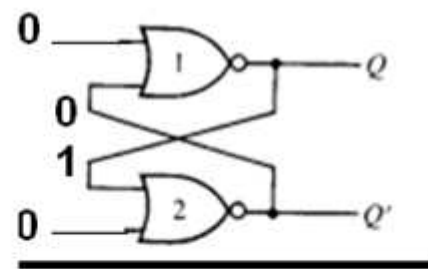
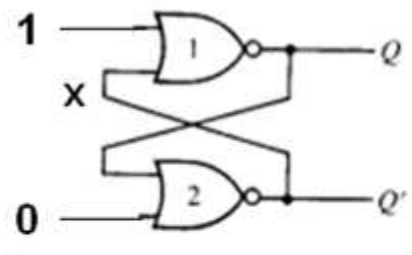
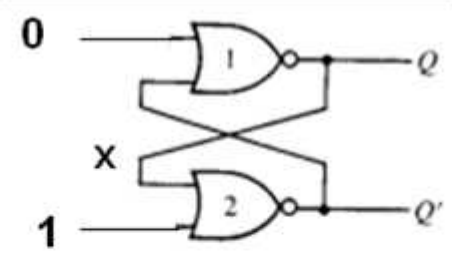
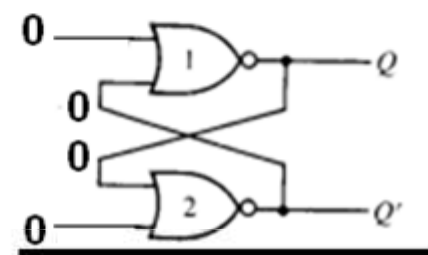
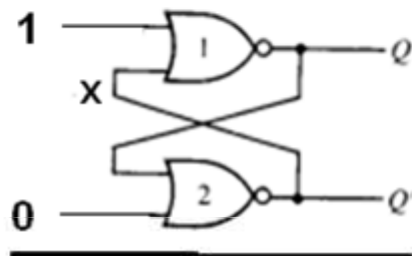
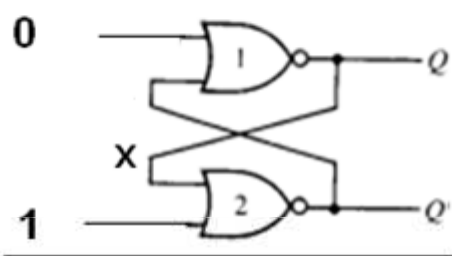
SR latch: (a) circuit using NOR gates; (b) truth table; (c) logic symbol.

# Análisis de FF SR. Preparación

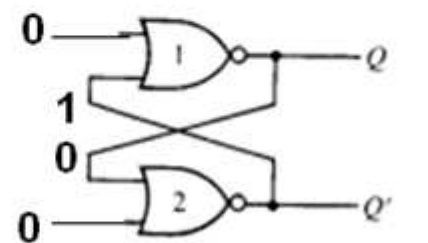
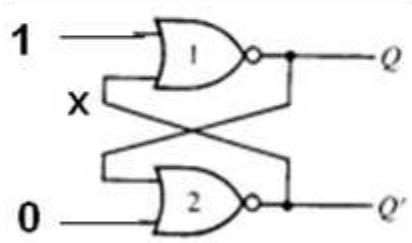
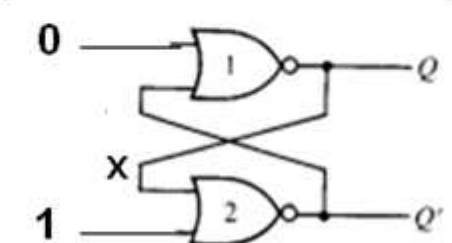


RS	Q	$\bar{Q}$
00		
01		
10		
11	0	0

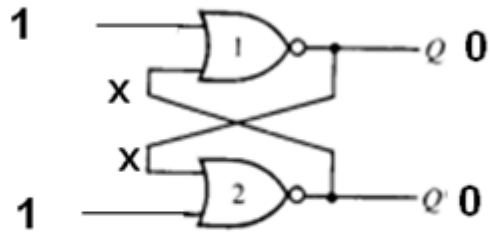
	OR	NOR
00	0	1
01	1	0
10	1	0
11	1	0



NOR :  
cuando  
tenemos un  
1 en  
cualquiera  
de las  
entradas, su  
salida es 0

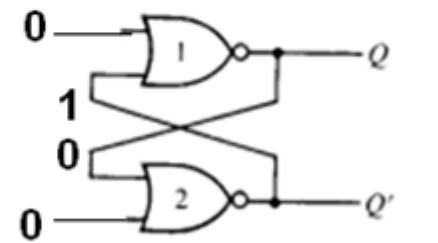
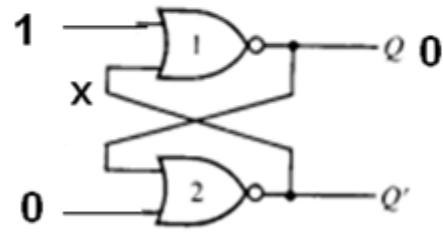
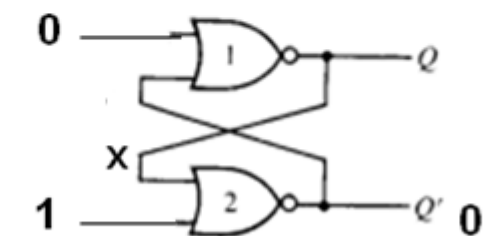
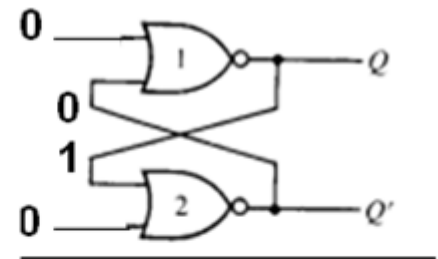
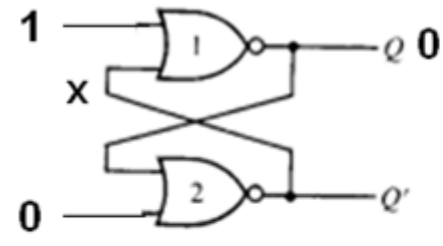
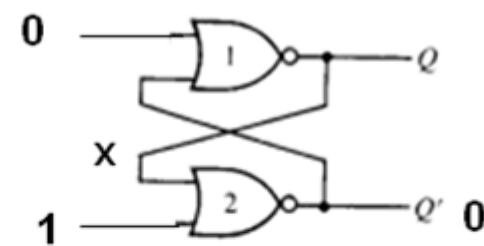
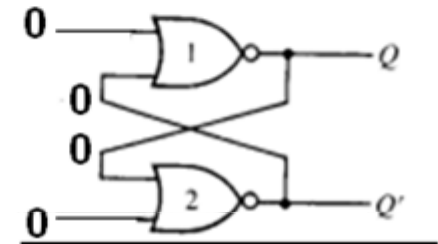
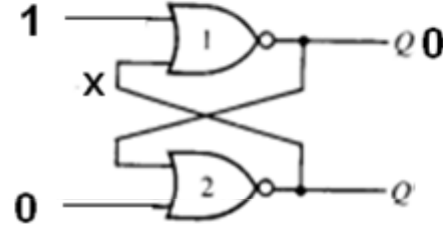
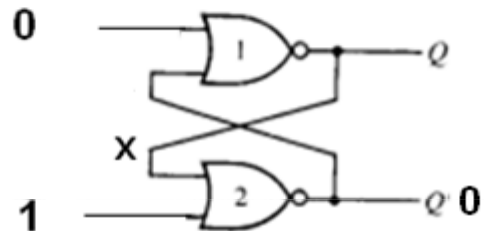


# Análisis de FF SR. Preparación

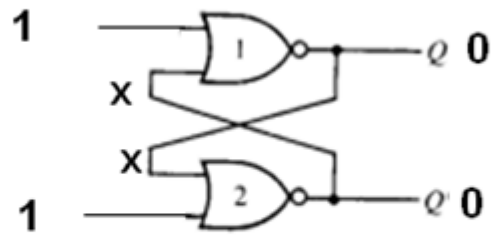


RS	Q	$\bar{Q}$
00		
01		
10		
11	0	0

	OR	NOR
00	0	1
01	1	0
10	1	0
11	1	0

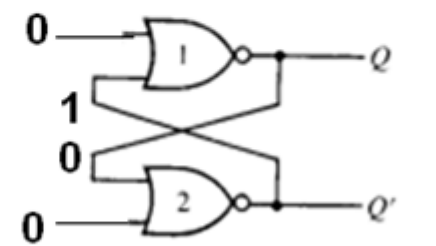
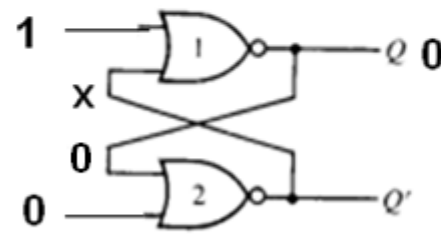
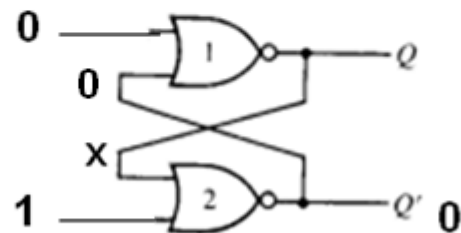
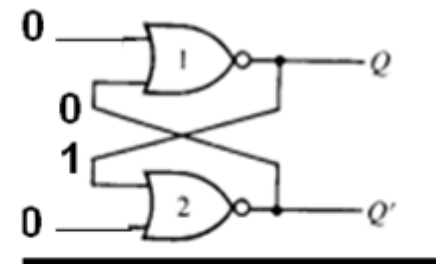
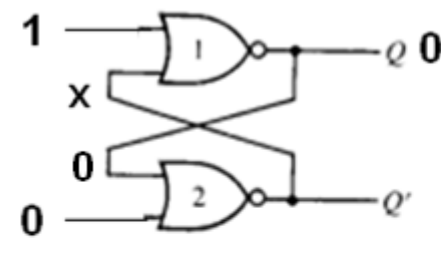
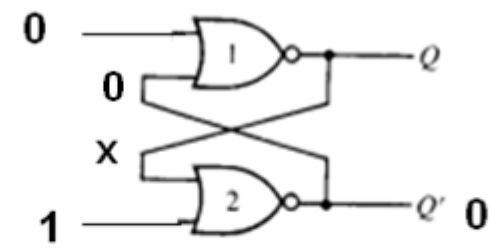
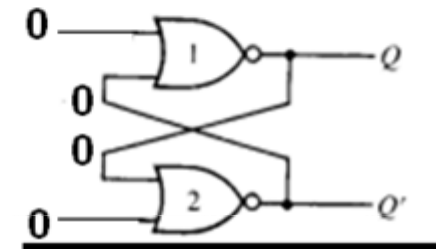
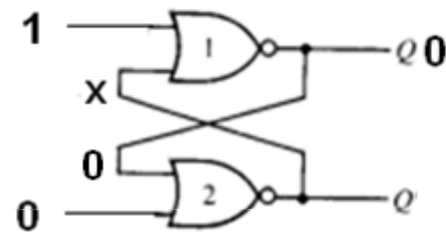
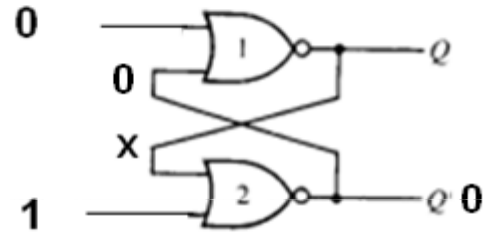


