

86.01 Técnica Digital

Circuitos Secuenciales Biestables

Ing. Jorge H. Fuchs

Objetivos de la clase:

Analizar los efectos de la realimentación en circuitos lógicos.

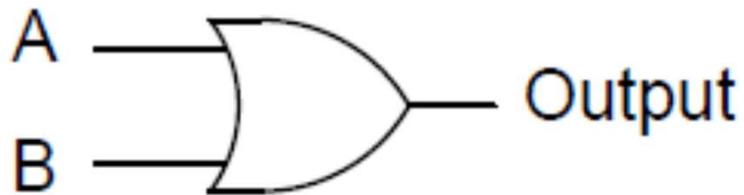
Estudiar las características de los circuitos secuenciales y su empleo como circuitos de memoria.

Conocer y aplicar los distintos tipos de circuitos biestables como así también sus características.

Aplicar técnicas para obtener un tipo de biestable a partir de otro.

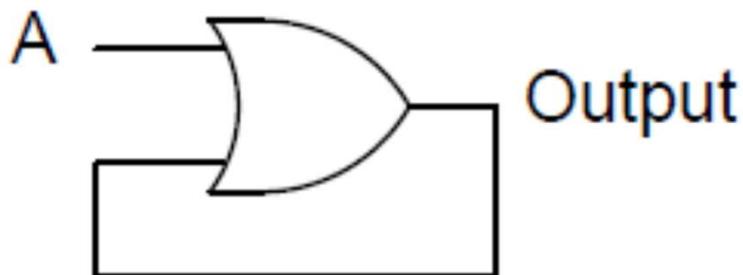
Realimentación en circuitos lógicos

Los circuitos vistos hasta ahora son circuitos **combinacionales**, ya que su salida responde a la **combinación** de sus **entradas**. Para cada una de las 2^n combinaciones, la salida queda unívocamente establecida mediante su TV. Como ejemplo una compuerta OR:



A	B	Output
0	0	0
0	1	1
1	0	1
1	1	1

Pero qué ocurre cuando realimentamos su salida:



A	Output
0	?
1	1

Circuitos secuenciales

La realimentación me permite obtener circuitos **secuenciales**, en los que la salida ya no responde solo a la combinación de sus **entradas**, sino que también depende del estado en que se encuentra su **salida**.

Combinacionales: $Z = f(A, B, C)$

Secuenciales: $Z^{n+1} = f(A^n, B^n, C^n, Z^n)$

El próximo estado depende también del estado actual.

El circuito recuerda en qué estado se encuentra, tiene memoria.

Un biestable es una celda básica de memoria RAM estática.

Circuitos secuenciales biestables

Los circuitos secuenciales más simples son los biestables, poseen solo **2 estados estables** (0 o 1). Es una celda básica SRAM. Podemos clasificarlos en principio:

Por su funcionamiento (TV):

SR
JK
D
T

Por su sincronismo:

Asincrónico

Latch / FF asincrónico?

Sincrónico de nivel (alto o bajo)

Latch c/habilit. / FF de nivel??

Sincrónico de flanco (asc. o desc.)

Flip Flop / FF de flanco??

Circuitos secuenciales biestables

Para los biestables más comunes analizaremos sus **características**:

TV reducida (o implícita)

TV (ampliada)

Ecuación característica

Circuito lógico (interno)

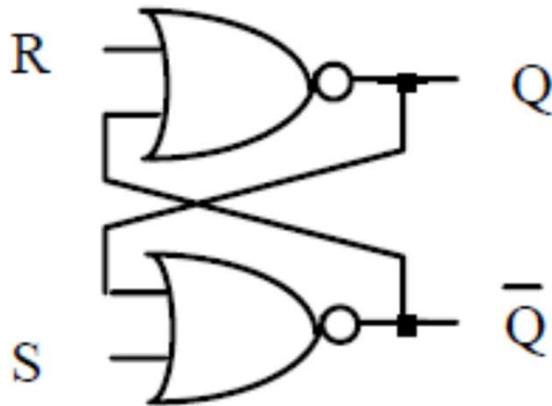
Símbolo gráfico

Tabla de transiciones

Diagrama de estados

Biastable S-R (Set-Reset)

Circuito lógico



TV reducida

S	R	Q _{n+1}
0	0	Q _n
0	1	0
1	0	1
1	1	x

$$S \cdot R = 0$$

TV ampliada

S	R	Q _n	Q _{n+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	x
1	1	1	x

Ecuación característica:

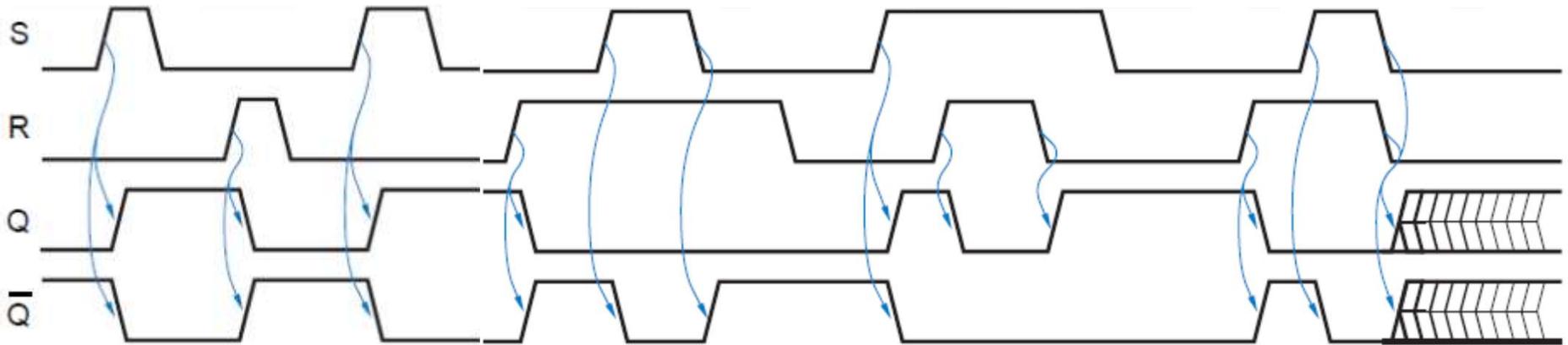
$$Q^{n+1} = f(S, R, Q^n)$$

$$Q^{n+1} = S + \bar{R} Q^n$$

S \ R Q _n	00	01	11	10
0	0	1	0	0
1	1	1	x	x

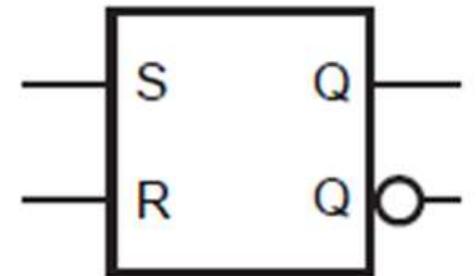
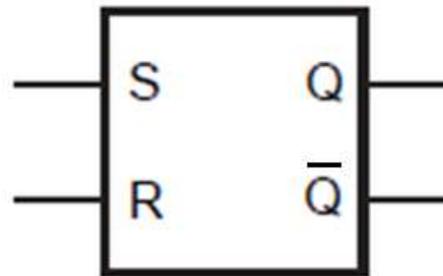
Biastable S-R (Set-Reset)

Diagrama temporal: funcionamiento normal y funcionamiento con $R = S = 1$



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

Símbolo gráfico



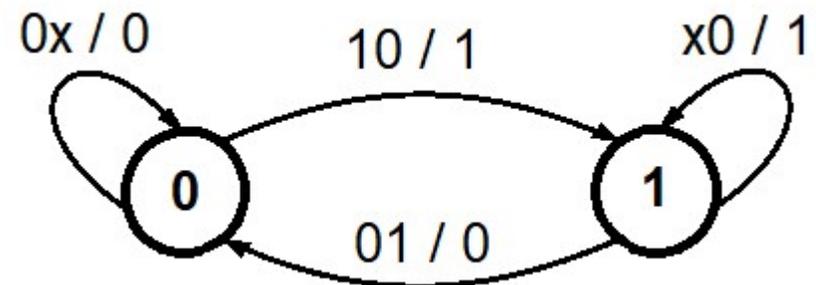
Biestable S-R (Set-Reset)

