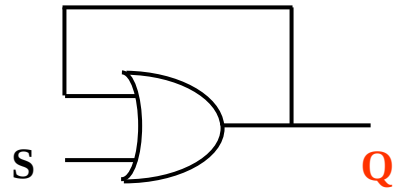
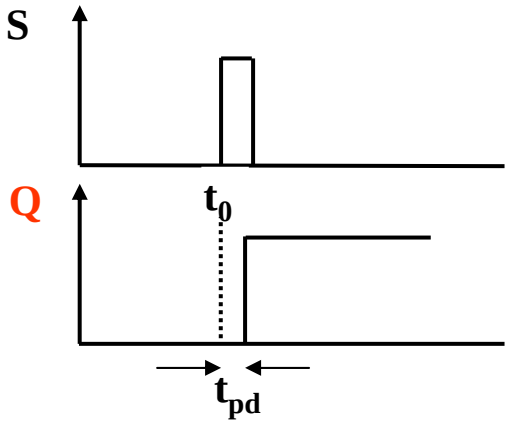
FF – PRINCIPIOS



S	Q
0	0
1	1
0	1

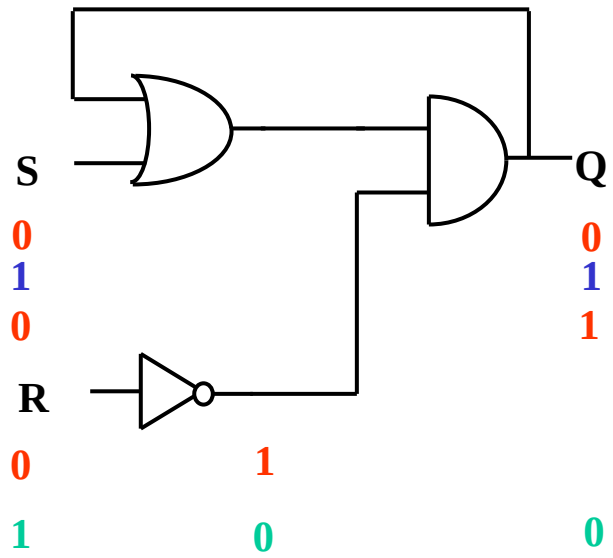
$Q_{t+1} = (S + Q)_t$

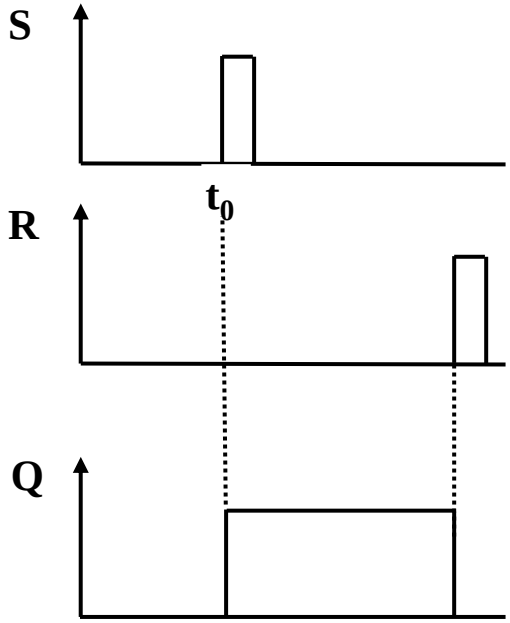
Futuro Presente

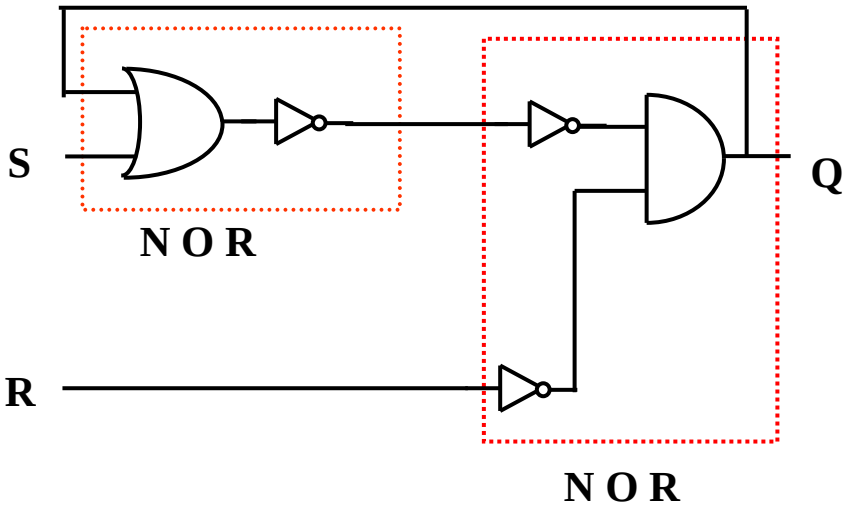


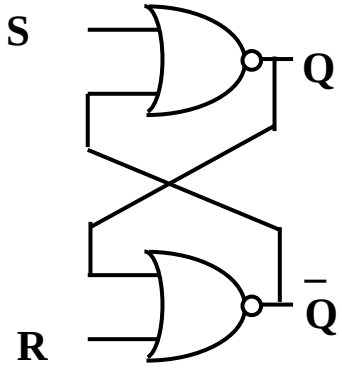
EN ESTE CIRCUITO NO HAY FORMA DE QUE LA SALIDA Q VUELVA A VALOR CERO "0", SIEMPRE QUEDA EN UNO "1". PROPONEMOS EL SIGUIENTE CIRCUITO EL CUAL PERMITE QUE LA SALIDA Q VAYA A CERO