# Análisis de FF SR. Preparación

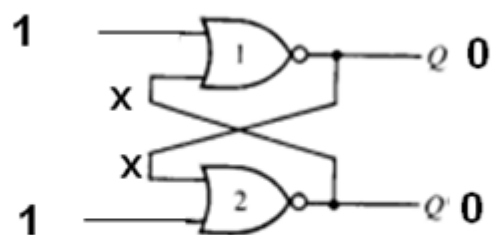


RS	Q	$\bar{Q}$
00		
01		
10		
11	0	0

	OR	NOR
00	0	1
01	1	0
10	1	0
11	1	0

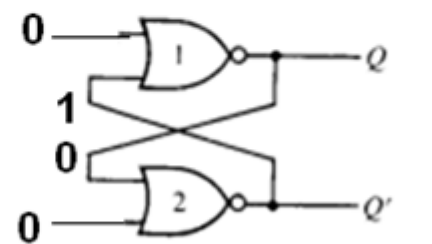
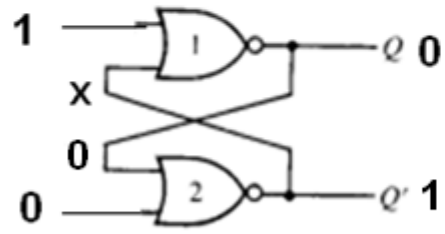
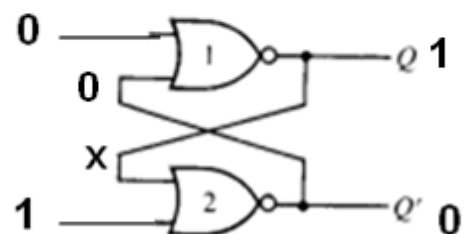
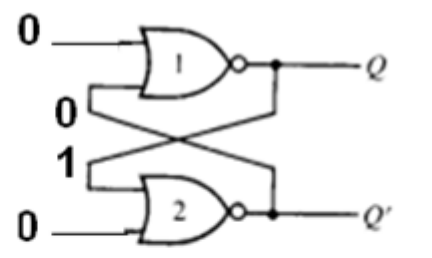
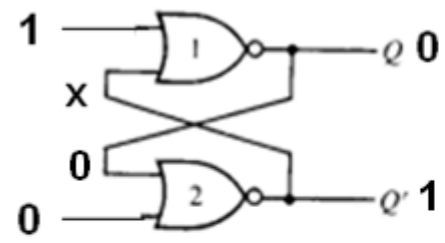
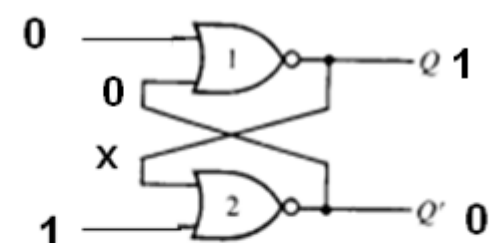
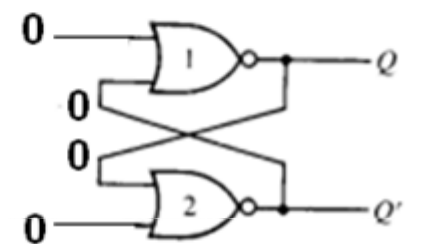
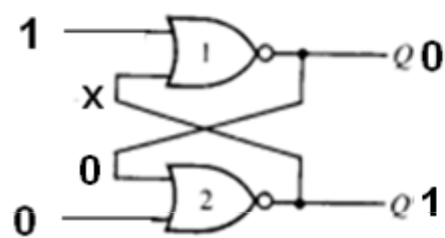
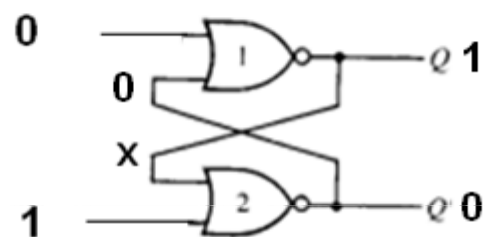


# Análisis de FF SR. Preparación

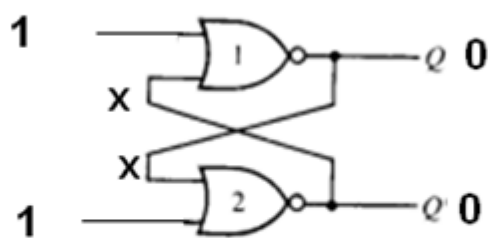


RS	Q	$\bar{Q}$
00		
01		
10		
11	0	0

	OR	NOR
00	0	1
01	1	0
10	1	0
11	1	0



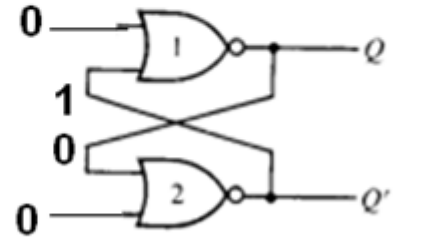
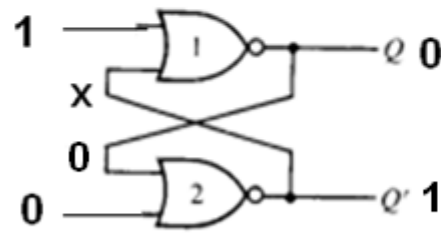
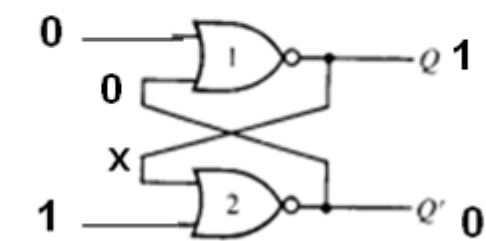
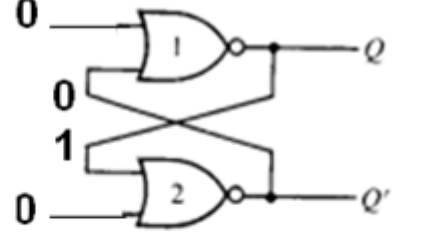
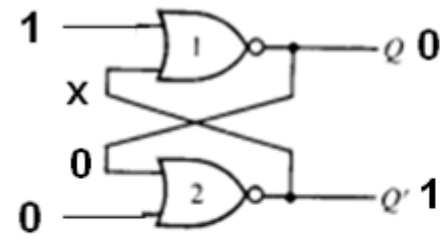
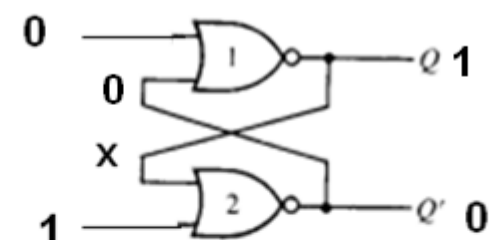
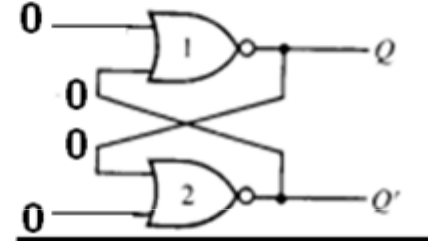
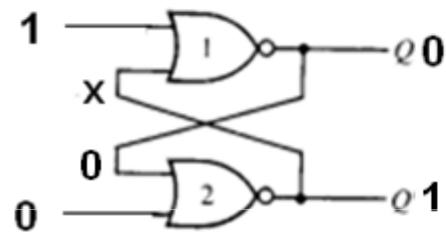
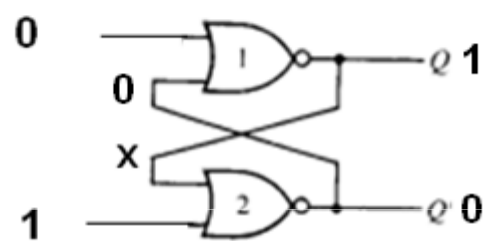
# Análisis de FF SR. Preparación