Tabla de transiciones

S	R	Q _n	Q _{n+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	x
1	1	1	x

Q _n	Q _{n+1}	S	R
0	0	0	x
0	1	1	0
1	0	0	1
1	1	x	0

Diagrama de estados

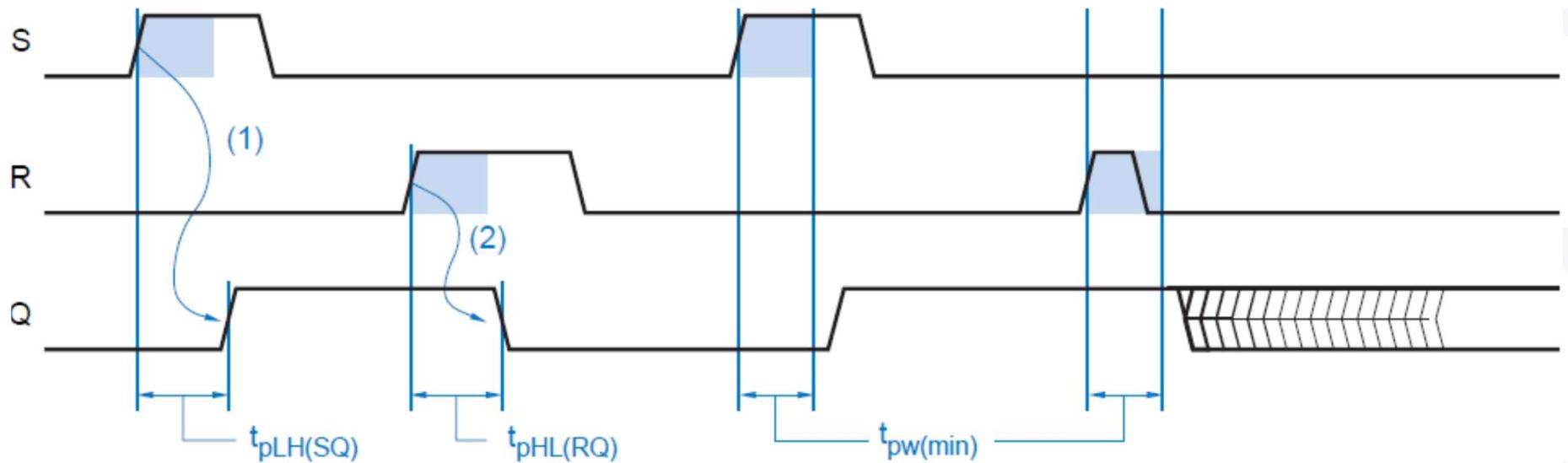


Biastable S-R (Set-Reset)

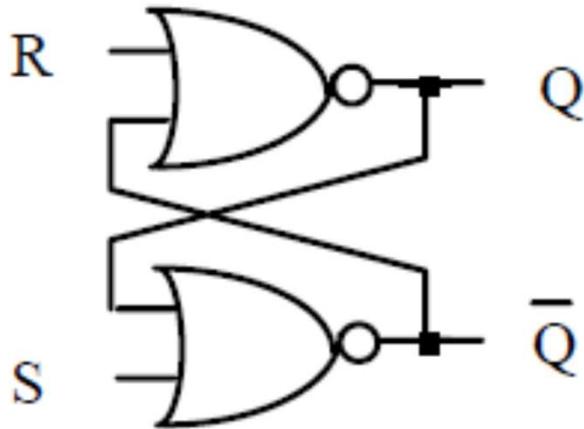
Tiempos:

Retardos de propagación: t_{pLH} y t_{pHL}

Ancho de pulso mínimo: $t_{pw(min)}$

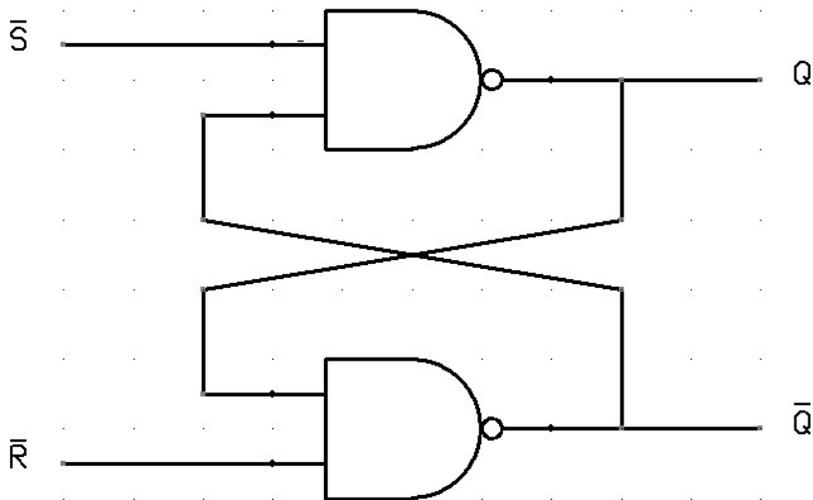


Biastable S-R con NAND



S	R	Q _{n+1}
0	0	Q _n
0	1	0
1	0	1
1	1	x

Implementación con NAND:



notS	notR	Q _{n+1}
0	0	x
0	1	1
1	0	0
1	1	Q _n

Biestables con habilitación por nivel (Ck)

TV reducida

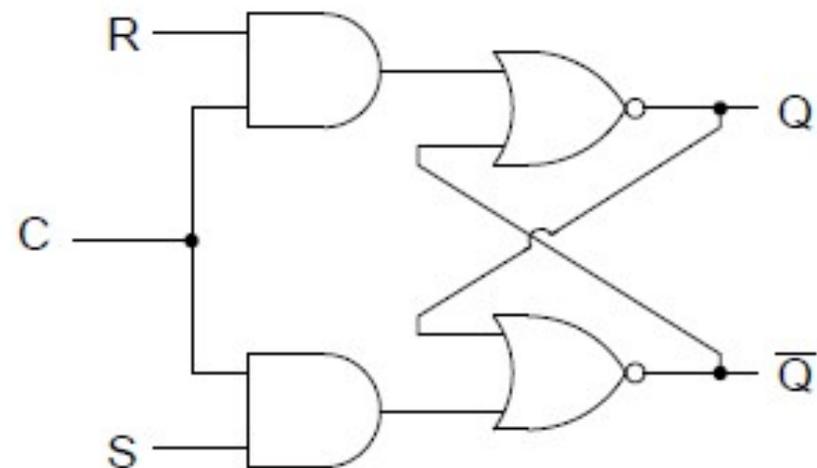
C	S	R	Q _{n+1}
1	0	0	Q _n
1	0	1	0
1	1	0	1
1	1	1	x
0	x	x	Q _n

TV ampliada

C	S	R	Q _n	Q _{n+1}
1	0	0	0	0
1	0	0	1	1
1	0	1	0	0
1	0	1	1	0
1	1	0	0	1
1	1	0	1	1
1	1	1	0	x
1	1	1	1	x
0	x	x	0	0
0	x	x	1	1

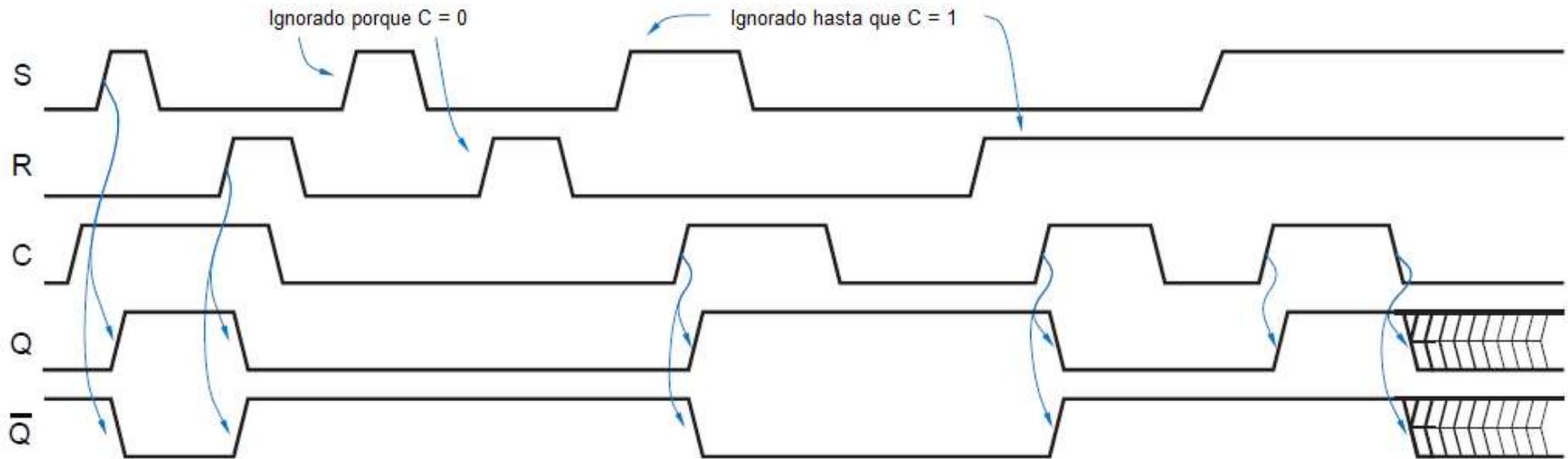
C es la temporización, reloj o clock (Ck)

Circuito lógico



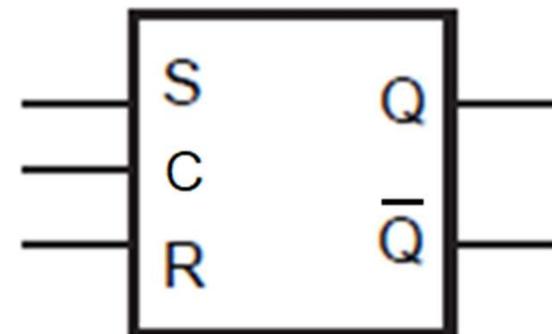
Biestables con habilitación por nivel (Ck)