MEMORIA CON BORRADO`-BIESTABLE





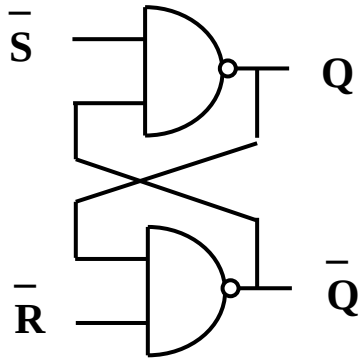




Pres	**Fut**
S R	**Q_{t+1}**
0 0	**Qt**
0 1	**0**
1 0	**1**
1 1	**NO**

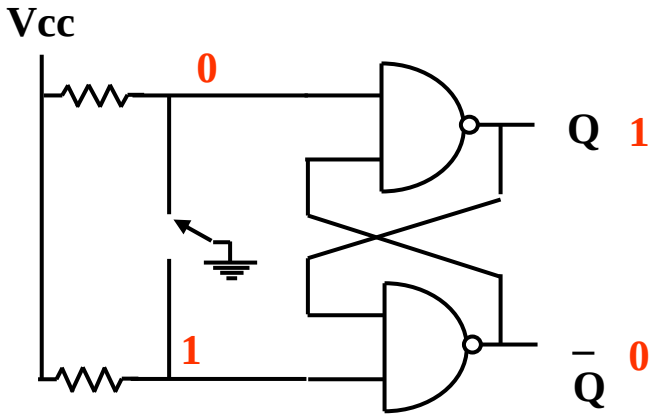
BIESTABLES

NAND

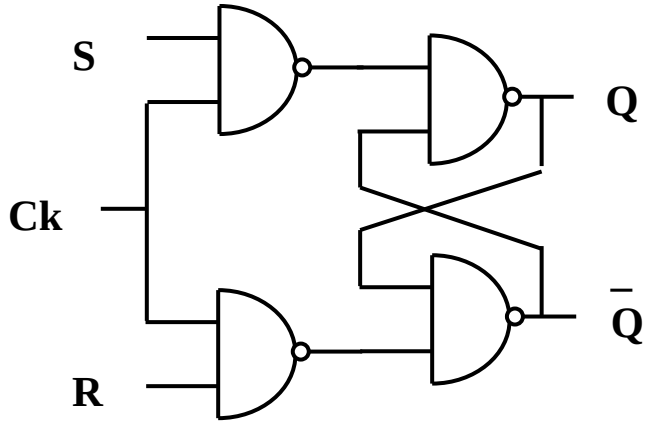


Pres		Fut
S	R	Qt+1
0	0	NO
0	1	1
1	0	0
1	1	Qt

APLICACION



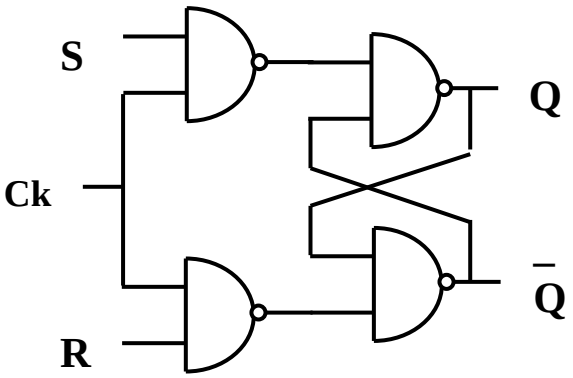
BIESTABLE SR SINCRONICO

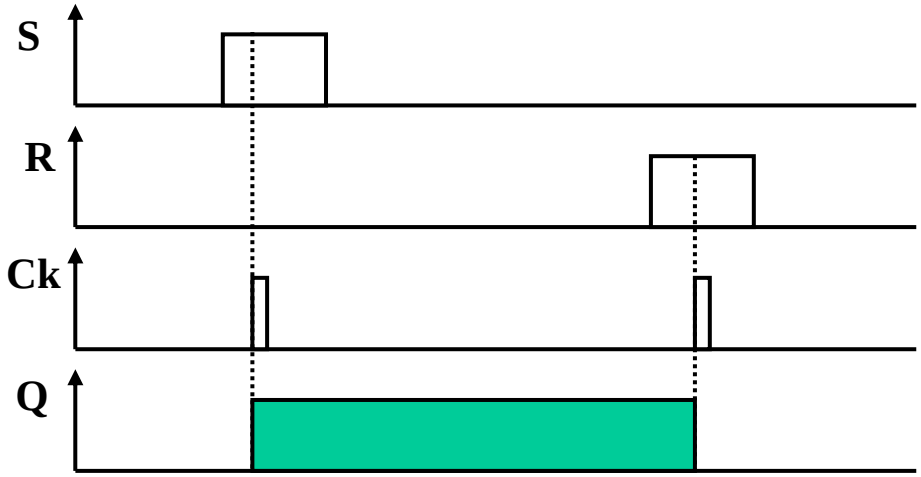


t		t+1
Pres		Fut
S	R	Qt+1
0	0	Qt
0	1	0
1	0	1
1	1	NO

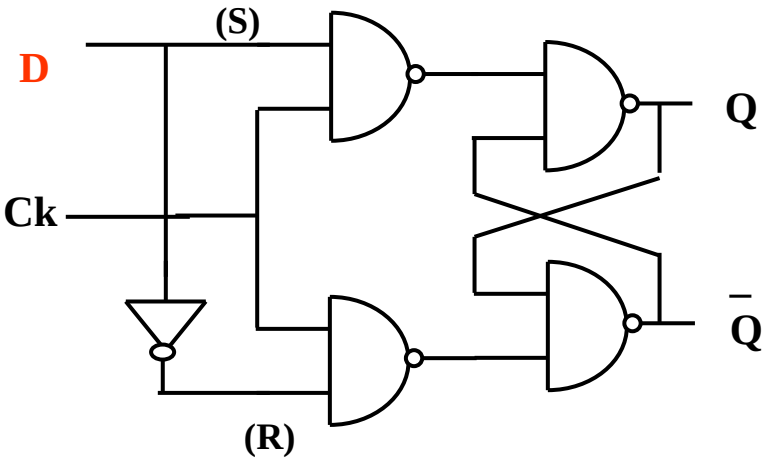
BIESTABLE- TIMING

BIESTABLE SR SINCRONICO





BIESTABLE D SINCRONICO - LATCH

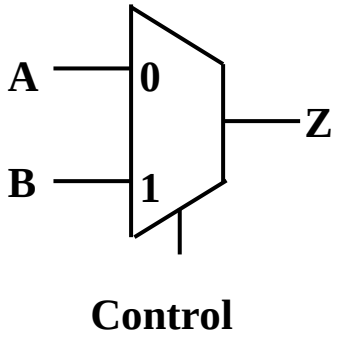


t	t+1
Pres	Fut
S R	Q_{t+1}
0 1	0
1 0	1

D

t	t+1
D	Q
0	0
1	1

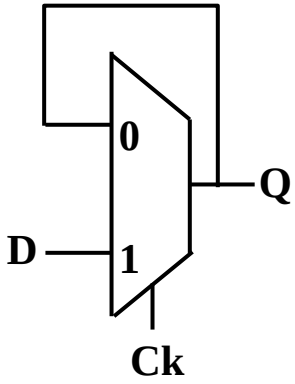
LATCH D CON MUX



Control	Z
0	A
1	B

$$Z = Control \cdot B + \overline{Control} \cdot A$$

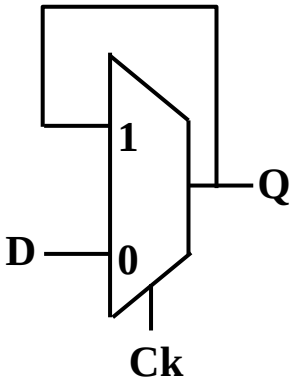
LATCH POSITIVO



Ck = 0, Q mem

Ck = 1, Q = D

LATCH NEGATIVO

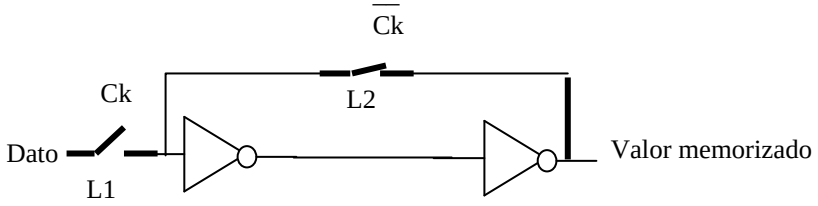


Ck=0, Q=D

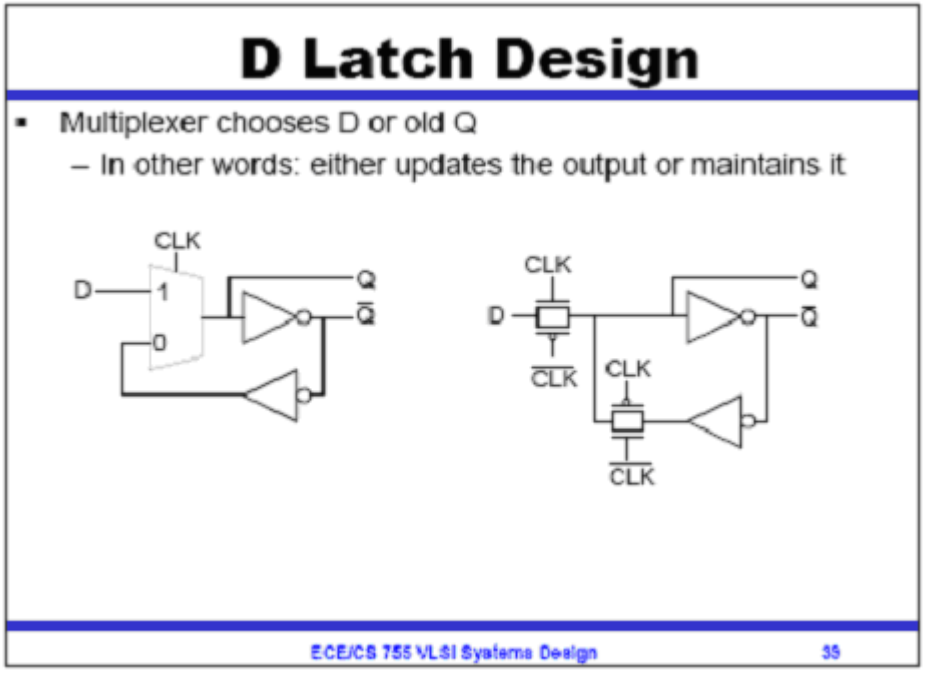
Ck=1, Q:mem

LATCH CON TG

Implementacion de un Latch con inversores y TG (Transmision Gate)