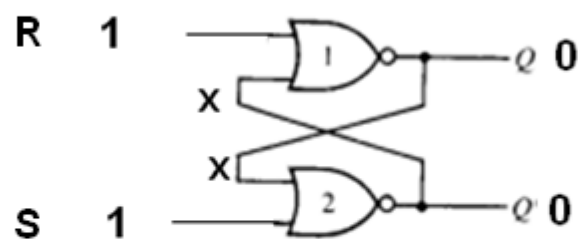
RS	Q	$\bar{Q}$
00		
01		
10		
11	0	0

←  $Q \neq \bar{Q}$  NO permitido

	OR	NOR
00	0	1
01	1	0
10	1	0
11	1	0



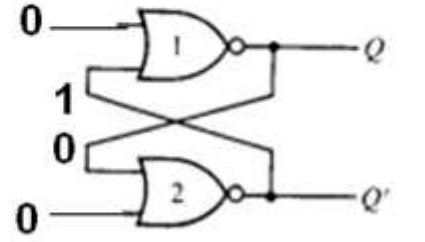
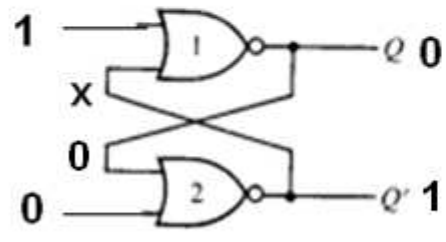
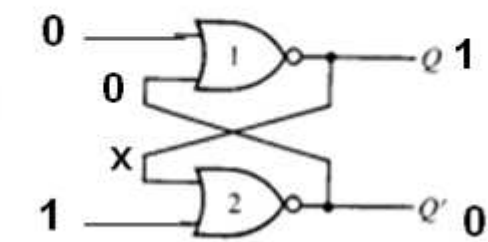
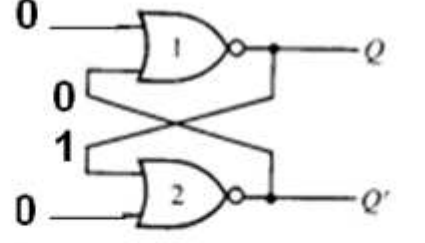
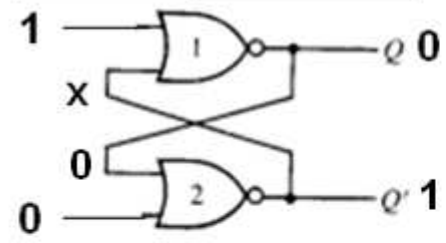
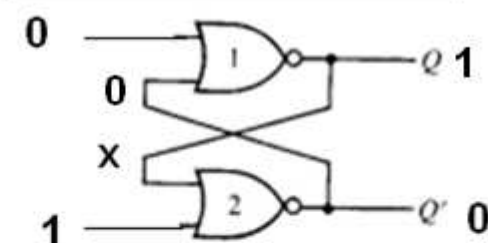
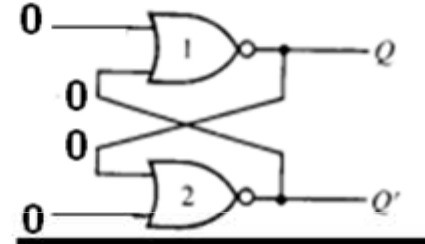
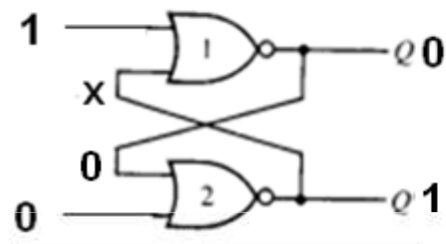
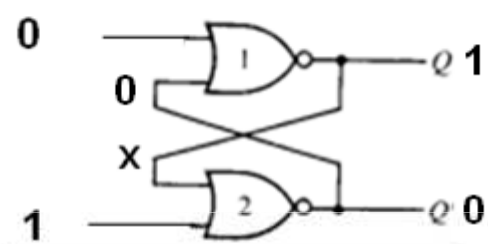
# Análisis de FF SR. Preparación



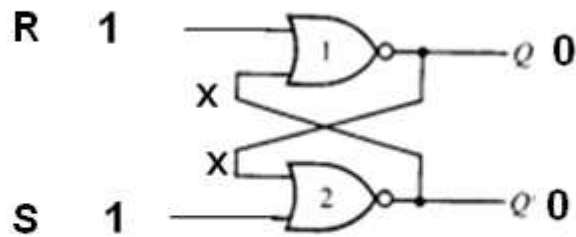
RS	Q	$\bar{Q}$
00		
01	1	0
10	0	1
11	0	0

←  $Q \neq \bar{Q}$  NO permitido

	OR	NOR
00	0	1
01	1	0
10	1	0
11	1	0



# Análisis de FF SR. Preparación

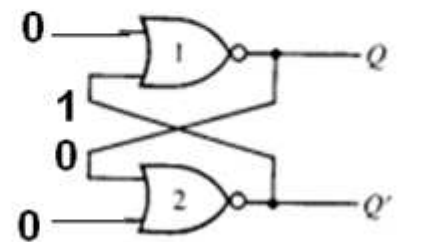
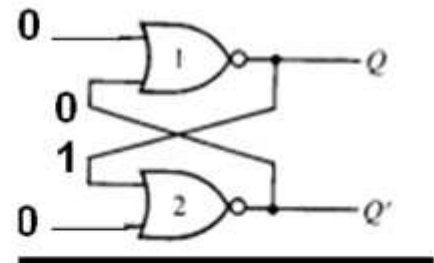
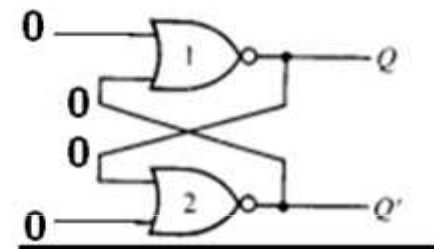
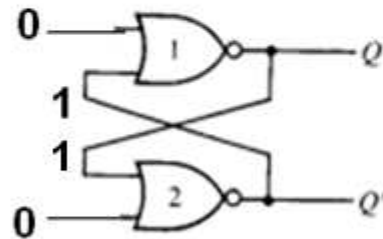


RS	Q	$\bar{Q}$
00		
01	1	0
10	0	1
11	0	0

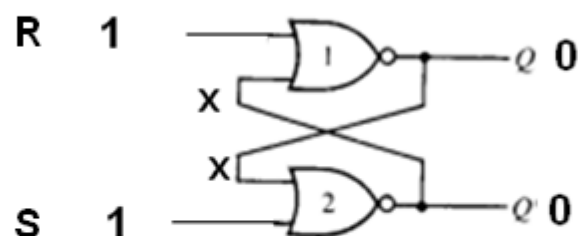
←  $Q \neq \bar{Q}$  NO permitido

	OR	NOR
00	0	1
01	1	0
10	1	0
11	1	0

para 00 suponemos todas las alternativas



# Análisis de FF SR. Preparación

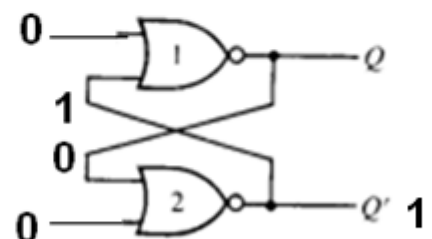
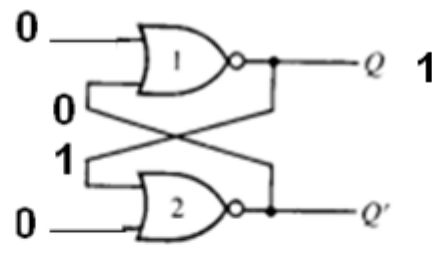
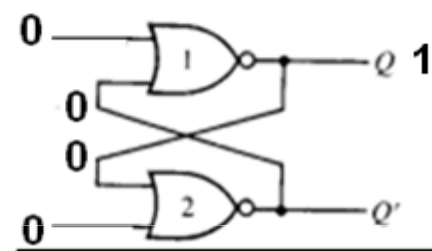
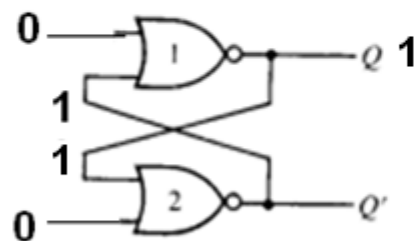


RS	Q	Q'
00		
01	1	0
10	0	1
11	0	0

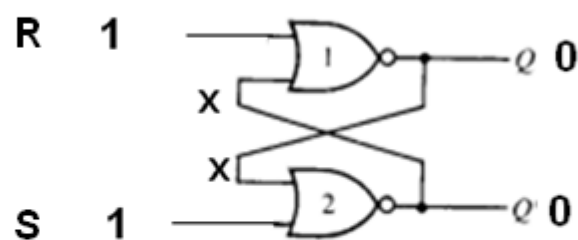
←  $Q \neq \bar{Q}$  NO permitido

	OR	NOR
00	0	1
01	1	0
10	1	0
11	1	0

para 00 suponemos todas las alternativas



# Análisis de FF SR. Preparación

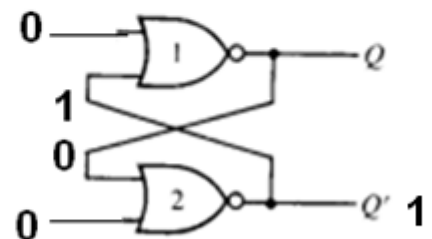
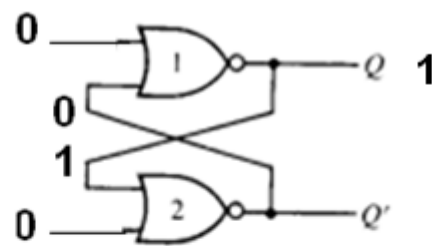
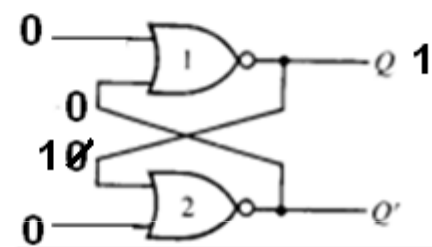
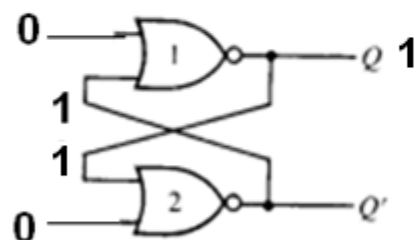


RS	Q	$\bar{Q}$
00		
01	1	0
10	0	1
11	0	0

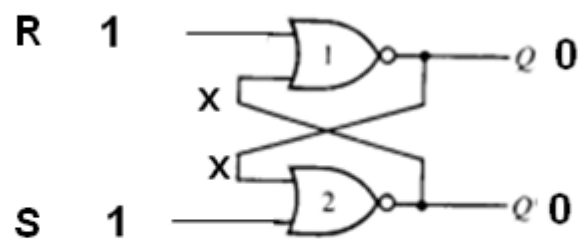
←  $Q \neq \bar{Q}$  NO permitido

	OR	NOR
00	0	1
01	1	0
10	1	0
11	1	0

para 00 suponemos todas las alternativas



# Análisis de FF SR. Preparación

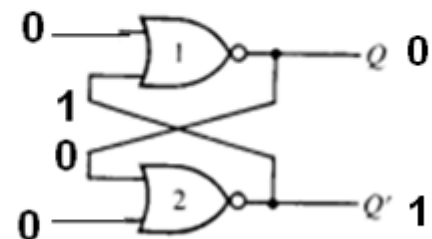
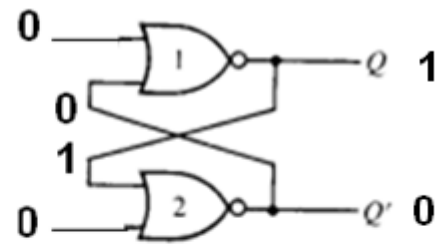
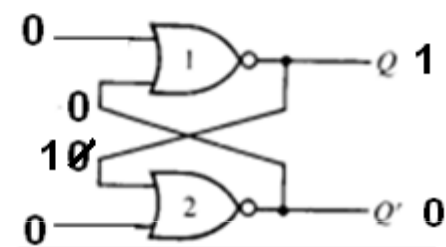
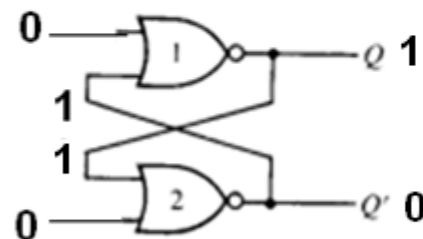