Diagrama temporal: funcionamiento normal y funcionamiento con R = S = 1



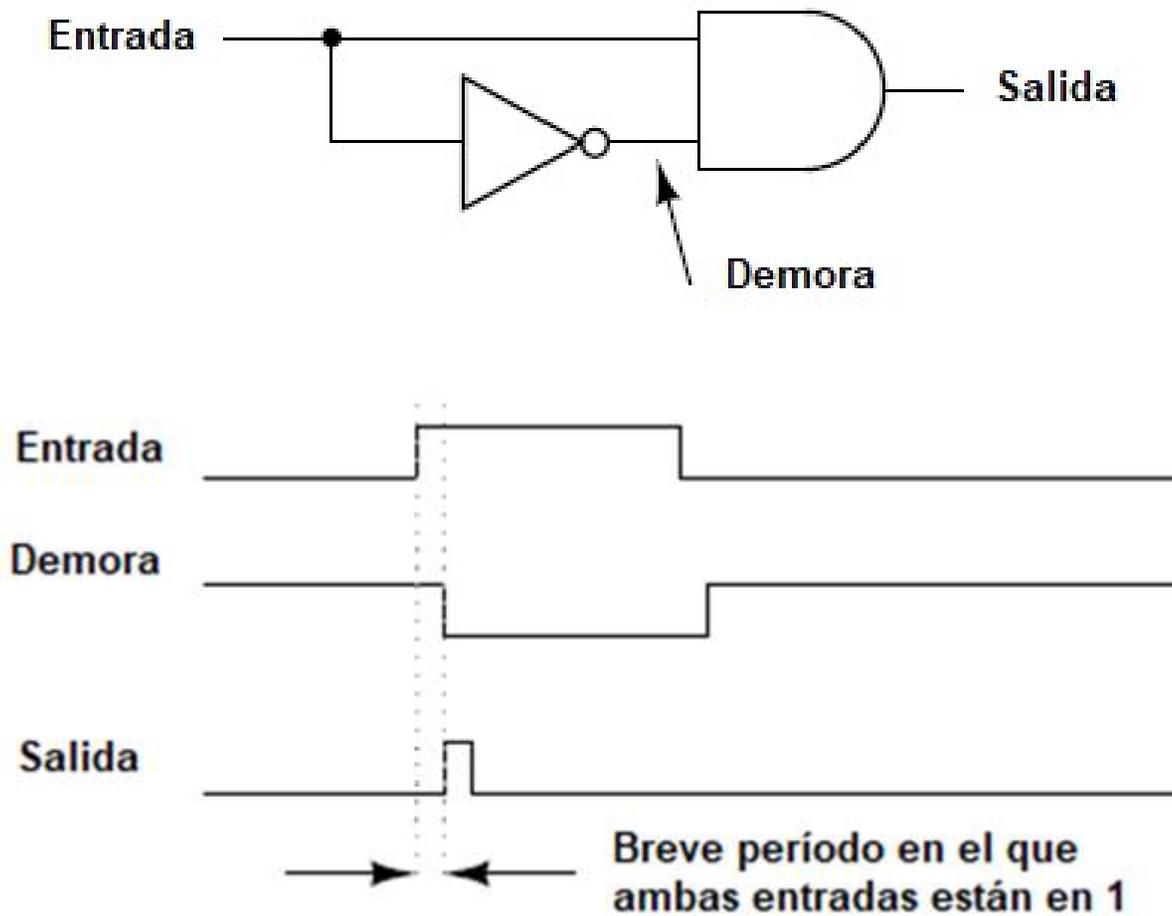
C	S	R	Q _{n+1}
1	0	0	Q _n
1	0	1	0
1	1	0	1
1	1	1	x
0	x	x	Q _n

Símbolo gráfico



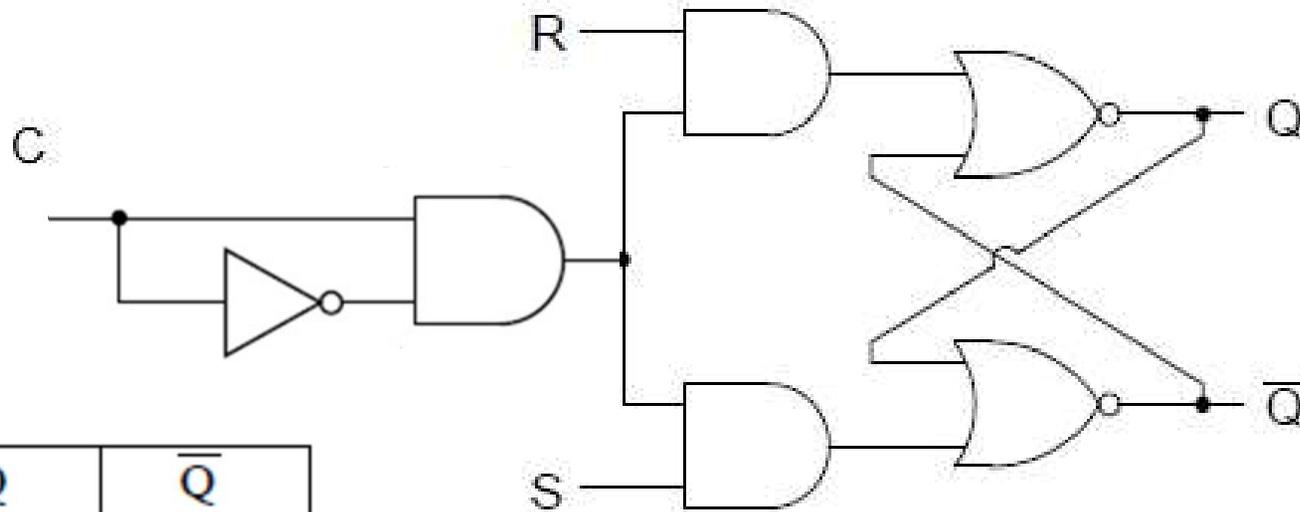
Biestables con habilitación por flanco (Ck)

Para que se produzca **un solo cambio** por cada pulso de Ck.



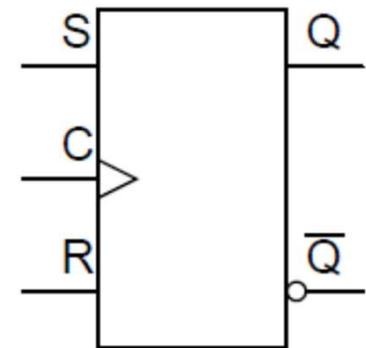
Biestables con habilitación por flanco (Ck)

Vemos el circuito y la TV de un SR de flanco ascendente.



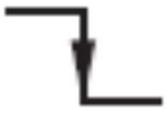
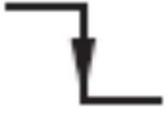
C	S	R	Q	\bar{Q}
┌	0	0	latch	latch
┌	0	1	0	1
┌	1	0	1	0
┌	1	1	0	0
x	0	0	latch	latch
x	0	1	latch	latch
x	1	0	latch	latch
x	1	1	latch	latch

Símbolo gráfico

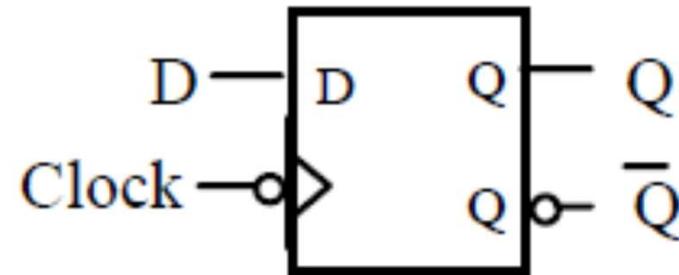


Biestables con habilitación por flanco (Ck)

También podemos obtener biestables de **flanco descendente**.