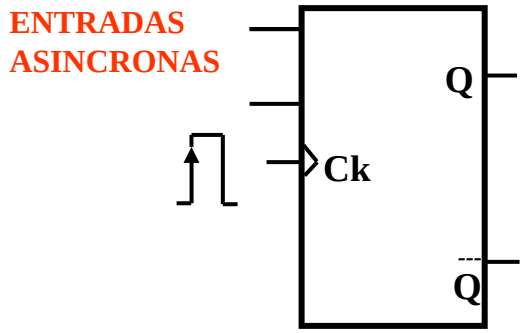


Como se observa ambas llaves trabajan en contraposición (L1: on, L2: off).
Si L1 esta cerrada el dato de entrada se repite a la salida (L2: off), cuando se abre L1 y se cierra L2 el Dato queda memorizado
En el dibujo que sigue observamos el diseño de un Latch con TG

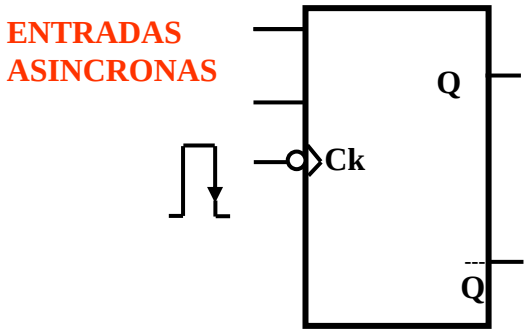


FLIP-FLOPS SINCRONIZADO POR RELOJ

La mayoria dispara por flanco (borde) y el símbolo que los identifica es

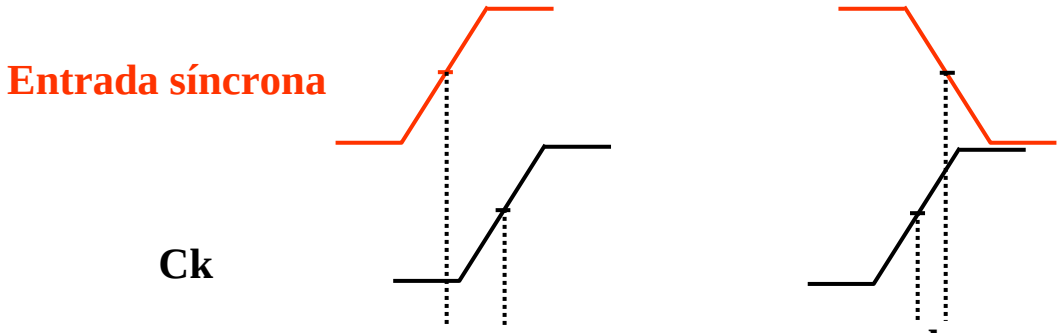


Flanco Positivo



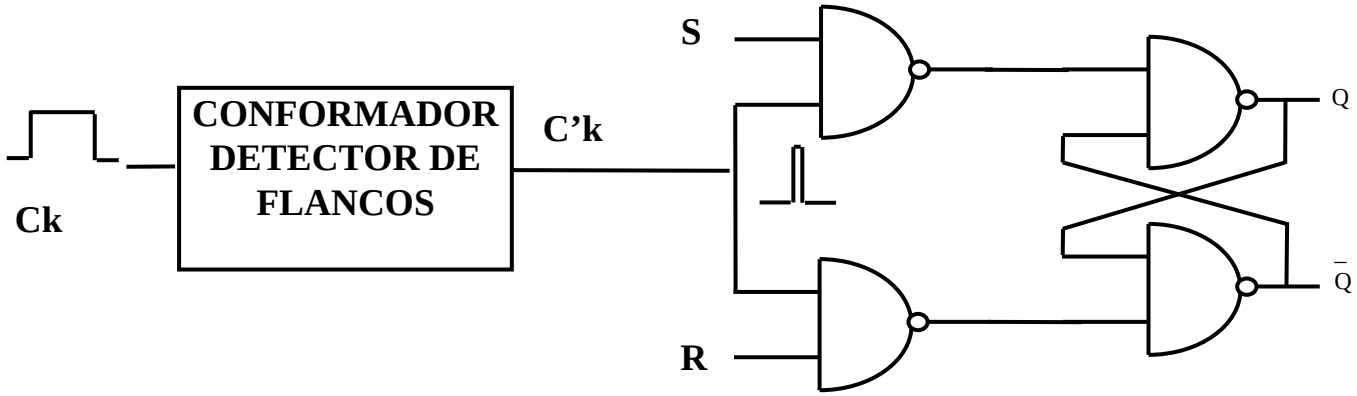
Flanco Negativo

Tiempos de establecimiento y retención (Setup, Hold Time)

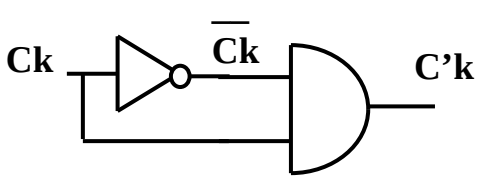


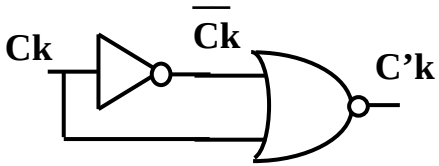
Establecimiento **Retencion**

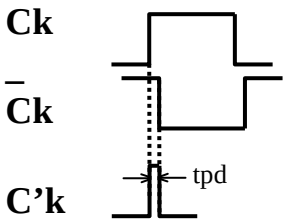
Biestable SR disparado por flanco (Edge Triggered)



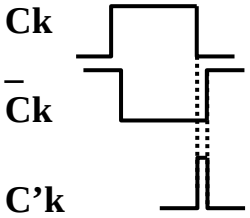
Circuito conformador/detector de flancos