RS	Q	$\bar{Q}$
00		
01	1	0
10	0	1
11	0	0

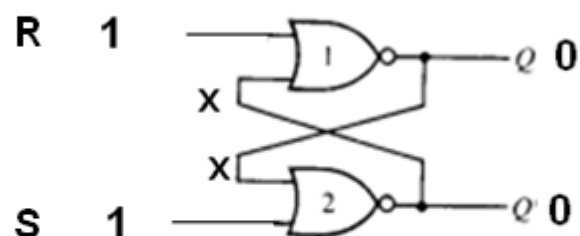
←  $Q \neq \bar{Q}$  NO permitido

	OR	NOR
00	0	1
01	1	0
10	1	0
11	1	0

para 00 suponemos todas las alternativas



# Análisis de FF SR. Preparación

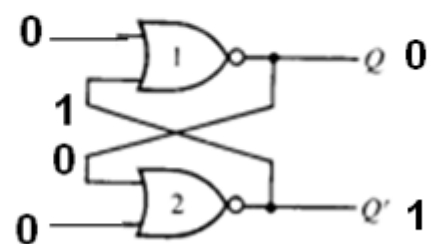
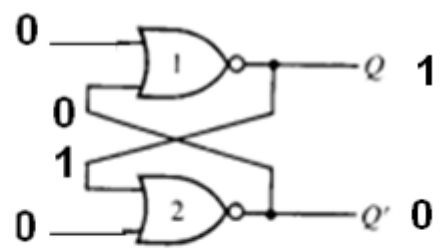
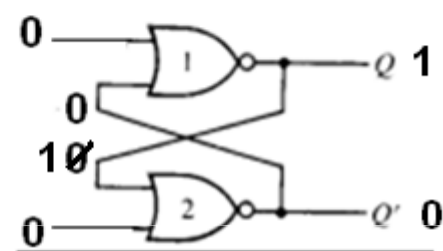
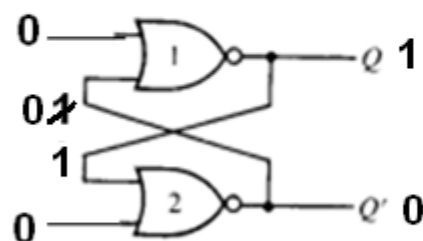


RS	Q	$\bar{Q}$
00		
01	1	0
10	0	1
11	0	0

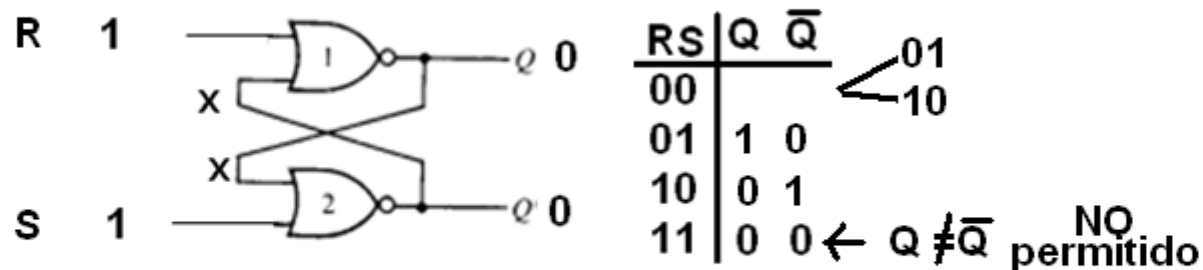
$\leftarrow Q \neq \bar{Q}$  NO permitido

	OR	NOR
00	0	1
01	1	0
10	1	0
11	1	0

para 00 suponemos todas las alternativas

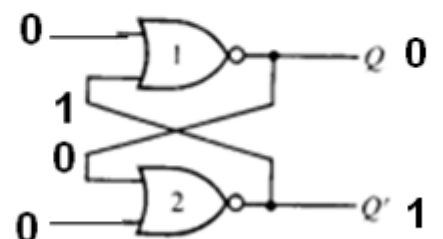
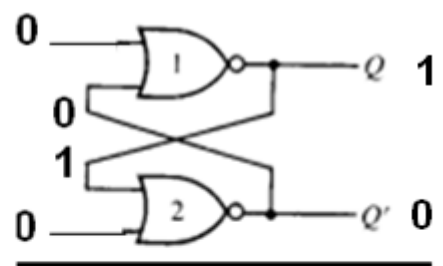
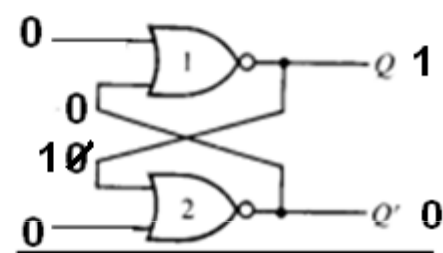
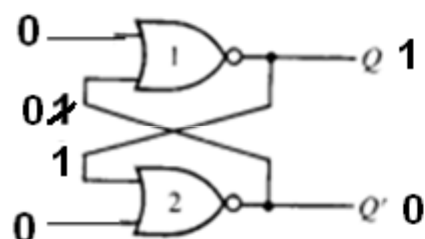