D	CLK_L	Q
0		0
1		1
x	0	last Q
x	1	last Q

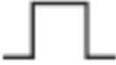
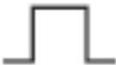
Símbolo gráfico

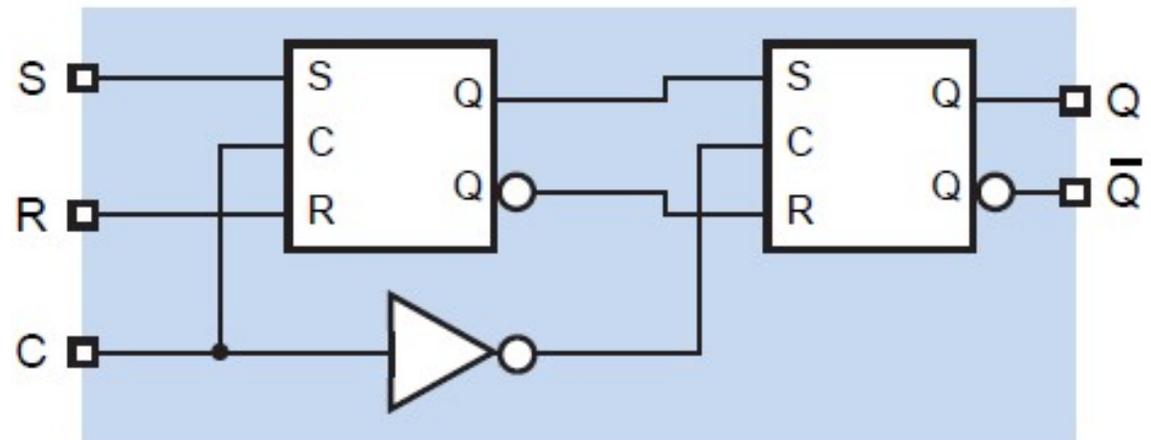


Biestables Maestro Esclavo

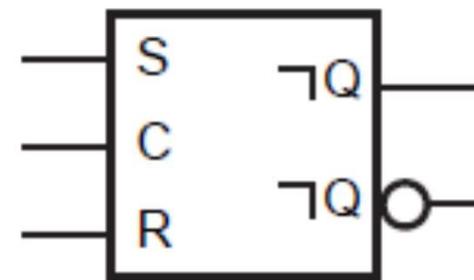
Necesitan que se produzcan 2 cambios en el nivel del Ck.

Con $Ck = 1$, el dato pasa a la salida del Master, y con $Ck = 0$, el dato pasa a la salida del Slave, quedando el Master deshabilitado.

S	R	C	Q
x	x	0	last Q
0	0		last Q
0	1		0
1	0		1
1	1		undef.

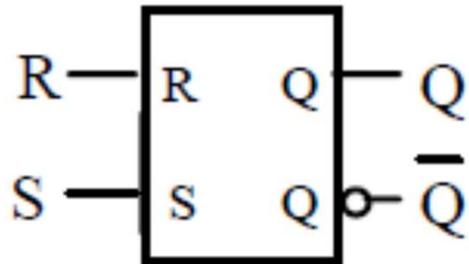


Símbolo gráfico

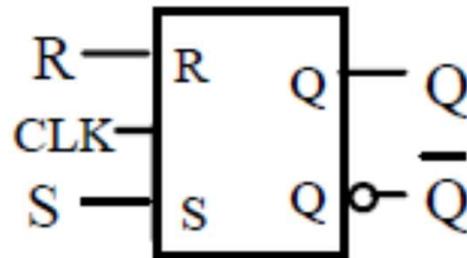


Resumen de sincronismos de biestables

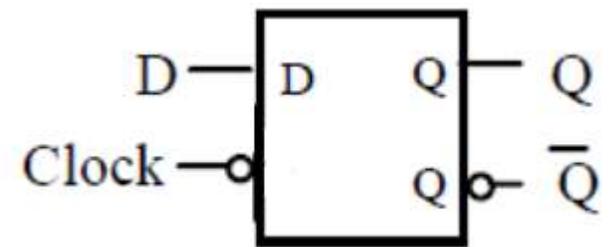
Asincrónico



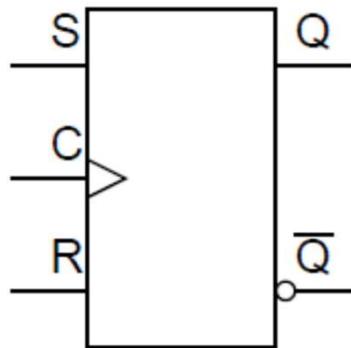
Nivel alto



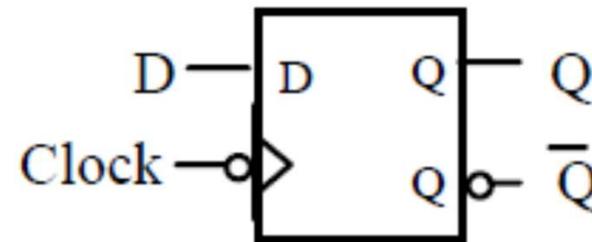
Nivel bajo



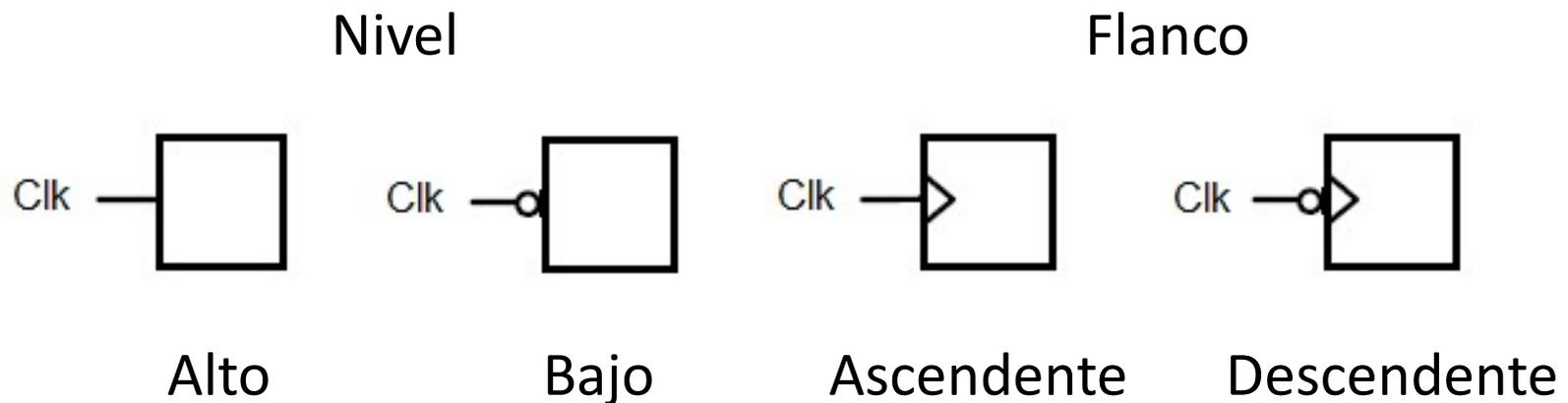
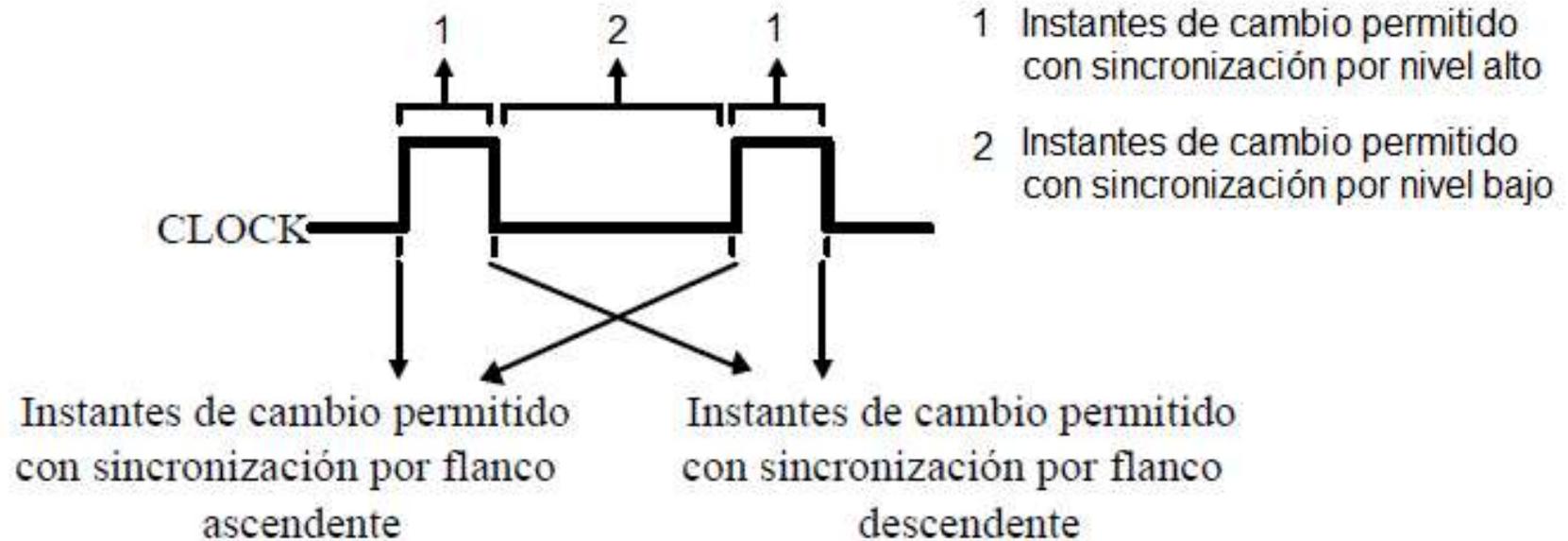
Flanco ascendente