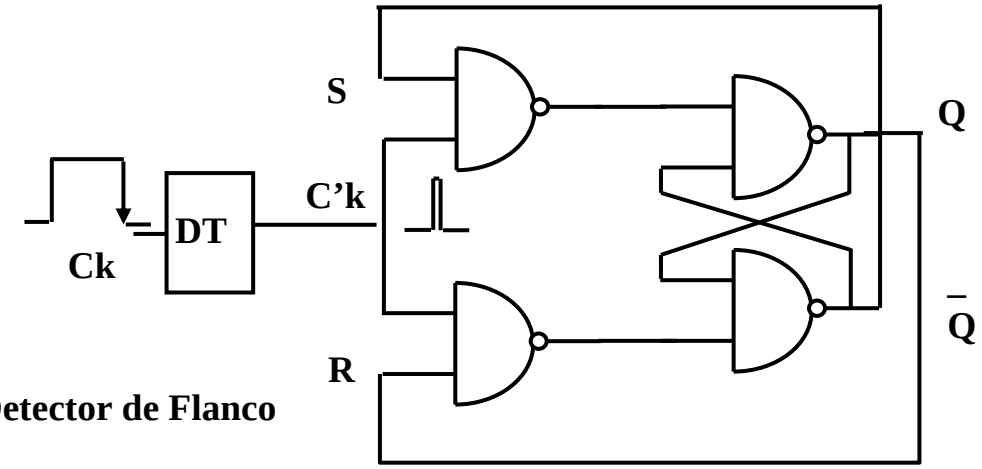


FLANCO POSITIVO



FLANCO NEGATIVO

Toggle Flip Flop



DT: Detector de Flanco

Salida Presente	Salida Futura
Qt	Qt+1
0	1
1	0

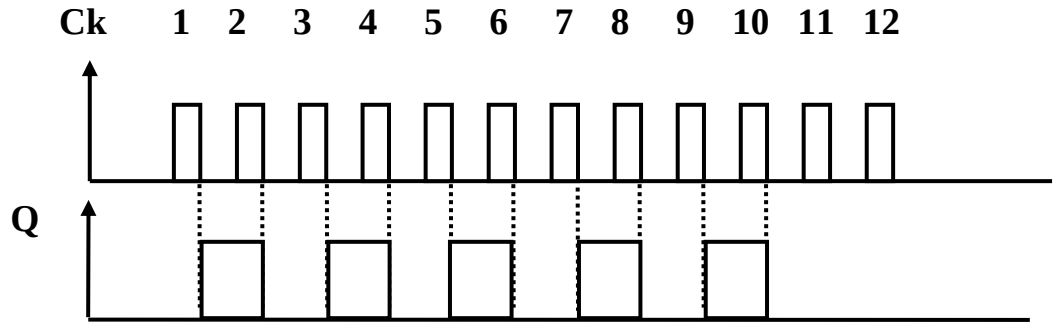
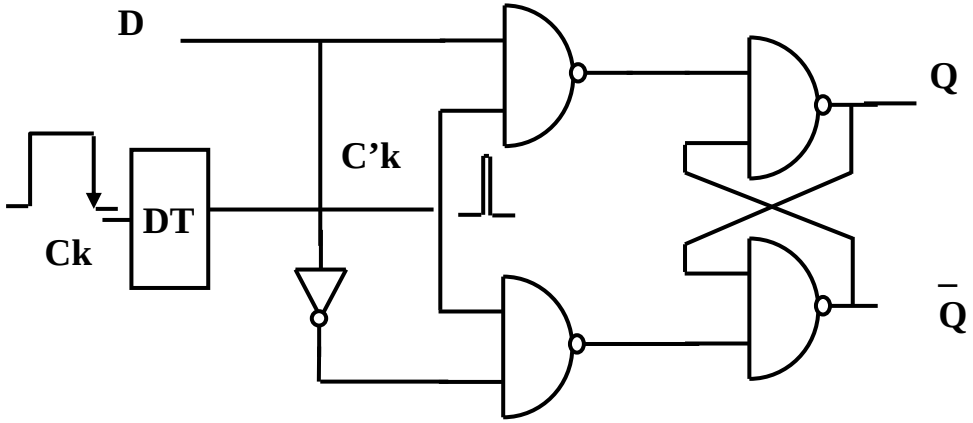


Diagrama temporal Toggle FF

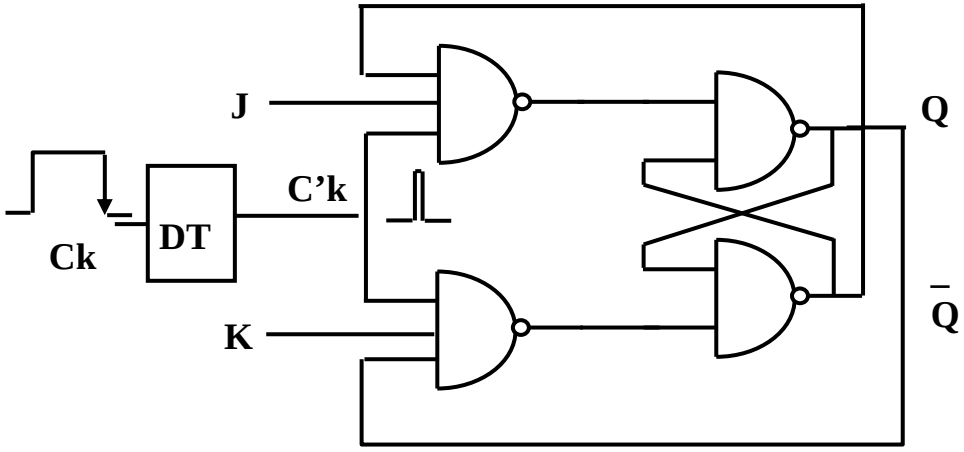
FLIP FLOP D y JK

FF D



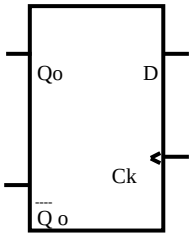
Entrada Presente	Salida Futura
D	Qt+1
0	0
1	1

FF JK

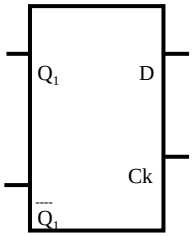


Entrada Presente		Salida Futura
J	K	Qt+1
0	0	Qt
0	1	0
1	0	1
1	1	Qt (negado)