# Análisis de FF SR. Preparación

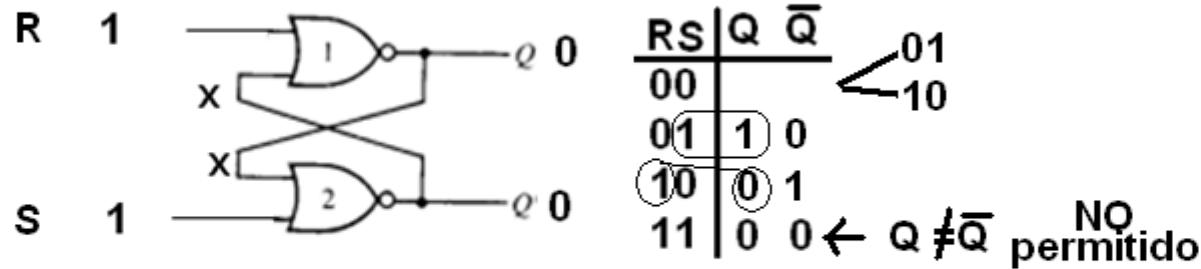


	OR	NOR
00	0	1
01	1	0
10	1	0
11	1	0

para 00 suponemos todas las alternativas

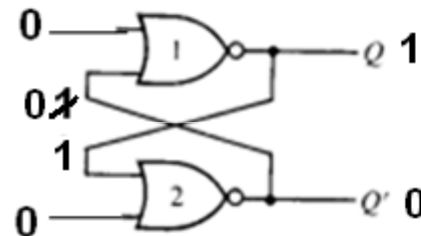


# Análisis de FF SR. Preparación



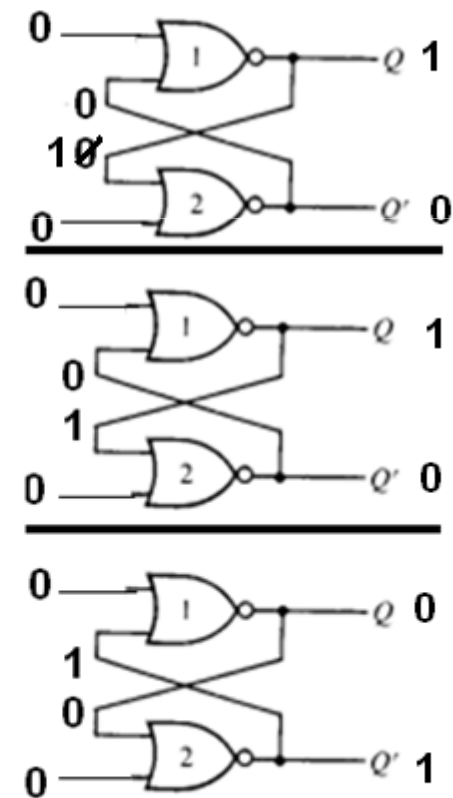
	OR	NOR
00	0	1
01	1	0
10	1	0
11	1	0

para 00 suponemos todas las alternativas

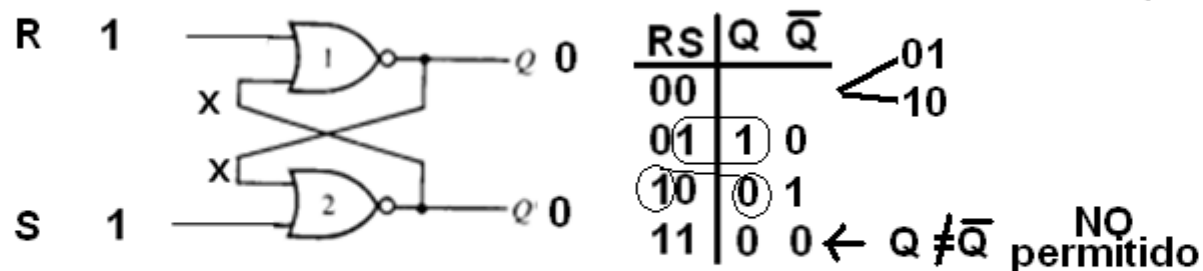


S Set=1  
R Reset=0

RS	Q+1	$\bar{Q}+1$
00 0		
00 1		
01 0	1	0
01 1	1	0
10 0	0	1
10 1	0	1
11 0		
11 1		



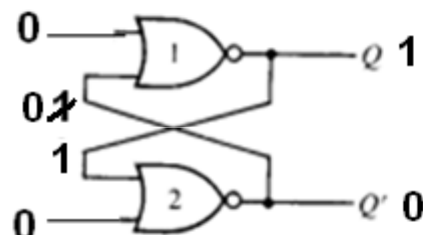
# Análisis de FF SR. Preparación



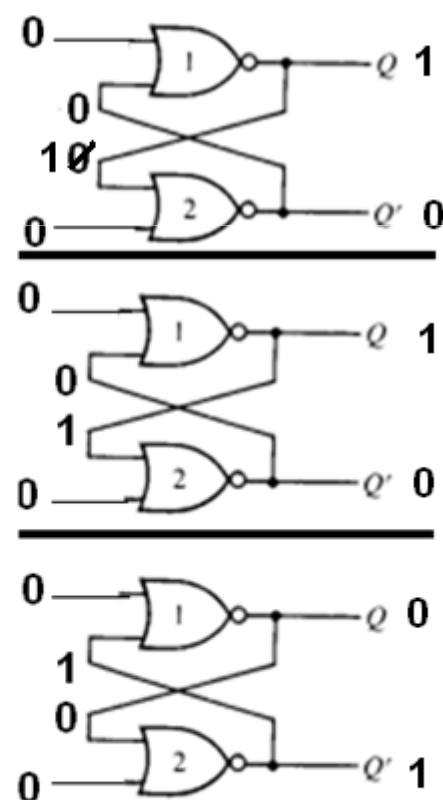
	OR	NOR
00	0	1
01	1	0
10	1	0
11	1	0

para 00 suponemos todas las alternativas

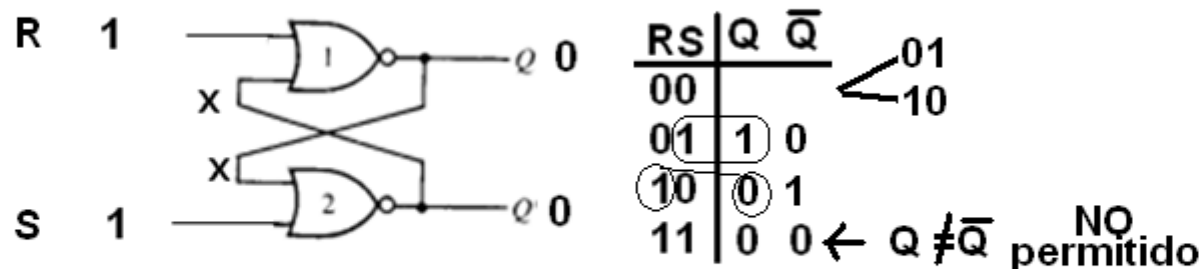
S Set=1  
R Reset=0



RS	Q+1	$\bar{Q}+1$
00	0	
00	1	
01	1	0
01	1	0
10	0	1
10	0	1
11	0	0
11	0	0



# Análisis de FF SR. Preparación

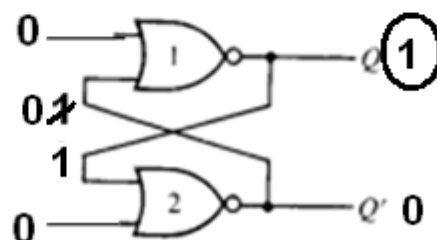
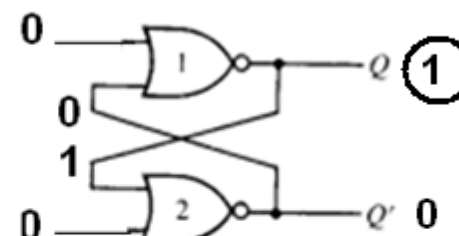
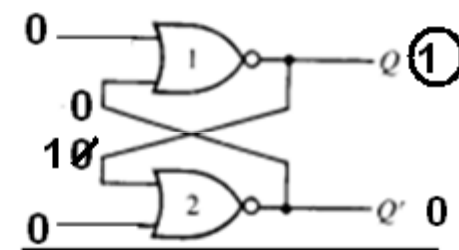
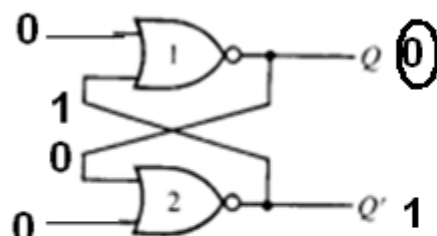


	OR	NOR
00	0	1
01	1	0
10	1	0
11	1	0

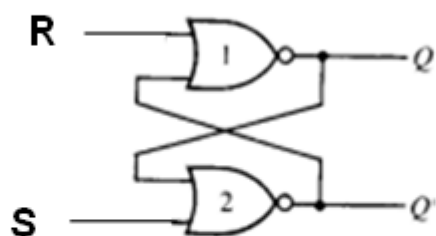
para 00 suponemos todas las alternativas

S Set=1  
R Reset=0

RSQ	Q+1	$\bar{Q}+1$
00 0	0	1
00 1	1	0
01 0	1	0
01 1	1	0
10 0	0	1
10 1	0	1
11 0	0	0
11 1	0	0



# FF SR conclusión de Analisis



RS	Q	Q̄
00		
01	1	0
10	0	1
11	0	0

Annotations: 01 and 10 are grouped together. An arrow points to the 11 row with the text "Q ≠ Q̄ NO permitido".

	OR	NOR
00	0	1
01	1	0
10	1	0
11	1	0

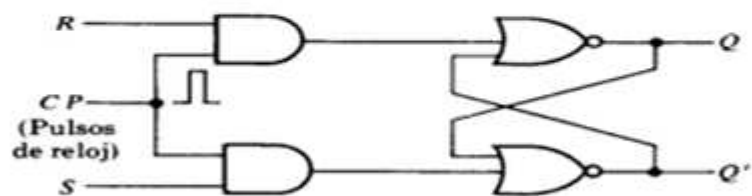
para 00 suponemos todas las alternativas

S Set=1  
R Reset=0

S	R	Q	Q'
1	0	1	0
0	0	1	0
0	1	0	1
0	0	0	1
1	1	0	0

(después de  $S = 1, R = 0$ ) =SR  
resultado anterior en memoria

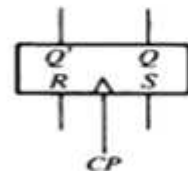
RSQ	Q+1	Q̄+1
00 0	0	1
00 1	1	0
01 0	1	0
01 1	1	0
10 0	0	1
10 1	0	1
11 0	0	0
11 1	0	0



(a) Diagrama lógico

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

(c) Tabla característica



(b) Símbolo gráfico

Q		SR			
		00	01	11	10
Q	0			X	1
	1	1		X	1

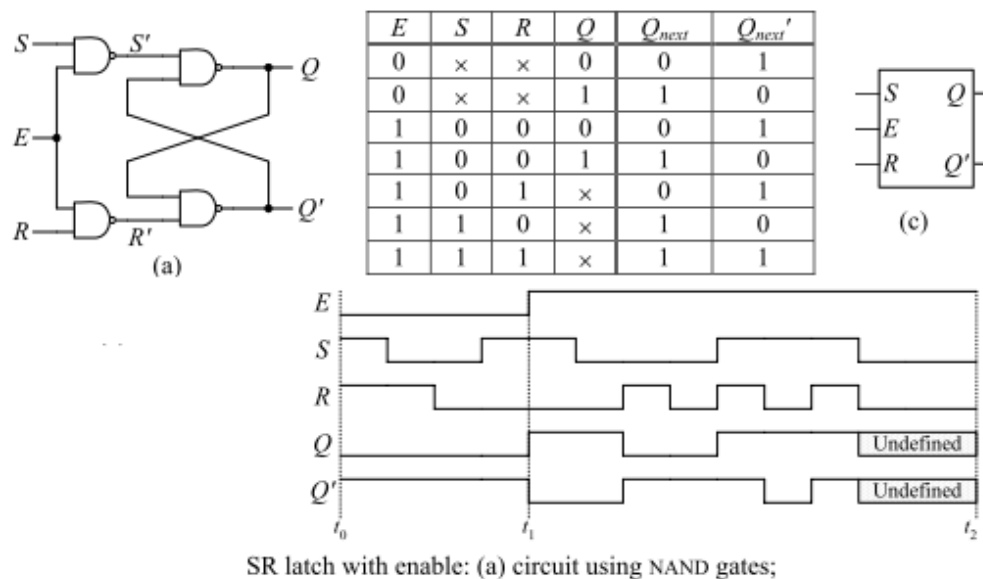
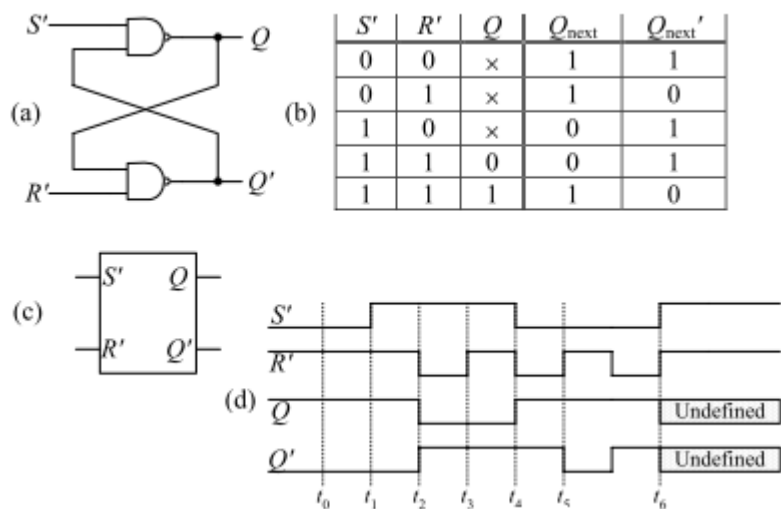
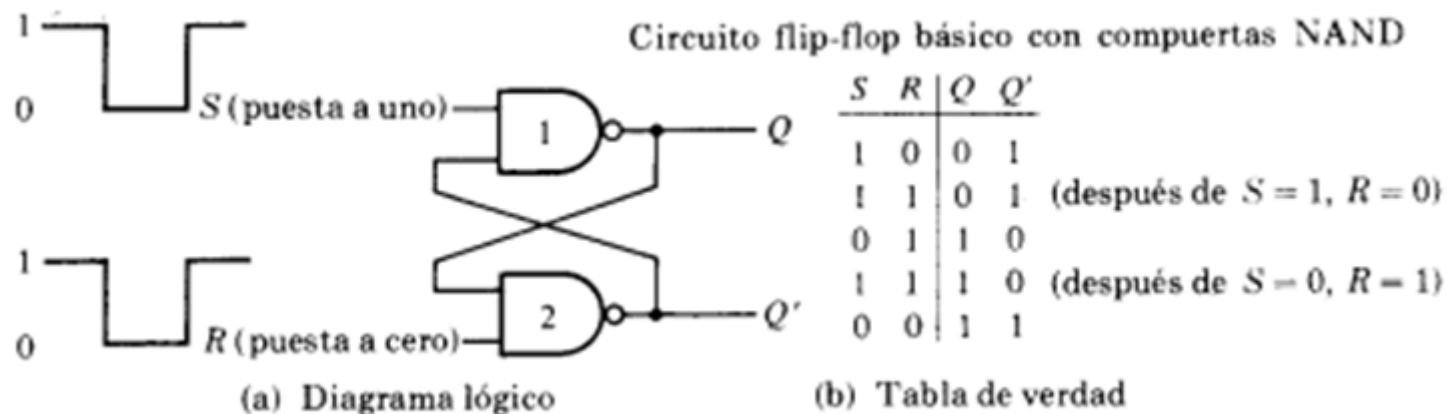
$$Q(t+1) = S + R'Q$$