Flanco descendente

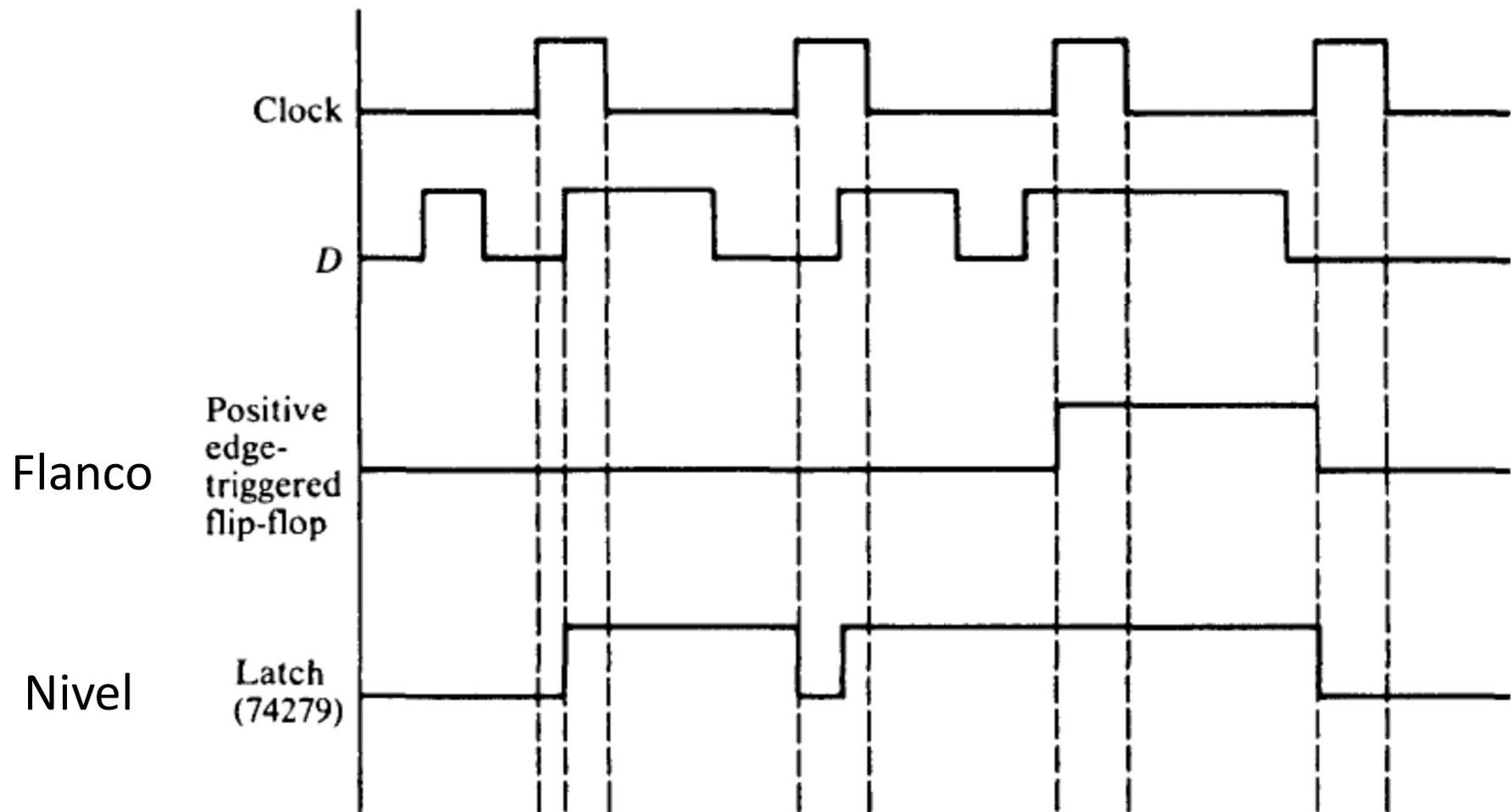


Resumen de sincronismos de biestables



Resumen de sincronismos de biestables

Respuesta de un biestable de **flanco ascendente** y de uno de **nivel alto**.

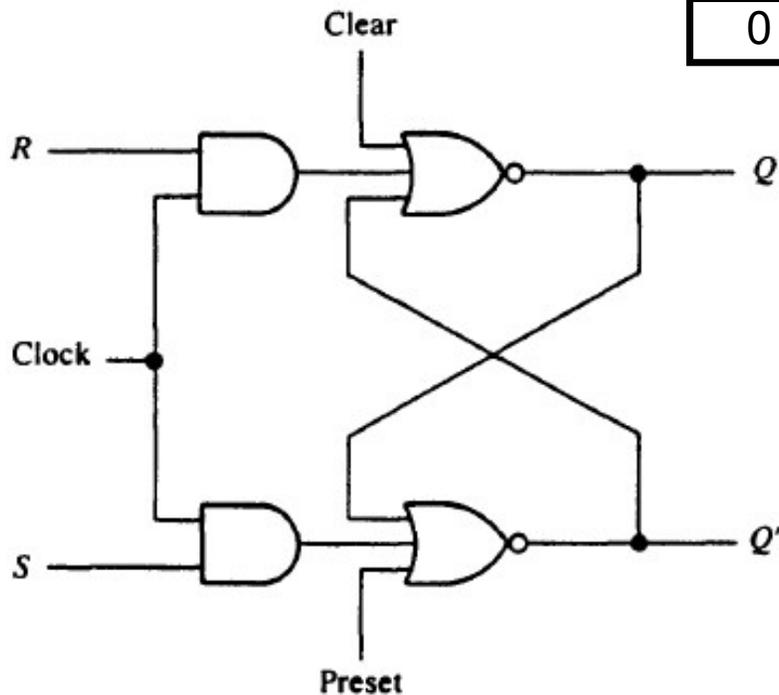


Biestables con Entradas Asincrónicas

TV reducida:

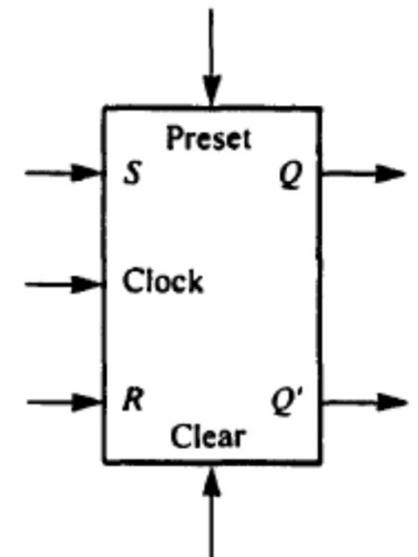
Pr y Clr son **entradas asincrónicas**, no dependen de la temporización o clock (Ck).

Pr	Clr	Ck	S	R	Q _{n+1}	not Q _{n+1}
0	0	1	0	0	Q _n	not Q _n
0	0	1	0	1	0	1
0	0	1	1	0	1	0
0	0	1	1	1	0	0
0	1	x	x	x	0	1
1	0	x	x	x	1	0
1	1	x	x	x	0	0
0	0	0	x	x	Q _n	not Q _n