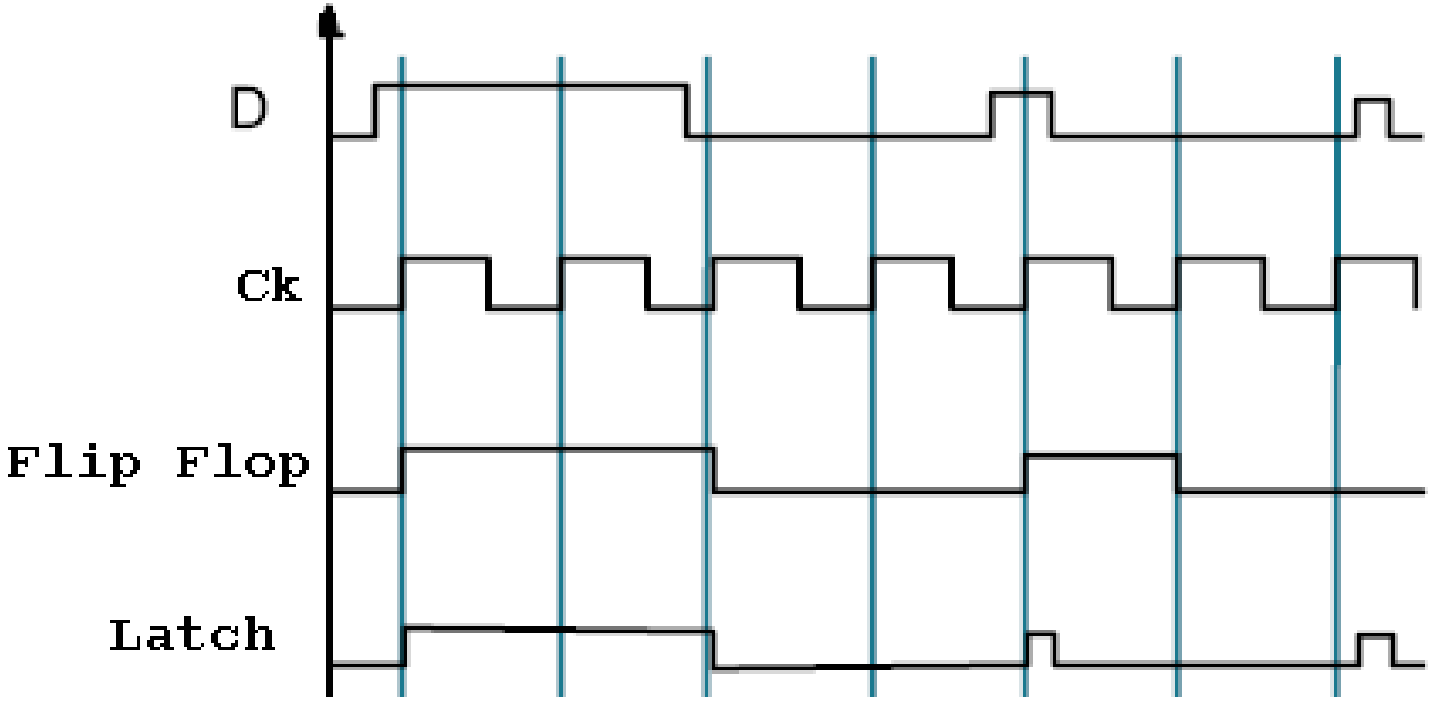
FF D vs LATCH D



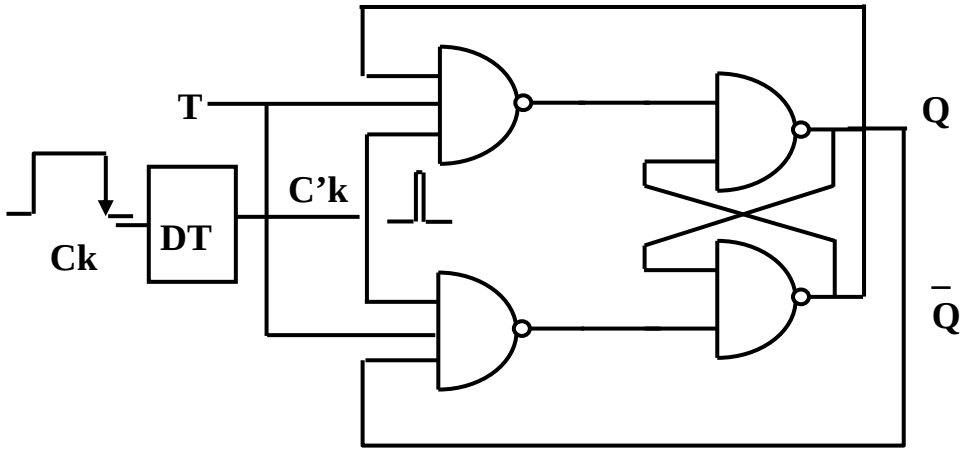
FF



LATCH

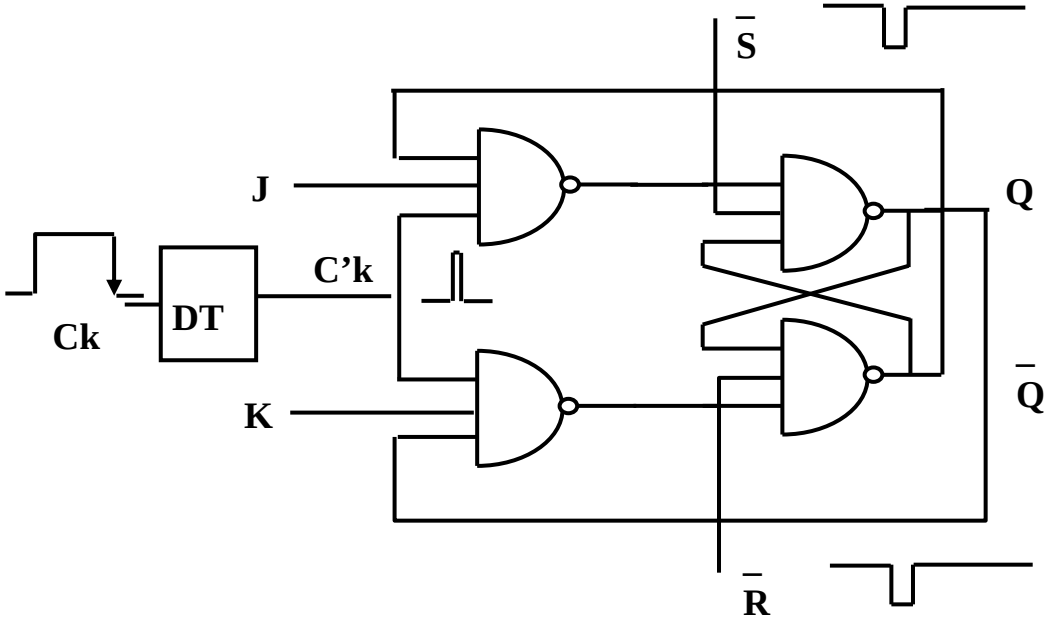


FF - T



Entrada Presente	Salida Futura
T	Qt+1
0	Qt
1	Qt (negado)

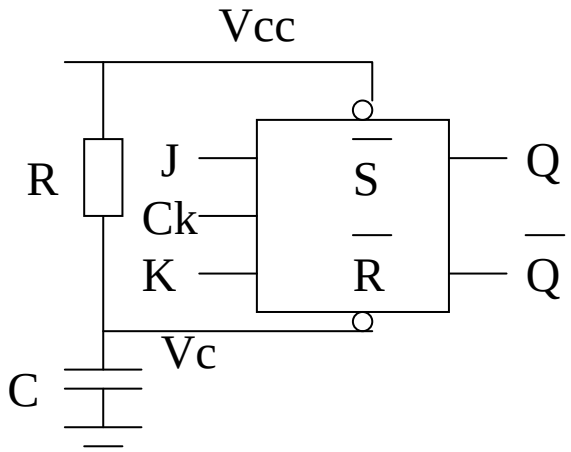
ENTRADAS ASINCRONAS

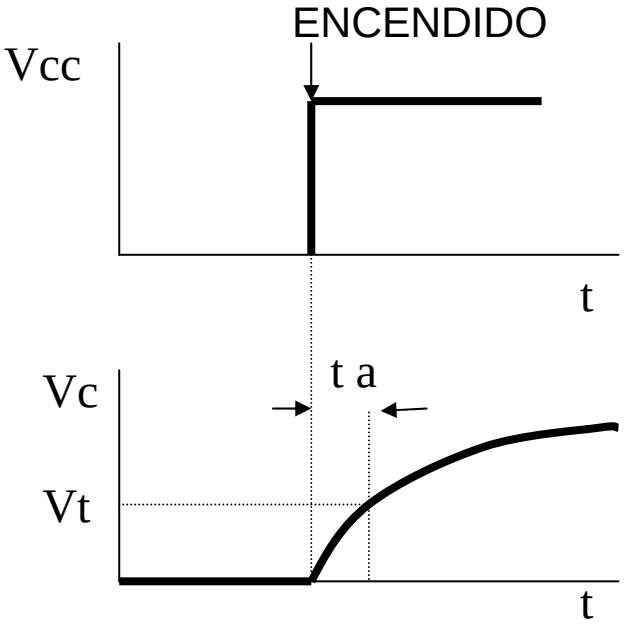


Entradas Asincronas		Entrada Presente		Salida Futura
S	R	J	K	Qt+1
1	1	0	0	Qt
1	1	0	1	0
1	1	1	0	1
1	1	1	1	Qt (negado)
0	1	x	x	1
1	0	x	x	0
0	0	x	x	No permitido