$$SR = 0$$

(d) Ecuación característica

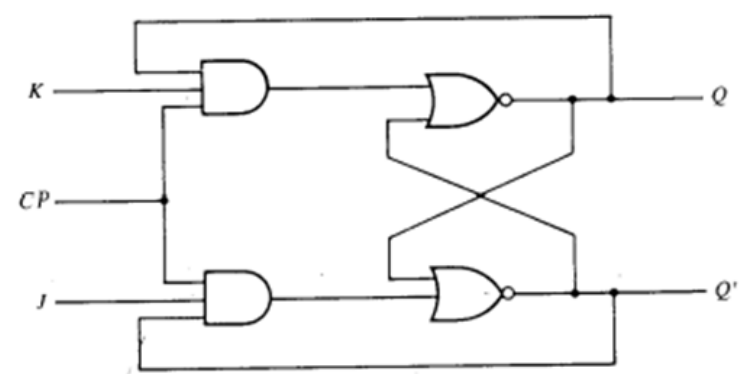
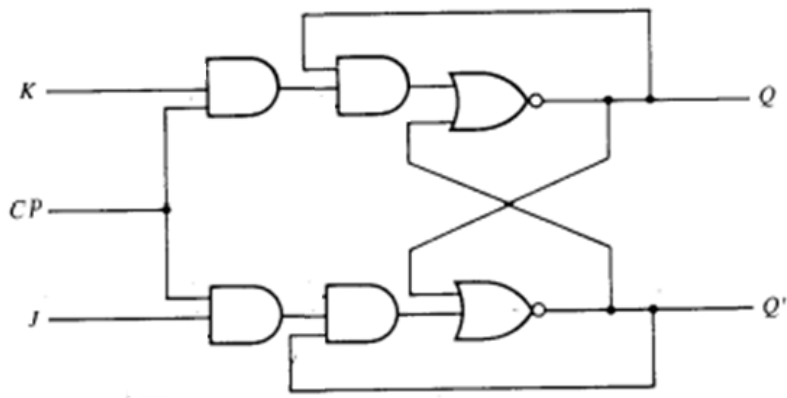
Flip-flop RS temporizado

# Análisis rápido para la NAND



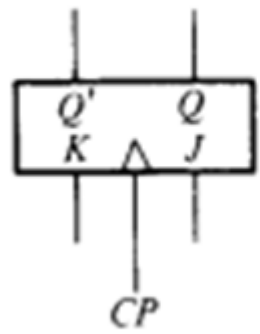
# FF JK

QSR	QJK	Q+1
000	000	0
001	001	0
010	010	1
011	011	1
100	100	1
101	101	0
110	110	1
111	111	0



JK	Q	Q̄
00	j	k mem
01	0	1
10	1	0
11	J̄	k̄ mem

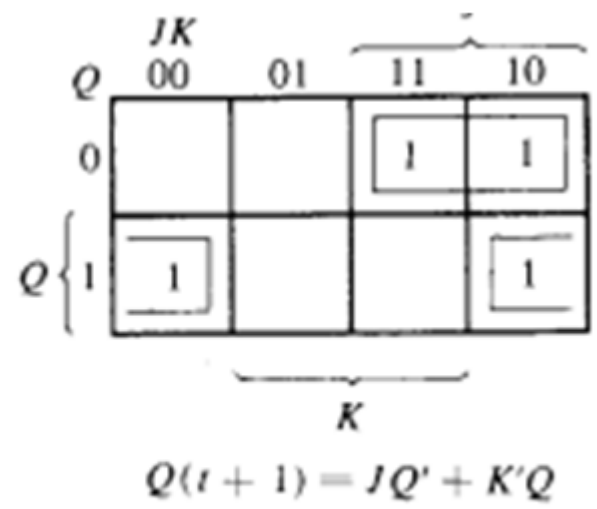
Si cp=1 JK empieza a funcionar  
 Si CP=0, and =0, se queda en estado de memoria.



(b) Símbolo gráfico

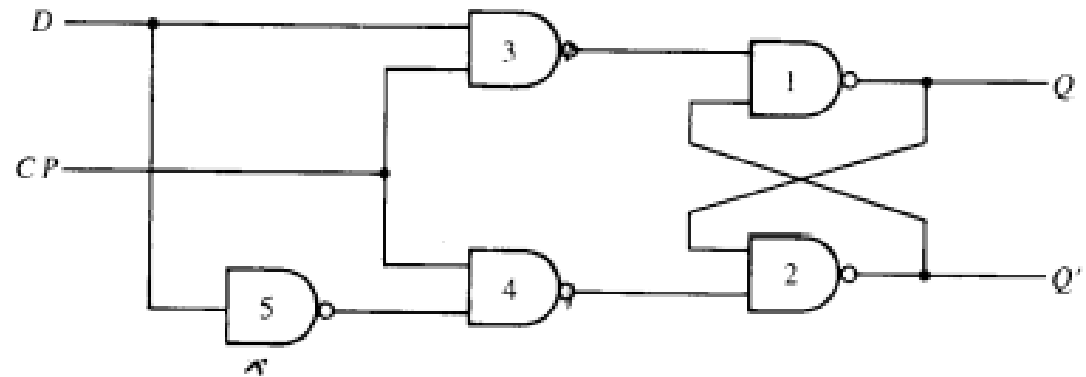
Q	J	K	Q(t+1)
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

(c) Tabla característica

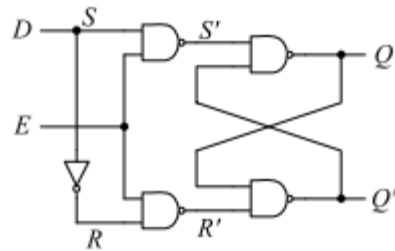


(d) Ecuación característica

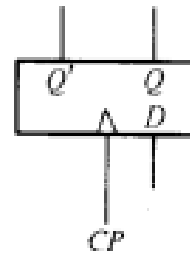
# FF D (Delay)



(a) Diagrama lógico con compuertas NAND



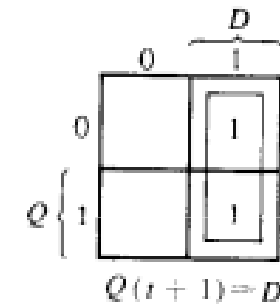
D latch with enable circuit using four NAND gates.



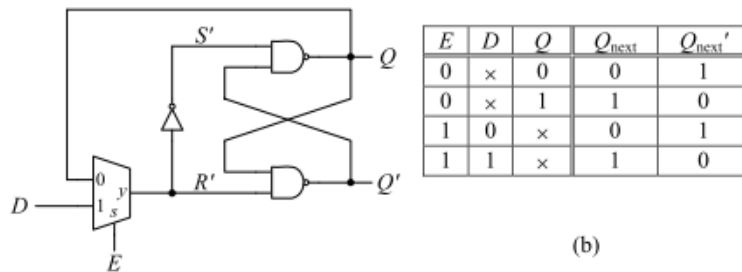
(b) Símbolo gráfico

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

(c) Tabla característica

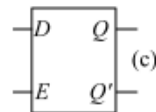


(d) Ecuación características

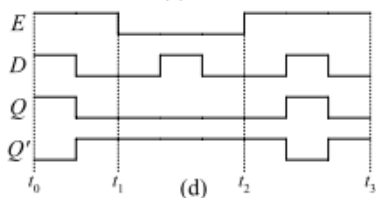


(b)

(a)



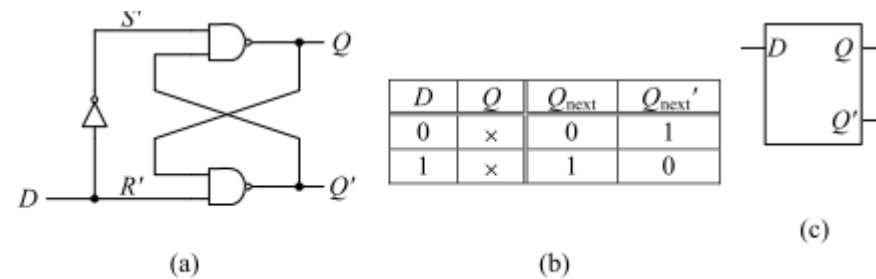
(c)



(d)

D latch with enable: (a) circuit; (b) truth table; (c) logic symbol; (d) sample trace.

## Flip-flop D temporizado



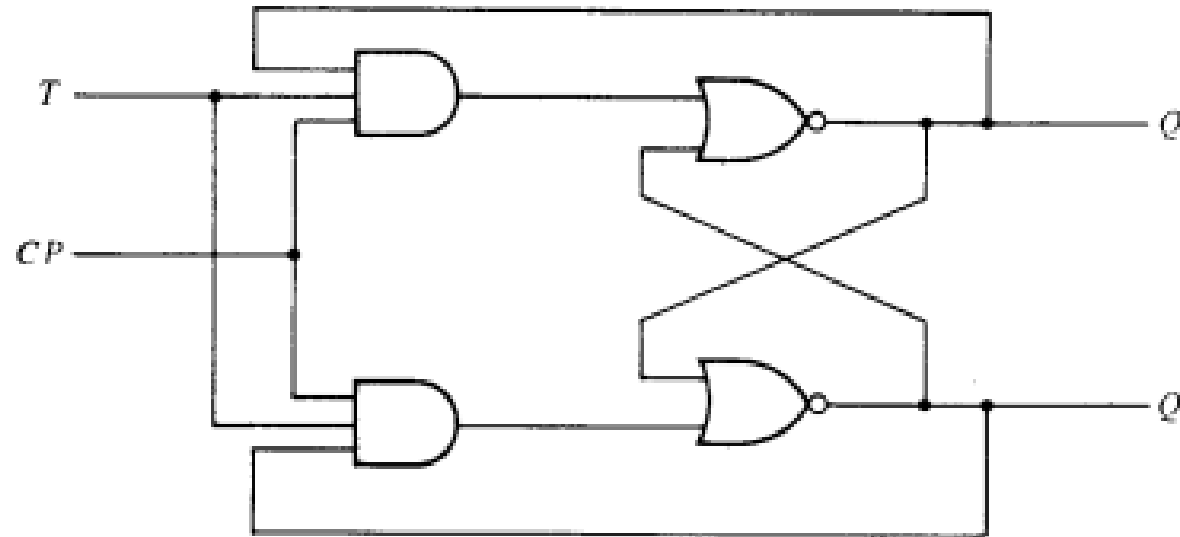
(a)

(b)

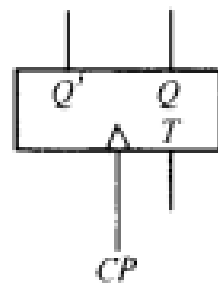
(c)

D latch: (a) circuit using NAND gates; (b) truth table; (c) logic symbol.