Circuito lógico

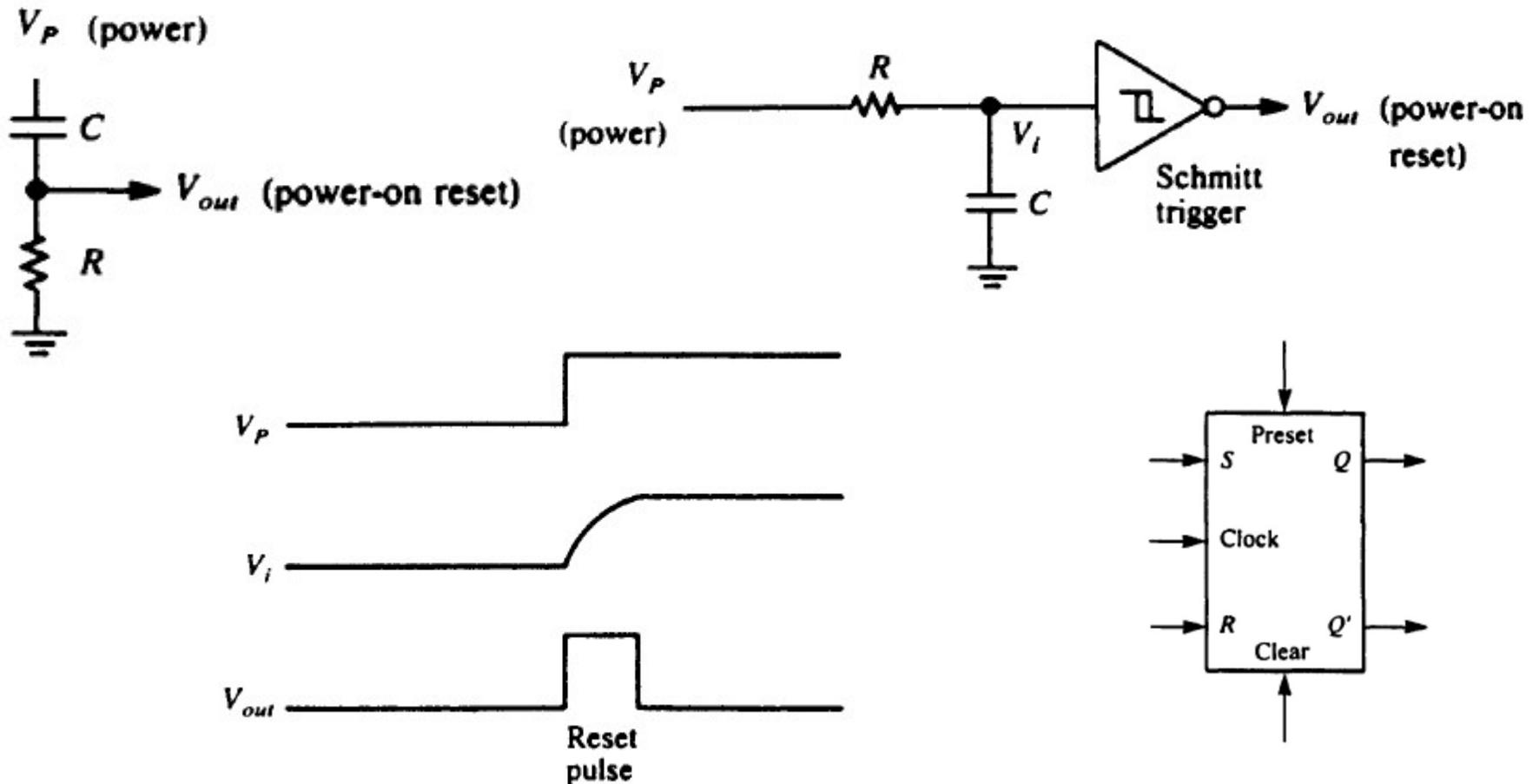
Símbolo gráfico



Power On Reset (POR)

Para forzar un biestable a un determinado estado inicial utilizo las entradas asincrónicas (Pr o Clr según sea 1 o 0 el estado inicial). El tiempo debe ser suficiente para que se establezca pero no demasiado largo.

Muchas veces se utilizan circuitos de **Power On Reset (POR)** para lograrlo.



Biestable J-K

TV reducida SR

S	R	Q _{n+1}
0	0	Q _n
0	1	0
1	0	1
1	1	x

TV reducida JK

J	K	Q _{n+1}
0	0	Q _n
0	1	0
1	0	1
1	1	not Q _n

TV ampliada

J	K	Q _n	Q _{n+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

Equivalencias: $J = S$ y $K = R$
 Invierte estado con $J = K = 1$

Ecuación característica:

$$Q^{n+1} = f(J, K, Q^n)$$

$$Q^{n+1} = J \overline{Q}^n + \overline{K} Q^n$$

$J \backslash K Q^n$	00	01	11	10
0	0	1	0	0
1	1	1	0	1

Note: In the original image, a green circle highlights the '1' in the cell (0, 01) and blue L-shaped brackets highlight the '1's in the cells (1, 00) and (1, 10).

Biestable J-K

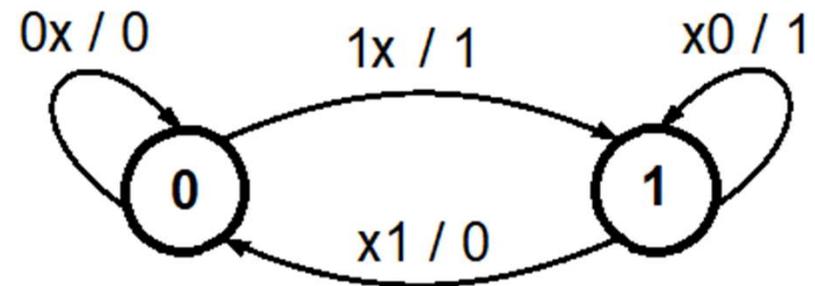
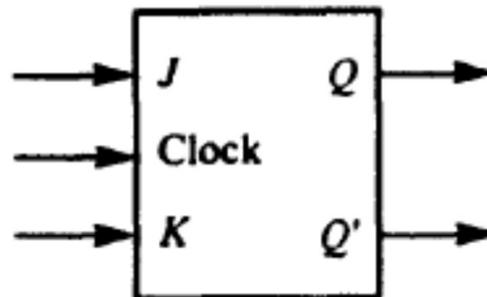
Tabla de transiciones

J	K	Q _n	Q _{n+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 _n	Q _{n+1}	J	K
0	0	0	x
0	1	1	x
1	0	x	1
1	1	x	0

Diagrama de estados

Símbolo gráfico



Biestable D (Delay)

Posee una sola entrada.

Guarda el dato.

TV reducida

D	Q ⁿ⁺¹
0	0
1	1

TV ampliada

D	Q ⁿ	Q ⁿ⁺¹
0	0	0
0	1	0
1	0	1
1	1	1

Ecuación característica:

$$Q^{n+1} = f(D, Q^n)$$

$$Q^{n+1} = D$$

No depende de Qⁿ

	Q ⁿ	0	1
D	0	0	0
	1	1	1

Biestable D (Delay)

Tabla de transiciones

D	Q _n	Q _{n+1}
0	0	0
0	1	0
1	0	1
1	1	1

Q _n	Q _{n+1}	D
0	0	0
0	1	1
1	0	0
1	1	1

Símbolo gráfico

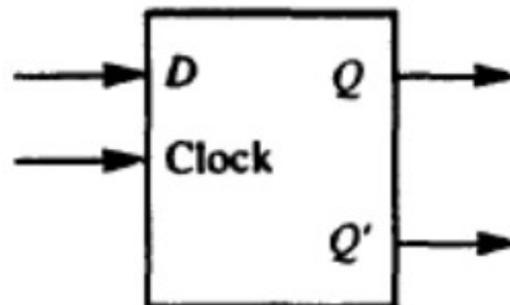
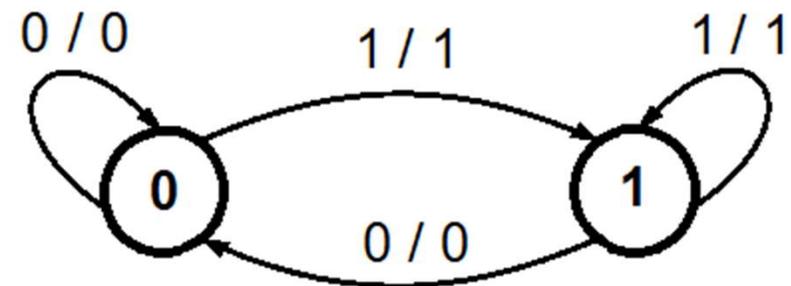


Diagrama de estados



Biastable T (Toggle)

Posee una sola entrada.

Con $T = 1$ invierte la salida.

TV reducida

T	Q^{n+1}
0	Q^n
1	not Q^n

TV ampliada

T	Q^n	Q^{n+1}
0	0	0
0	1	1
1	0	1
1	1	0

Ecuación característica:

$$Q^{n+1} = f(T, Q^n)$$

$$Q^{n+1} = T \overline{Q^n} + \overline{T} Q^n$$

	Q^n	0	1
T	0	0	1
	1	1	0

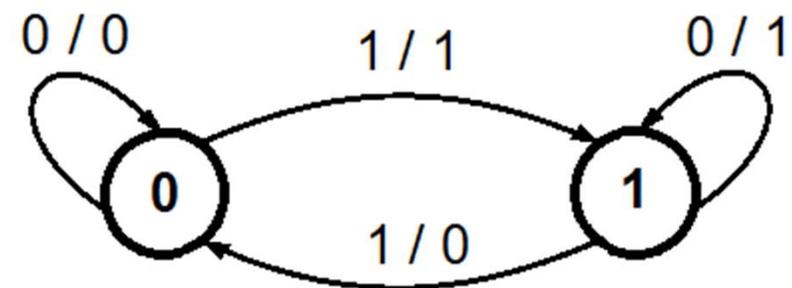
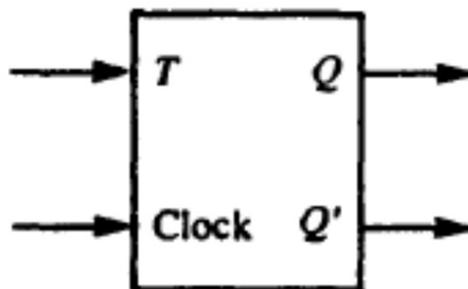
Biastable T (Toggle)

Tabla de transiciones

T	Q _n	Q _{n+1}
0	0	0
0	1	1
1	0	1
1	1	0

Q _n	Q _{n+1}	T
0	0	0
0	1	1
1	0	1
1	1	0

Símbolo gráfico



SN74LS76A

Dual JK Flip-Flop with Set and Clear

MODE SELECT – TRUTH TABLE

OPERATING MODE	INPUTS				OUTPUTS	
	\overline{S}_D	\overline{C}_D	J	K	Q	\overline{Q}
Set	L	H	X	X	H	L
Reset (Clear)	H	L	X	X	L	H
*Undetermined	L	L	X	X	H	H
Toggle	H	H	h	h	\overline{q}	q
Load "0" (Reset)	H	H	l	h	L	H
Load "1" (Set)	H	H	h	l	H	L
Hold	H	H	l	l	q	\overline{q}

* Both outputs will be HIGH while both \overline{S}_D and \overline{C}_D are LOW, but the output states are unpredictable if \overline{S}_D and \overline{C}_D go HIGH simultaneously.