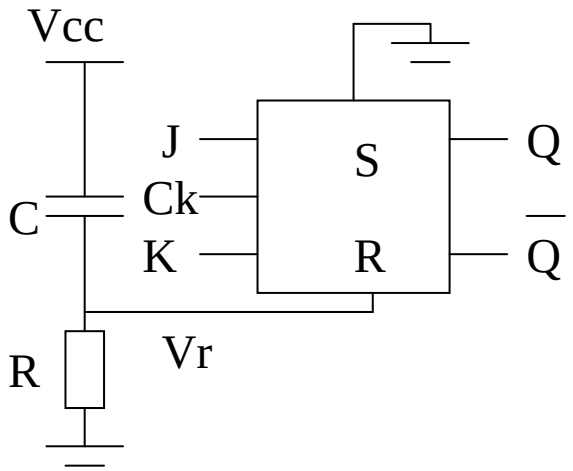
CONDICIONES INICIALES

ACTIVACION POR BAJO





ACTIVACION POR ALTO



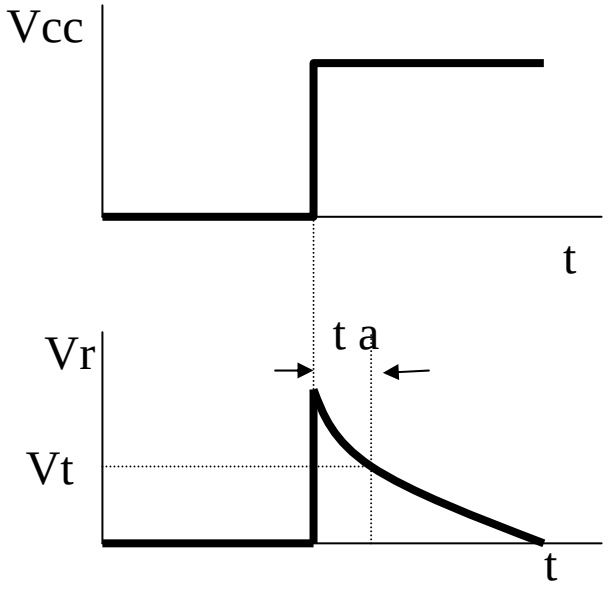
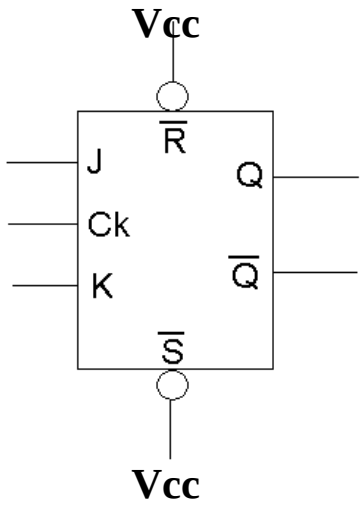


TABLA Y DIAGRAMA DE ESTADO FFJK



TV

Entrada Presente		Salida Futura
J	K	Qt+1
0	0	Qt
0	1	0
1	0	1
1	1	Qt (negado)

ECUACION CARACTERISTICA

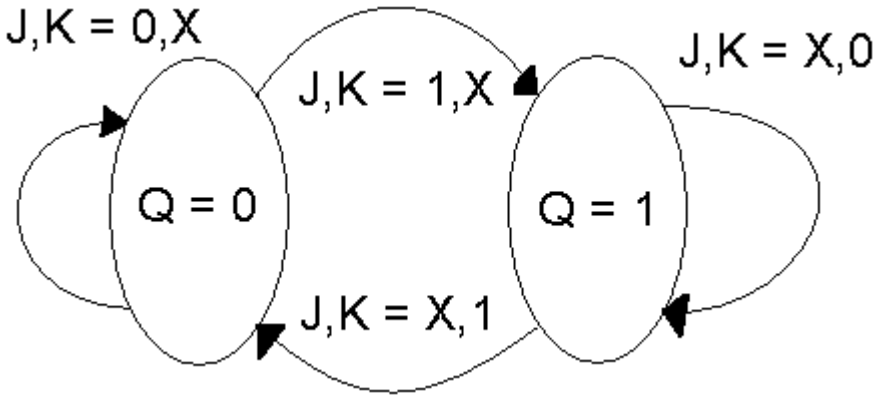
J	K	Q_t	Q_{t+1}
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

$$Q_{t+1} = J\bar{Q}_t + \bar{K}Q_t$$

TABLA DE ESTADOS

ESTADO PRESENTE Qt	ESTADO FUTURO Qt + 1	ENTRADAS	
		J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

DIAGRAMA DE ESTADO O DIAGRAMA DE TRANSICIONES



CODIGO VHDL – LATCH D

```
LIBRARY ieee;
USE ieee.std_logic_1164.all ;
ENTITY latch IS
      PORT ( D, Clk    : IN   STD_LOGIC ;
                   Q          : OUT STD_LOGIC ) ;
END latch ;
ARCHITECTURE Behavior OF latch IS
BEGIN
       PROCESS (D, Clk)
       BEGIN
            IF Clk = ´1´ THEN
                 Q <=  D ;
            END IF ,
       END PROCESS ;
END Behavior ;
```

Si Clk=0, no se especifica el valor de Q, y Q retiene el valor actual

Código para un latch D

CODIGO VHDL FF D

```
LIBRARY ieee;
USE ieee.std_logic_1164.all ;
ENTITY flipflop IS
        PORT ( D, Clock    : IN   STD_LOGIC ;
                     Q             : OUT STD_LOGIC ) ;
END flipflop ;

ARCHITECTURE Behavior OF flipflop IS
BEGIN
    PROCESS ( Clock)
    BEGIN
        IF Clock‘EVENT AND Clock = ´1´ THEN  (flanco positivo)
            Q <=  D ;
        END IF ,
    END PROCESS ;
END Behavior ;
```

CODIGO VHDL FF D (alternativo)

```
LIBRARY ieee;
USE ieee.std_logic_1164.all ;
ENTITY flipflop IS
        PORT ( D, Clock    : IN   STD_LOGIC ;
                      Q             : OUT STD_LOGIC ) ;
END flipflop ;


ARCHITECTURE Behavior OF flipflop IS
BEGIN
    PROCESS
    BEGIN
        WAIT UNTIL Clock‘EVENT AND Clock = ´1´ ;
            Q <=  D ;
     END PROCESS ;
END Behavior ;
```

CODIGO VHDL FFD CON CLEAR ASINCRONICO

Se describe un FFD con una entrada reset (clear) asíncrono que se activa por nivel bajo. Cuando *Resetn*, la entrada reset es igual a 0 y la salida del FF va a 0 (Q=0)

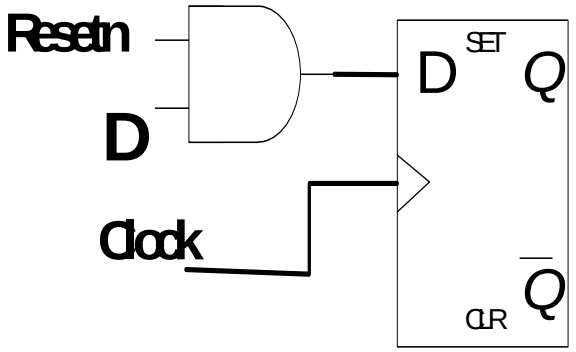
```
LIBRARY ieee;
USE ieee.std_logic_1164.all ;
ENTITY flipflop IS
        PORT ( D, Resetn, Clock    : IN   STD_LOGIC ;
                    Q                      : OUT STD_LOGIC ) ;
END flipflop ;

ARCHITECTURE Behavior OF flipflop IS
BEGIN
    PROCESS ( Resetn, Clock)
    BEGIN
        IF Resetn = ´0´ THEN
            Q <=  ´0´ ;
        ELSIF Clock`EVENT AND Clock = ´1´ THEN
            Q <= D ;
        END IF ;
    END PROCESS ;
END Behavior ;
```

CODIGO VHDL FF D CLEAR SINCRONICO

En este caso la señal de reset (clear) solamente actua cuando llega un flanco positivo de Clock. Este código VHDL genera el circuito que se muestra a continuación

```
LIBRARY ieee;
USE ieee.std_logic_1164.all ;
ENTITY flipflop IS
PORT ( D, Resetn, Clock    : IN   STD_LOGIC ;
            Q                        : OUT STD_LOGIC ) ;
END flipflop ;
```