# FF T (Toggle)



(a) Diagrama lógico



(b) Símbolo gráfico

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

(c) Tabla característica

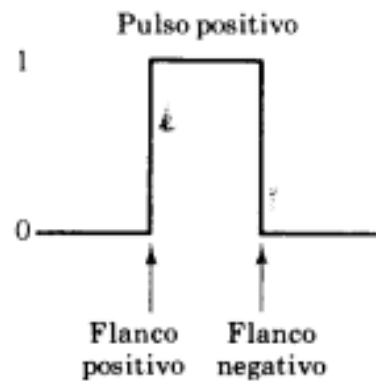
		$T$	
		0	1
$Q$	0		1
	1	1	

$$Q(t+1) = TQ' + TQ$$

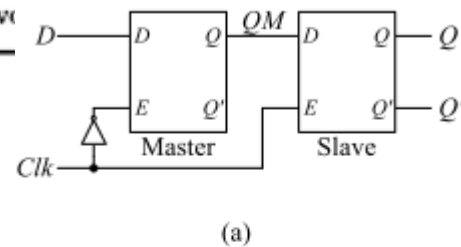
(d) Ecuación característica

Flip-flop  $T$  temporizado

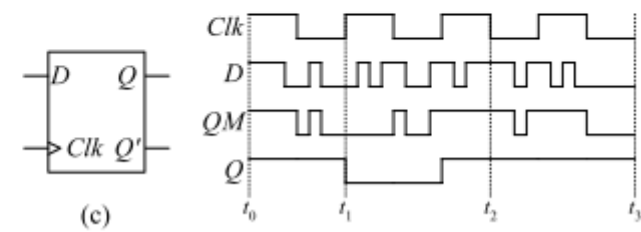
# FF Maestro-Esclavo



Definición de la transición de un pulso de reloj



Clk	D	Q	Q <sub>next</sub>	Q <sub>next</sub> '
0	x	0	0	1
0	x	1	1	0
1	x	0	0	1
1	x	1	1	0
↑	0	x	0	1
↑	1	x	1	0



Master-slave positive edge-triggered D flip-flop

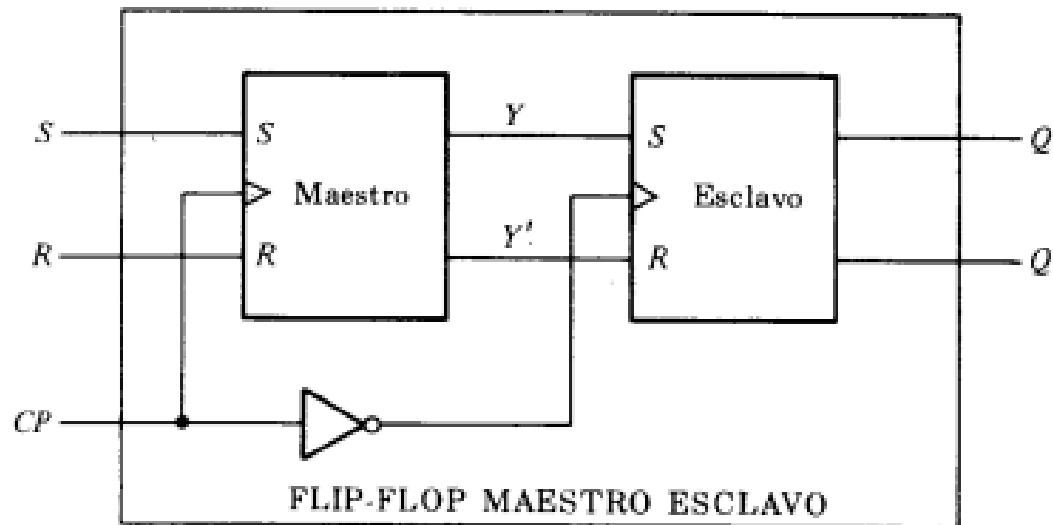
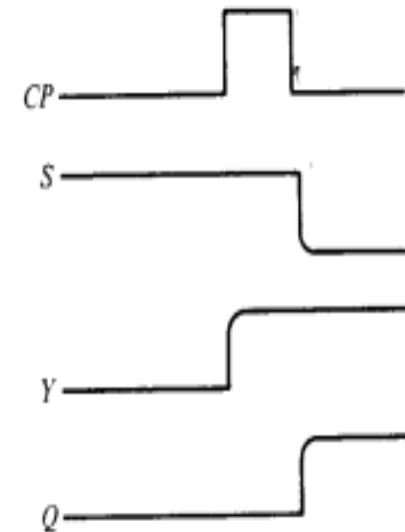
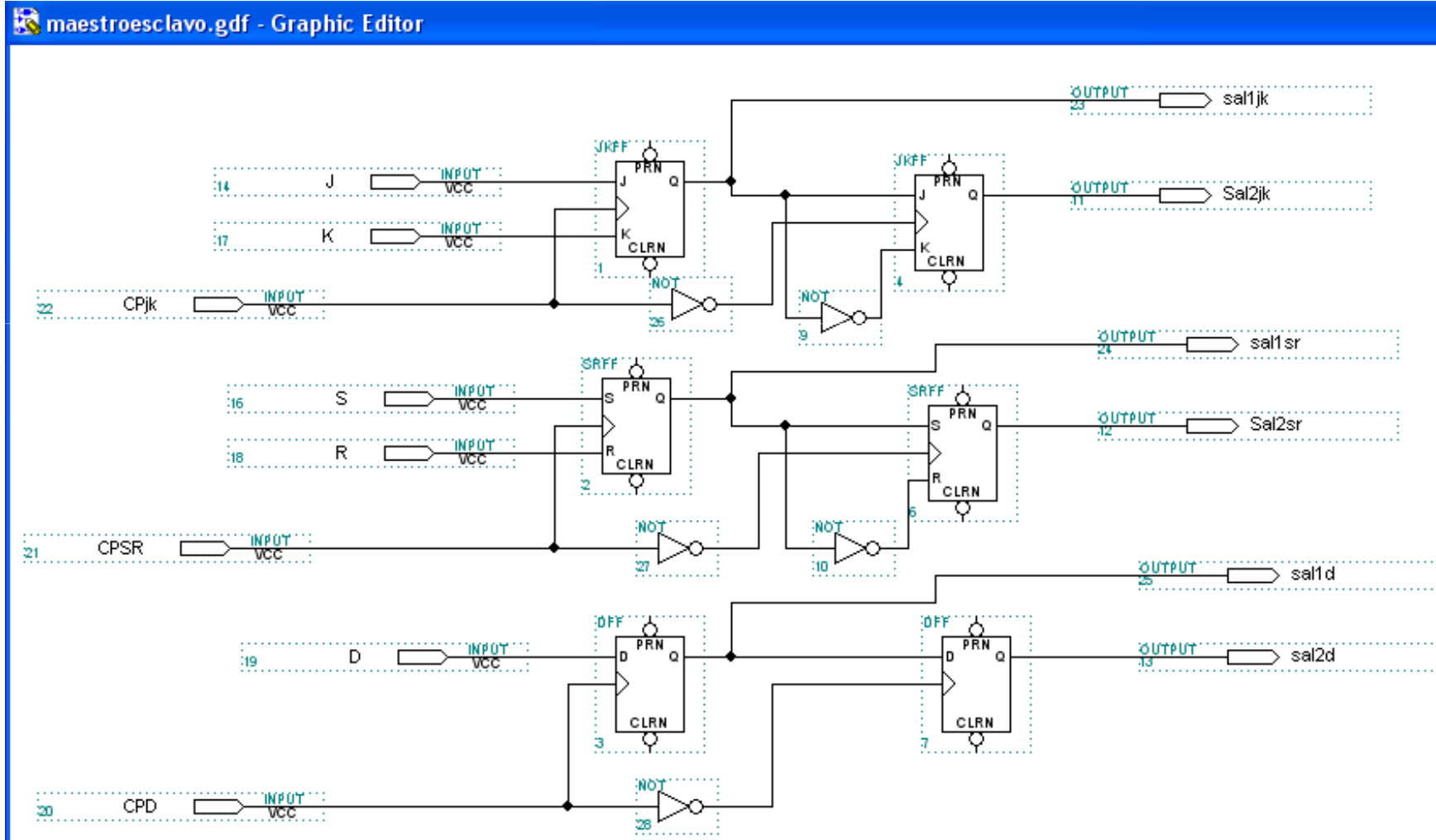