H, h = HIGH Voltage Level

L, l = LOW Voltage Level

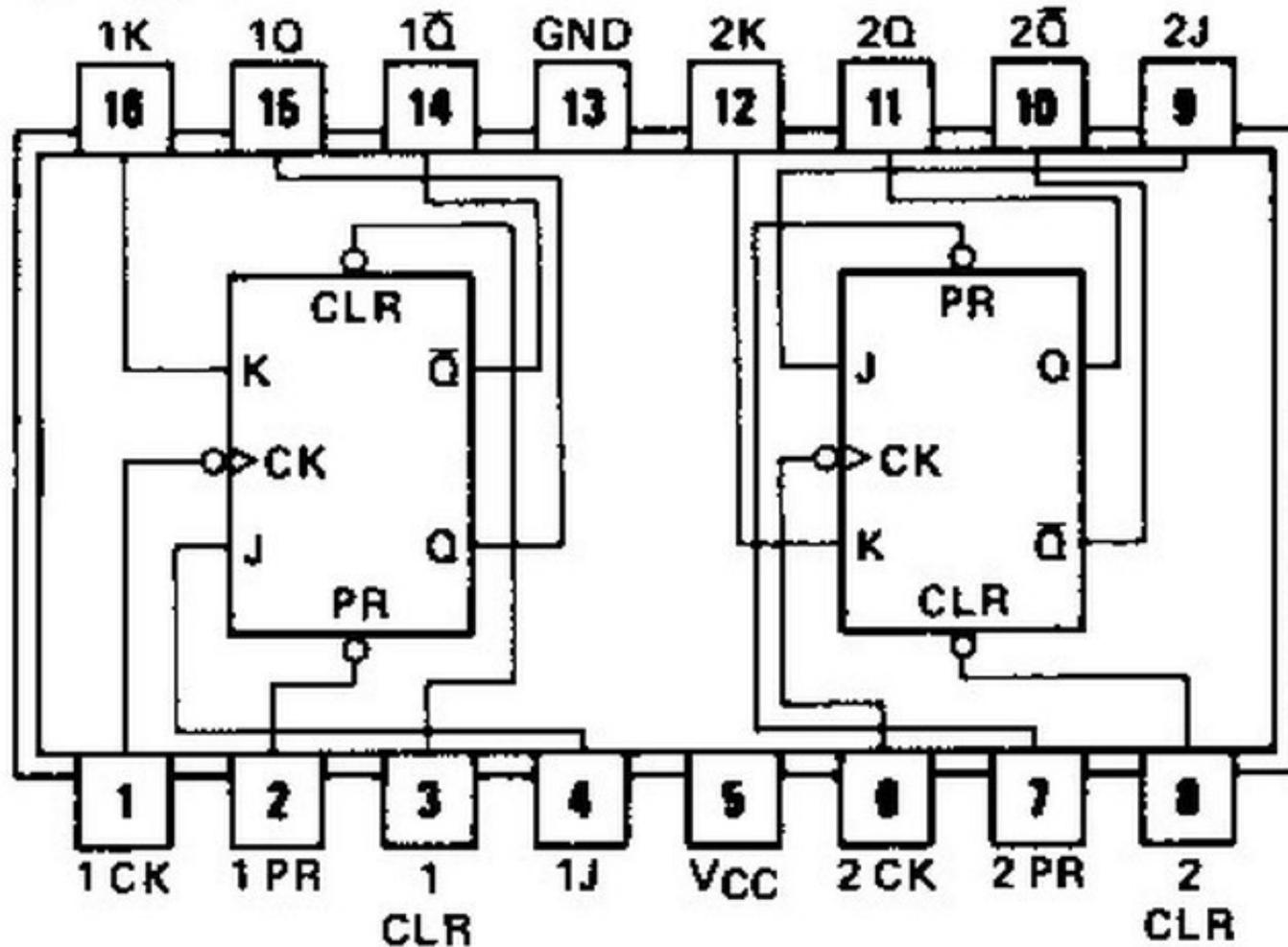
X = Immaterial

l, h (q) = Lower case letters indicate the state of the referenced input

(or output) one setup time prior to the HIGH-to-LOW clock transition

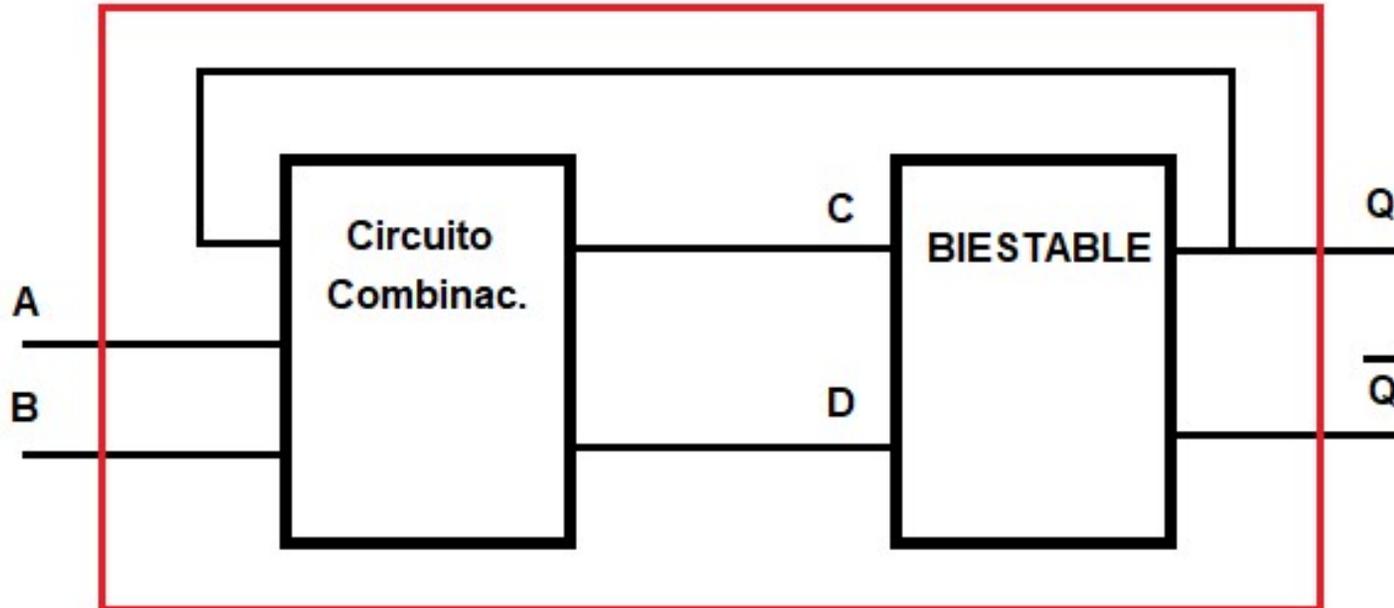
Data sheets

7476



Método de las Transiciones

El objetivo es obtener un FF (AB) a partir de otro (CD). (Sincrónico o asincrónico)



$$C = f_1 (A, B, Q^n)$$

$$D = f_2 (A, B, Q^n)$$

Vamos a estudiarlo mediante un ejemplo: Obtener el FF JK a partir del RS.

Obtendremos: $S = f_1 (J, K, Q^n)$ y $R = f_2 (J, K, Q^n)$

Método de las Transiciones

Partimos de la TV ampliada del JK y analizamos sus transiciones con la TT del RS.

Qn	Qn+1	S	R
0	0	0	x
0	1	1	0
1	0	0	1
1	1	x	0

TT SR

TV JK

Obtenemos:

J	K	Qn	Qn+1	S	R
0	0	0	0	0	x
0	0	1	1	x	0
0	1	0	0	0	x
0	1	1	0	0	1
1	0	0	1	1	0
1	0	1	1	x	0
1	1	0	1	1	0
1	1	1	0	0	1