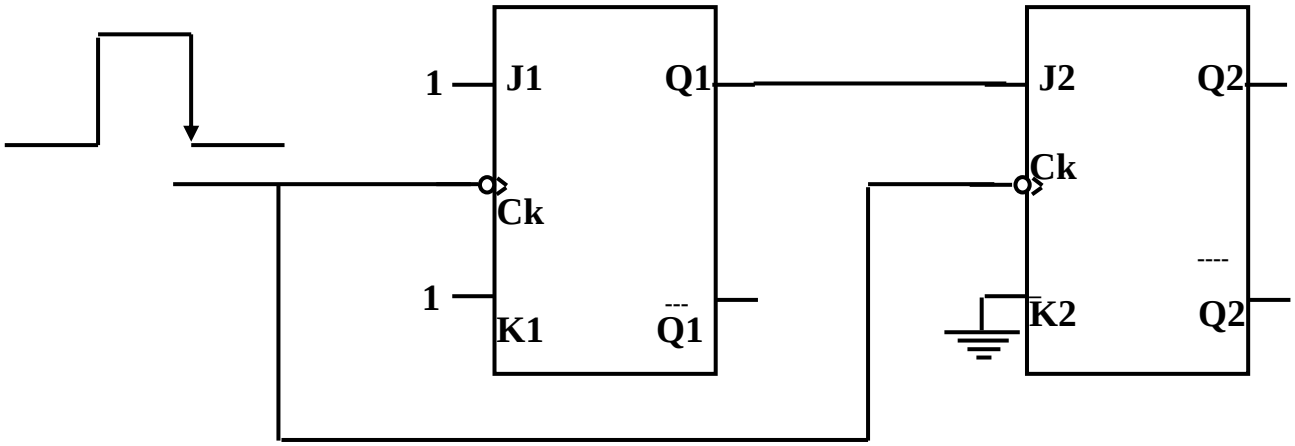
```
ARCHITECTURE Behavior OF flipflop IS
BEGIN
      PROCESS ( Resetn, Clock)
      BEGIN
            WAIT UNTIL Clock = ´1´ ;
             IF Resetn = ´0´ THEN
                  Q <=  ´0´ ;
             ELSE
                  Q <= D ;
             END IF ;
      END PROCESS ;
END Behavior ;
```



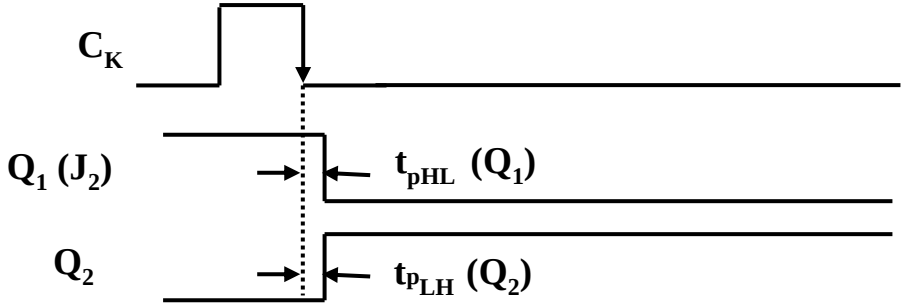
```
LIBRARY ieee ;
USE ieee.std_logic_1164.all ;

ENTITY muxdff IS
    PORT (  D0, D1, Sel, Clock : IN     STD_LOGIC ;
            Q                  : OUT    STD_LOGIC ) ;
END muxdff ;

ARCHITECTURE Behavior OF muxdff IS
BEGIN
    PROCESS
    BEGIN
        WAIT UNTIL Clock'EVENT AND Clock = '1' ;
        IF Sel = '0' THEN
            Q <= D0 ;
        ELSE
            Q <= D1 ;
        END IF ;
    END PROCESS ;
END Behavior ;
```

Figure 7.48 Code for a D flip-flop with a 2-to-1 multiplexer on the *D* input

FLIP`FLOP EN CASCADA



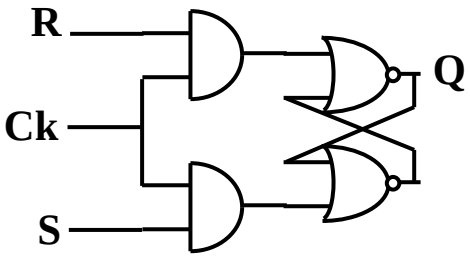
Condiciones Iniciales: CI: Q1 = 1 ; Q2 = 0
El funcionamiento correcto es el indicado en la siguiente figura

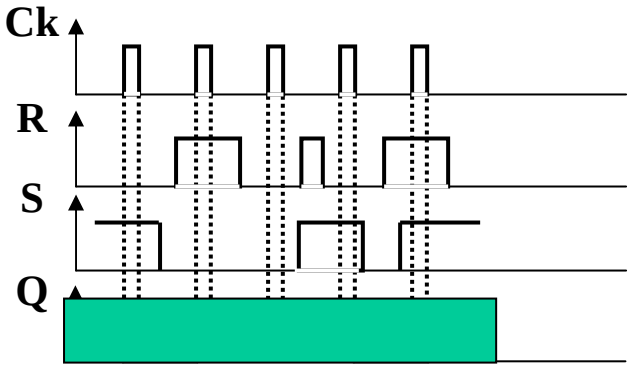


Para que ello ocurra tpHL debe ser mayor a th, tiempo de retención, (tpHL > th) de Q2, cosa que ciertamente se cumple, si ello no ocurriera la respuesta de Q2 seria incierta. Los FFs actuales tienen un th < 5 nseg., tendiendo a 0 nseg