Diagrama lógico de un flip-flop maestro esclavo

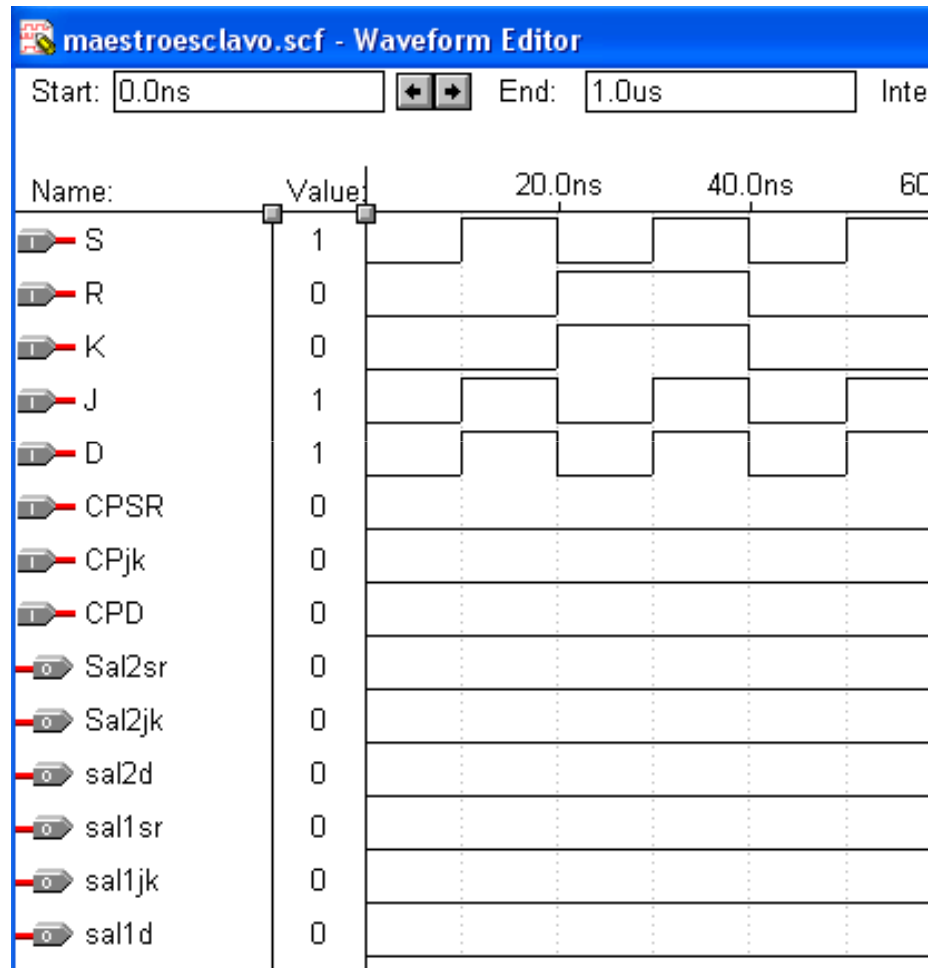


Relaciones de tiempo de un flip-flop maestro esclavo

# Maestro esclavo simulación

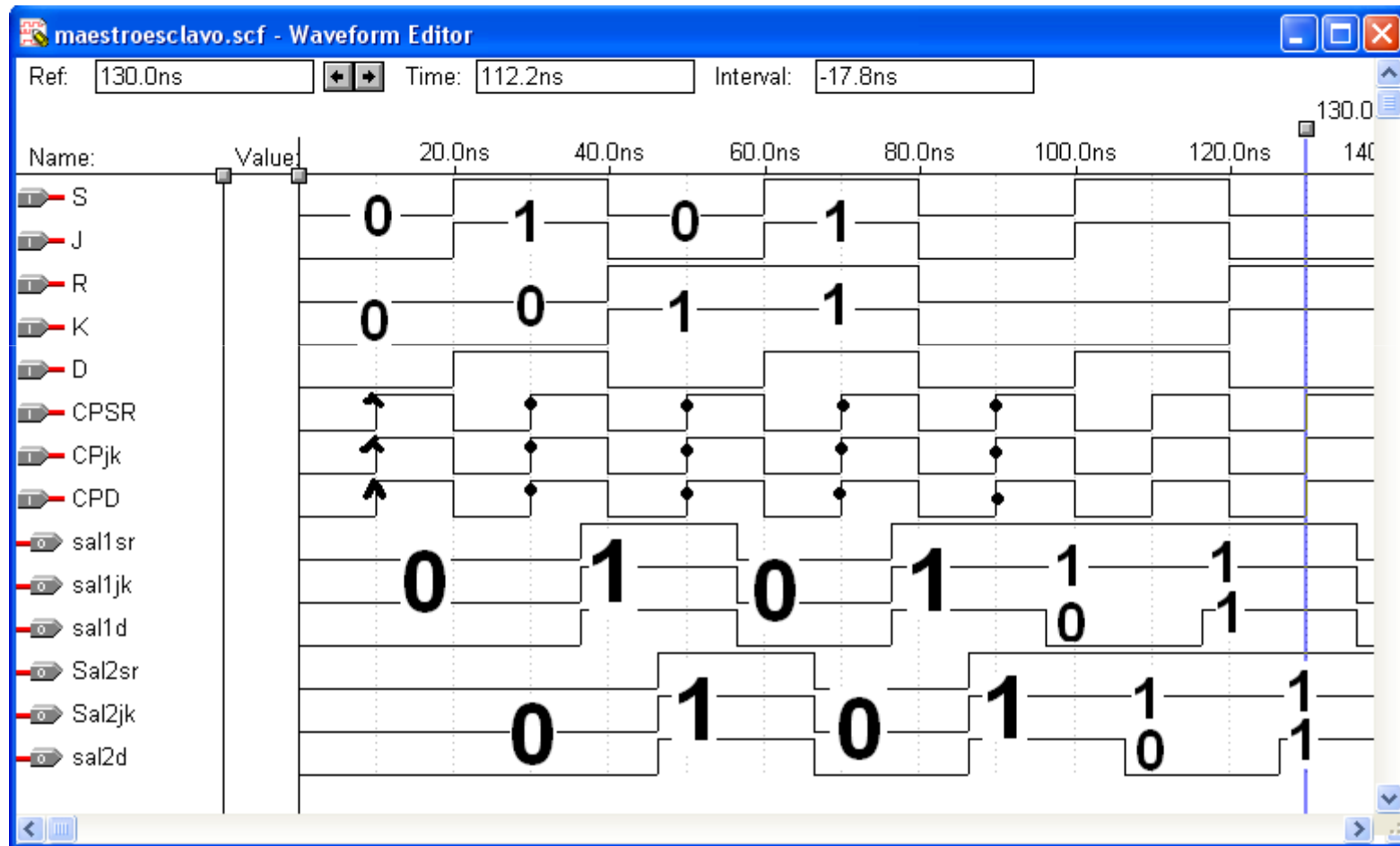


# Maestro esclavo Con Reloj=0



No trabaja el  
ningún FF

# Maestro esclavo Cp=1 Trabaja FF



# FF JK con borrado

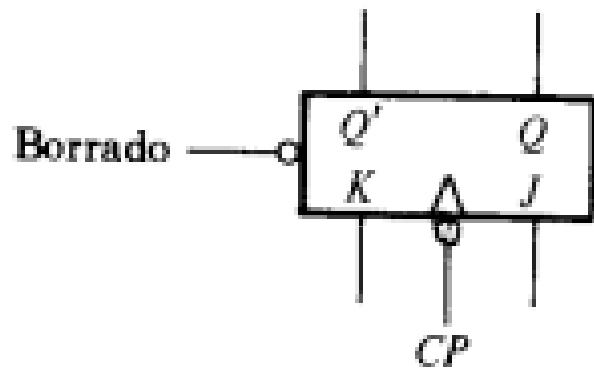
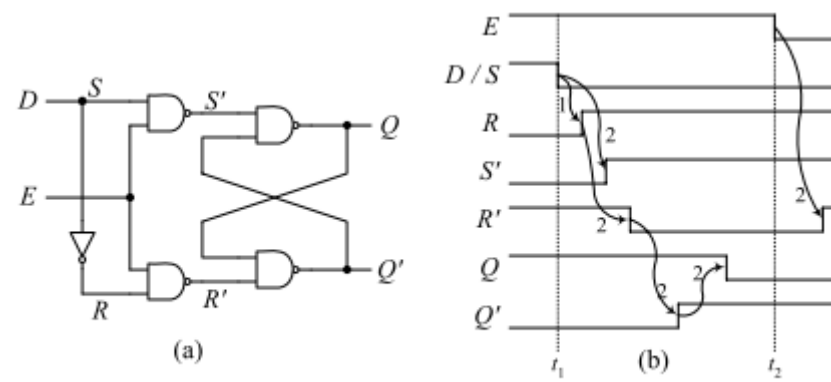
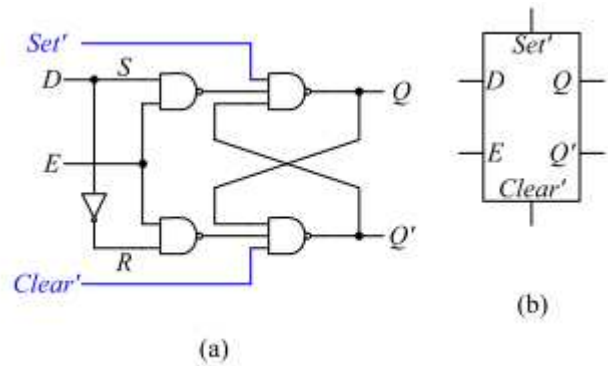


Tabla de función

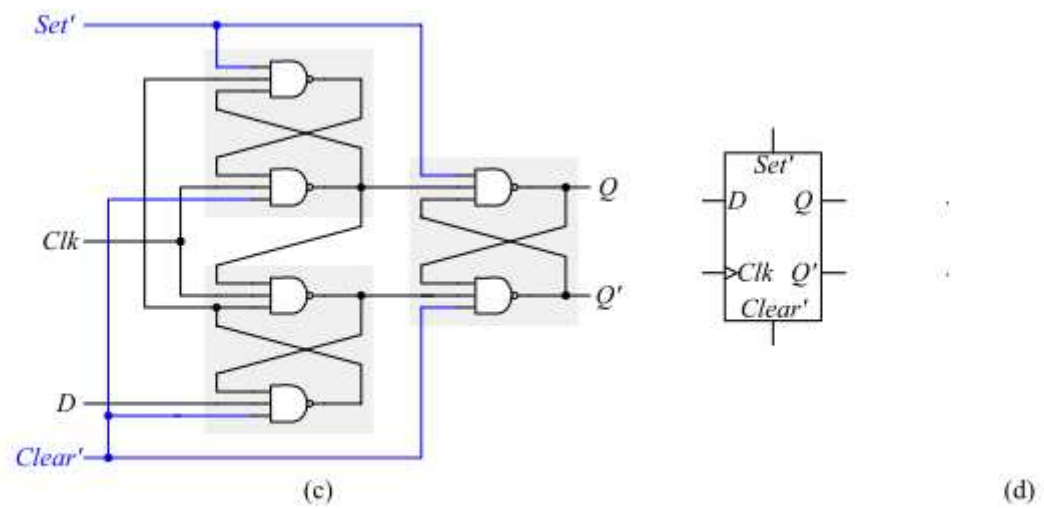
Entradas				Salidas	
Borrado	Reloj	J	K	Q	Q'
0	X	X	X	0	1
1	↓	0	0	No cambio	
1	↓	0	1	0	1
1	↓	1	0	1	0
1	↓	1	1	Conmuta	

Flip-flop JK con entrada directa de puesta a cero

# Entradas asíncronas



D latch with enable: (a) circuit; (b) timing diagram with delays.



Storage elements with asynchronous inputs: (a) D latch with active-low set and clear; (b) logic symbol for (a); (c) D edge-triggered flip-flop with active-low set and clear; (d) logic symbol for (c).

# VHDL

```
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;

ENTITY D_latch_with_enable IS PORT (
    D, Enable: IN STD_LOGIC;
    Q: OUT STD_LOGIC);
END D_latch_with_enable;

ARCHITECTURE Behavior OF D_latch_with_enable IS
BEGIN
    PROCESS(D, Enable)
    BEGIN
        IF (Enable = '1') THEN
            Q <= D;
        END IF;
    END PROCESS;
END Behavior;
```

```
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;

ENTITY no_memory_element IS PORT (
    D, Enable: IN STD_LOGIC;
    Q: OUT STD_LOGIC);
END no_memory_element;

ARCHITECTURE Behavior OF no_memory_element IS
BEGIN
    PROCESS(D, Enable)
    BEGIN
        IF (Enable = '1') THEN
            Q <= D;
        ELSE
            Q <= '0';
        END IF;
    END PROCESS;
END Behavior;
```

Sample VHDL description of a combinational circuit.

```
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;

ENTITY D_flipflop IS PORT (
    D, Clock: IN STD_LOGIC;
    Q: OUT STD_LOGIC);
END D_flipflop;

ARCHITECTURE Behavior OF D_flipflop IS
BEGIN
    PROCESS(Clock) -- sensitivity list is used
    BEGIN
        IF (Clock'EVENT AND Clock = '1') THEN
            Q <= D;
        END IF;
    END PROCESS;
END Behavior;
```

Behavioral VHDL code for a positive edge-triggered D flip-flop using an IF statement.