J	K	Qn	00	01	11	10
0			0	x	0	0
1			1	x	0	1

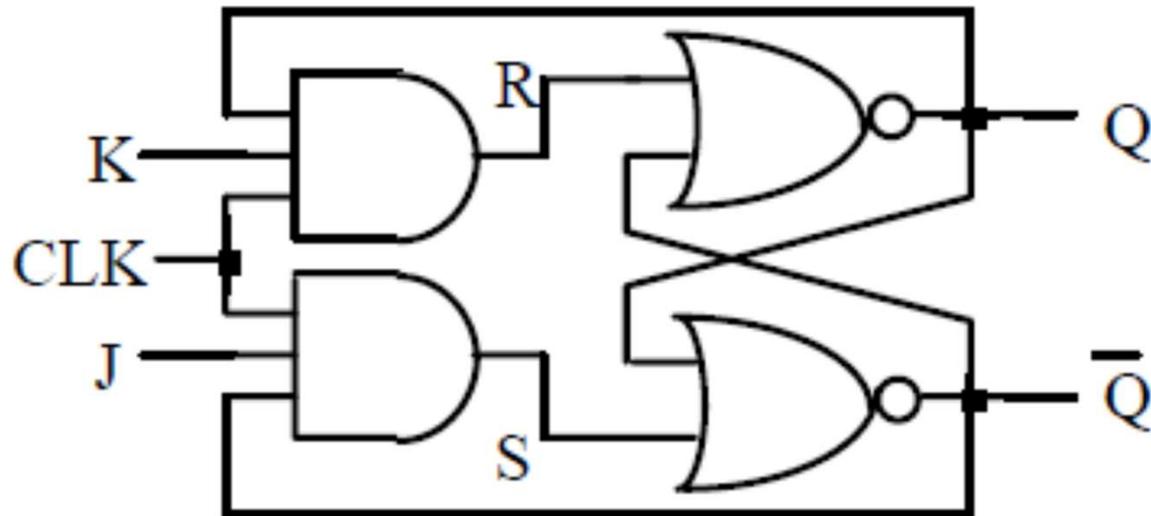
J	K	Qn	00	01	11	10
0			x	0	1	x
1			0	0	1	0

$$S = J \bar{Q}^n$$

$$R = K Q^n$$

Método de las Transiciones

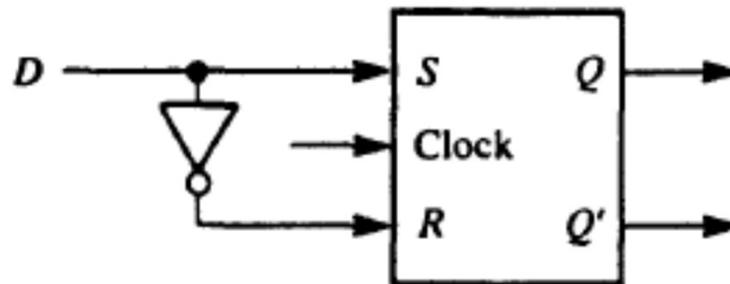
El circuito lógico de un JK síncrono de nivel alto quedará:



$$S = J \overline{Q^n} \text{ Clk}$$

$$R = K Q^n \text{ Clk}$$

También puedo obtener mediante este método un D a partir de un SR, un D a partir de un JK, etc.



Método de las Ecuaciones Características

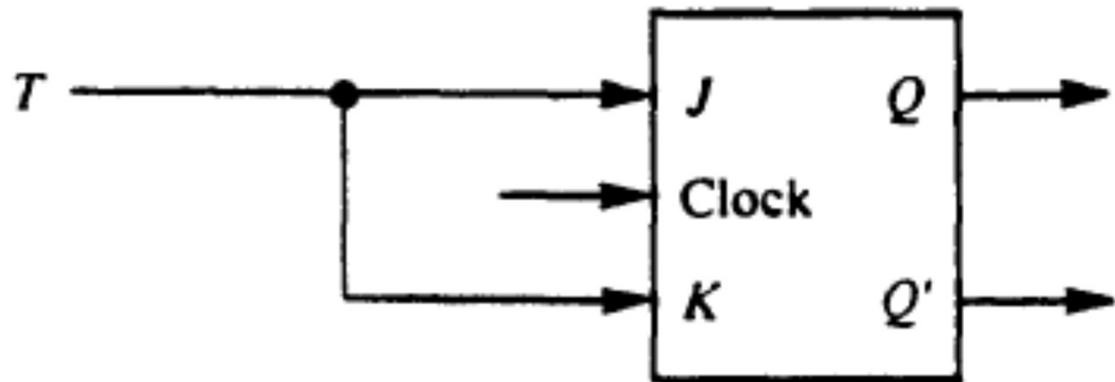
Consiste en igualar las EC de ambos FF. Ejemplo: Obtener un T a partir de un JK.

$$Q^{n+1} = J \bar{Q}^n + \bar{K} Q^n$$

$$Q^{n+1} = T \bar{Q}^n + \bar{T} Q^n$$

Comparando obtenemos:

$$J = T \quad \text{y} \quad K = T$$



Método de las Ecuaciones Características

Otro ejemplo: Obtener un D a partir de un JK.

$$Q^{n+1} = J \bar{Q}^n + \bar{K} Q^n$$

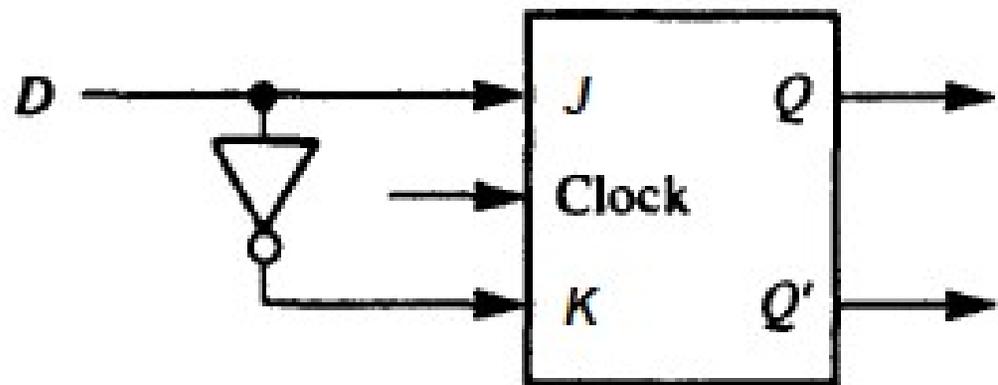
$$Q^{n+1} = D$$

$$Q^{n+1} = D \bar{Q}^n + D Q^n$$

Comparando obtenemos:

$$J = D \quad \text{y} \quad \bar{K} = D$$

$$\text{Por lo tanto:} \quad K = \bar{D}$$

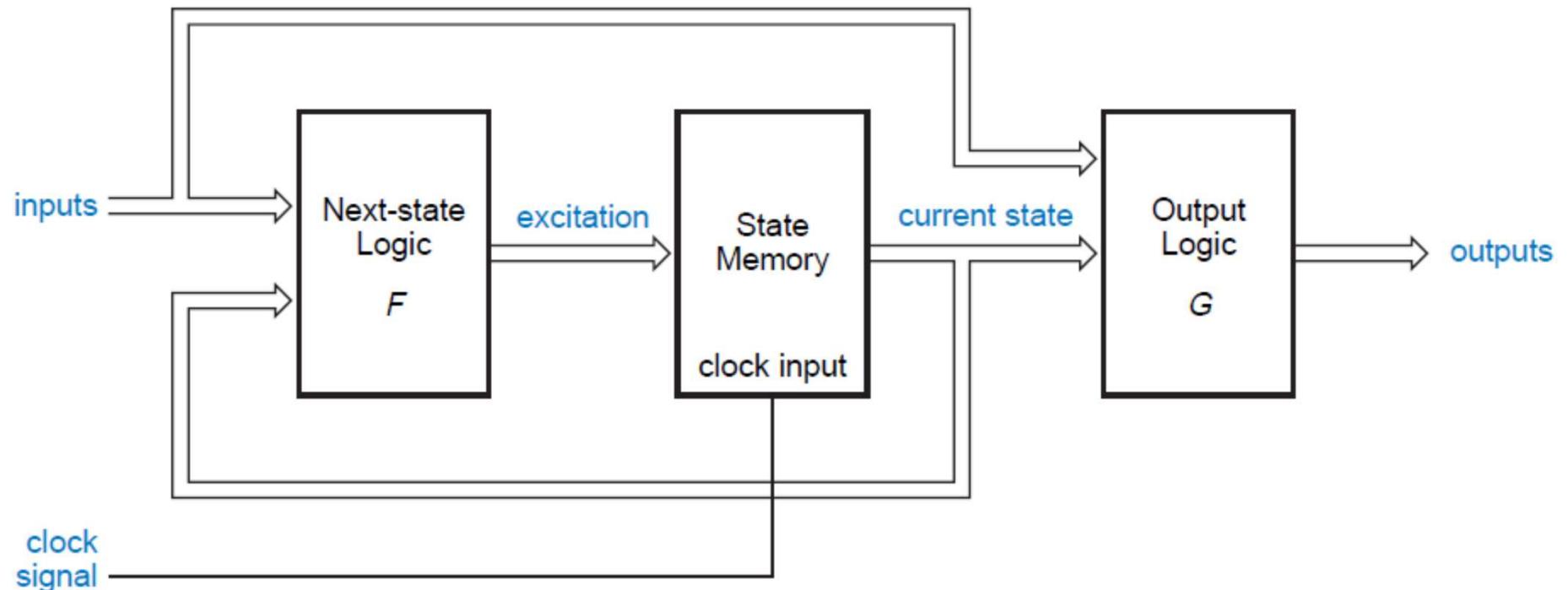


Máquina de Mealy

Memoria de estado: n biestables 2^n estados posibles reloj

Lógica de próximo estado: $F = f(\text{entradas, estado actual})$

Lógica de salida: $G = g(\text{entradas, estado actual})$



Máquina de Moore

Memoria de estado: n biestables 2^n estados posibles reloj

Lógica de próximo estado: $F = f(\text{entradas, estado actual})$

Lógica de salida: $G = g(\text{estado actual})$