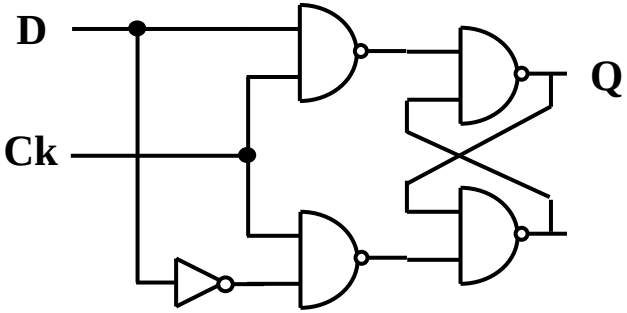
EJERCICIOS - 1

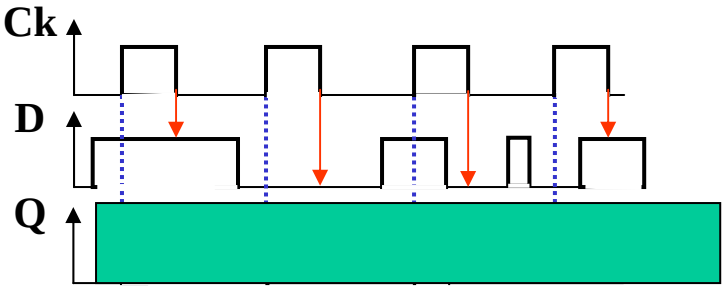
1-REALIZAR EL DIAGRAMA TEMPORAL DEL CIRCUITO DE LA FIGURA





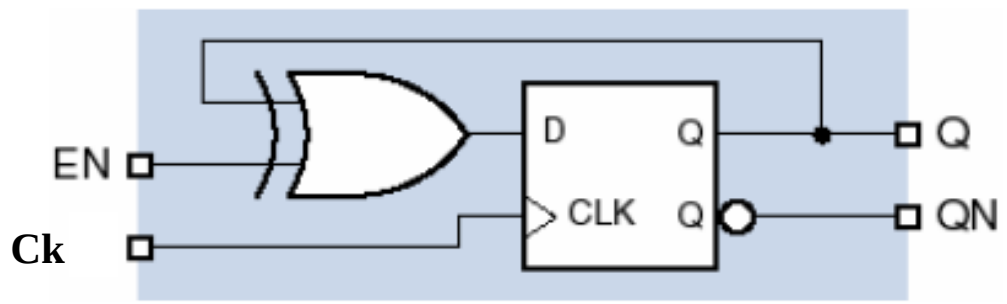
2-REALIZAR EL DIAGRAMA TEMPORAL DEL CIRCUITO DE LA FIGURA

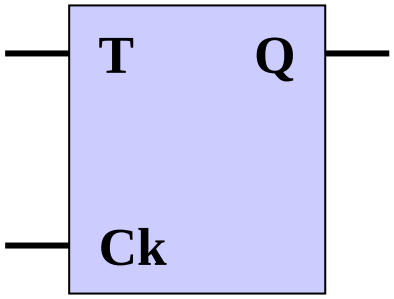


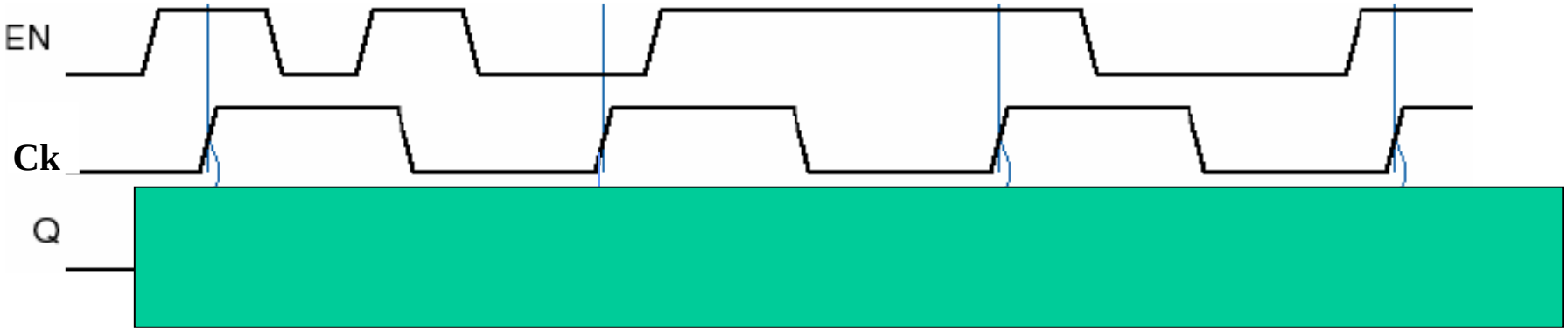


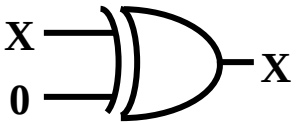
EJERCICIOS – 2

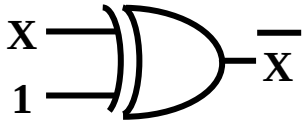
3-REALIZAR EL DIAGRAMA TEMPORAL DEL CIRCUITO DE LA FIGURA





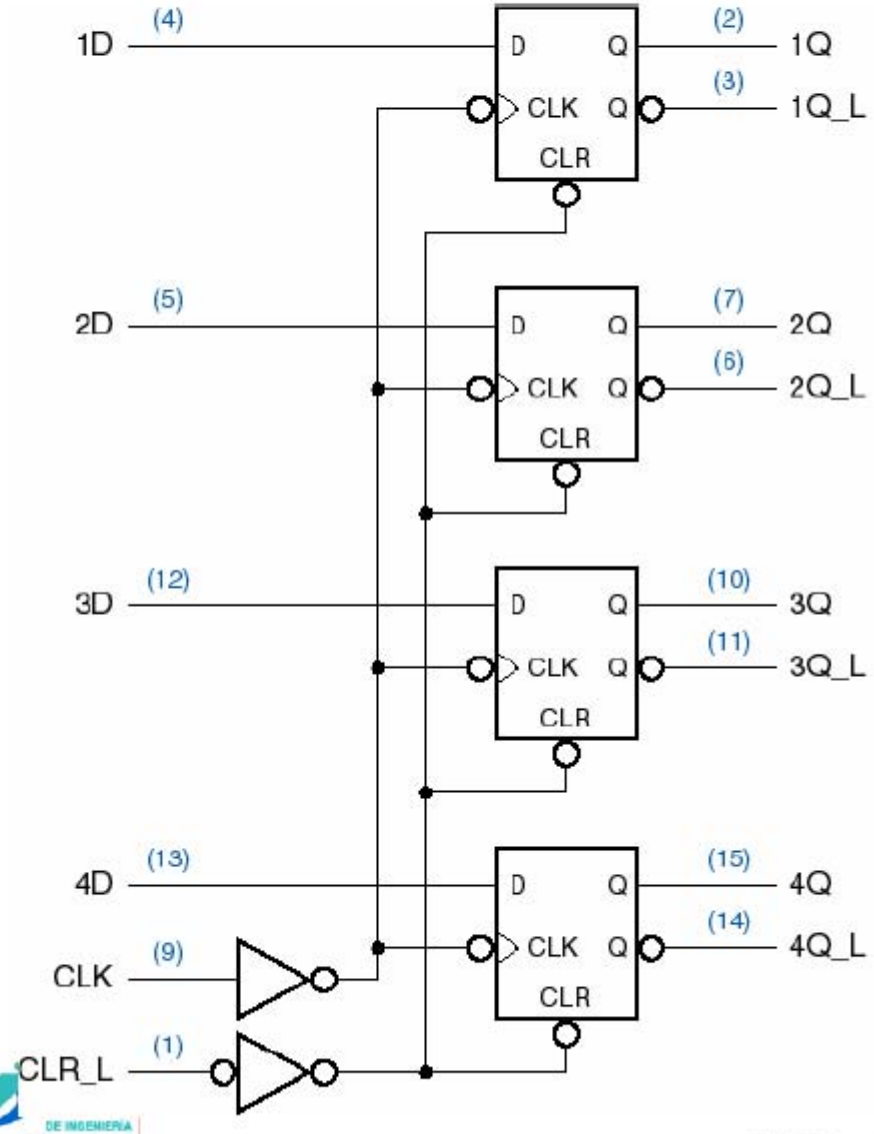




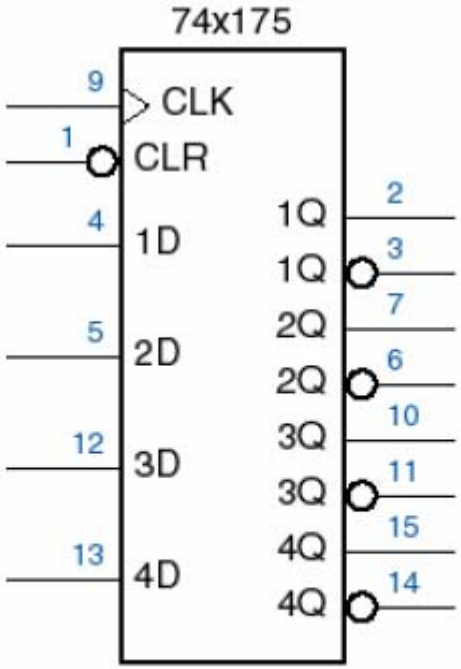


COMO PUEDE VERSE LA SALIDA Q RESPONDE A UN FF T

Registros y “latches” de varios bits



- 74x175
- Dispone de una señal de “clear” asíncrono CLR_L



Registro de 8 bits (octal)

- 74x374
- Salida triestado controlada por OE_L