

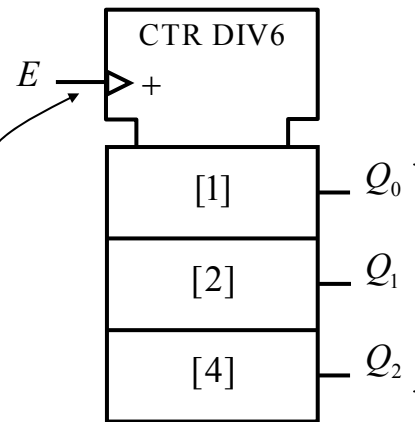
5.3.2: Bloques funcionales síncronos

5.3.2.1: **Contadores:** definiciones y características generales

DEF. 1: Un *contador* es un circuito *secuencial síncrono* que cuenta en un código binario dado los **flancos** de subida o bien de bajada que presenta la señal que se aplica a su entrada.

Nota: contar los flancos de subida o bien de bajada de una señal binaria equivale a contar sus pulsos.

*entrada dinámica
sensible a los flancos
de subida*

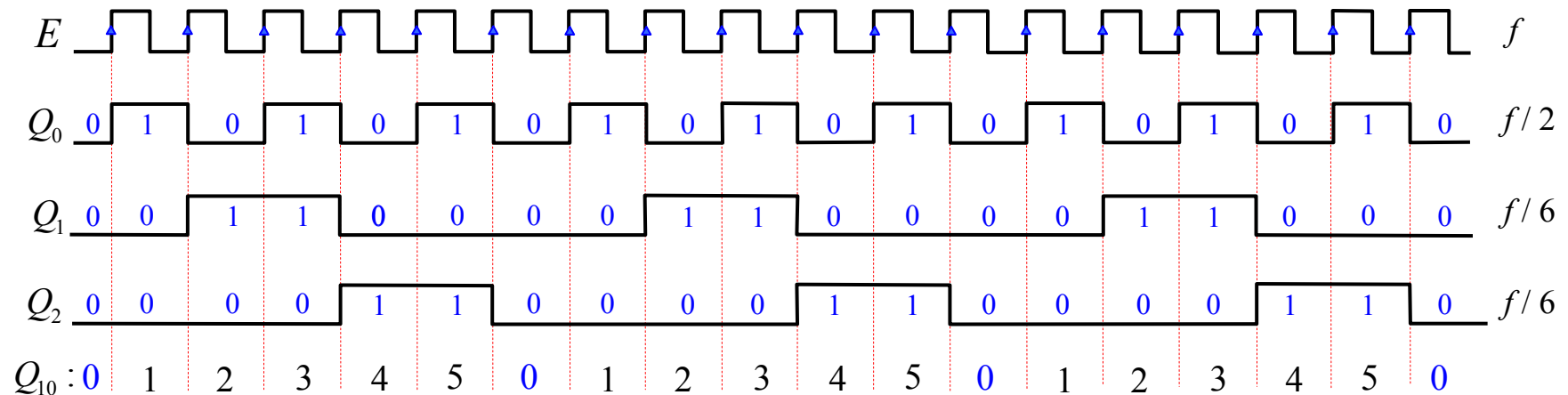


CTR \equiv **CounTeR**

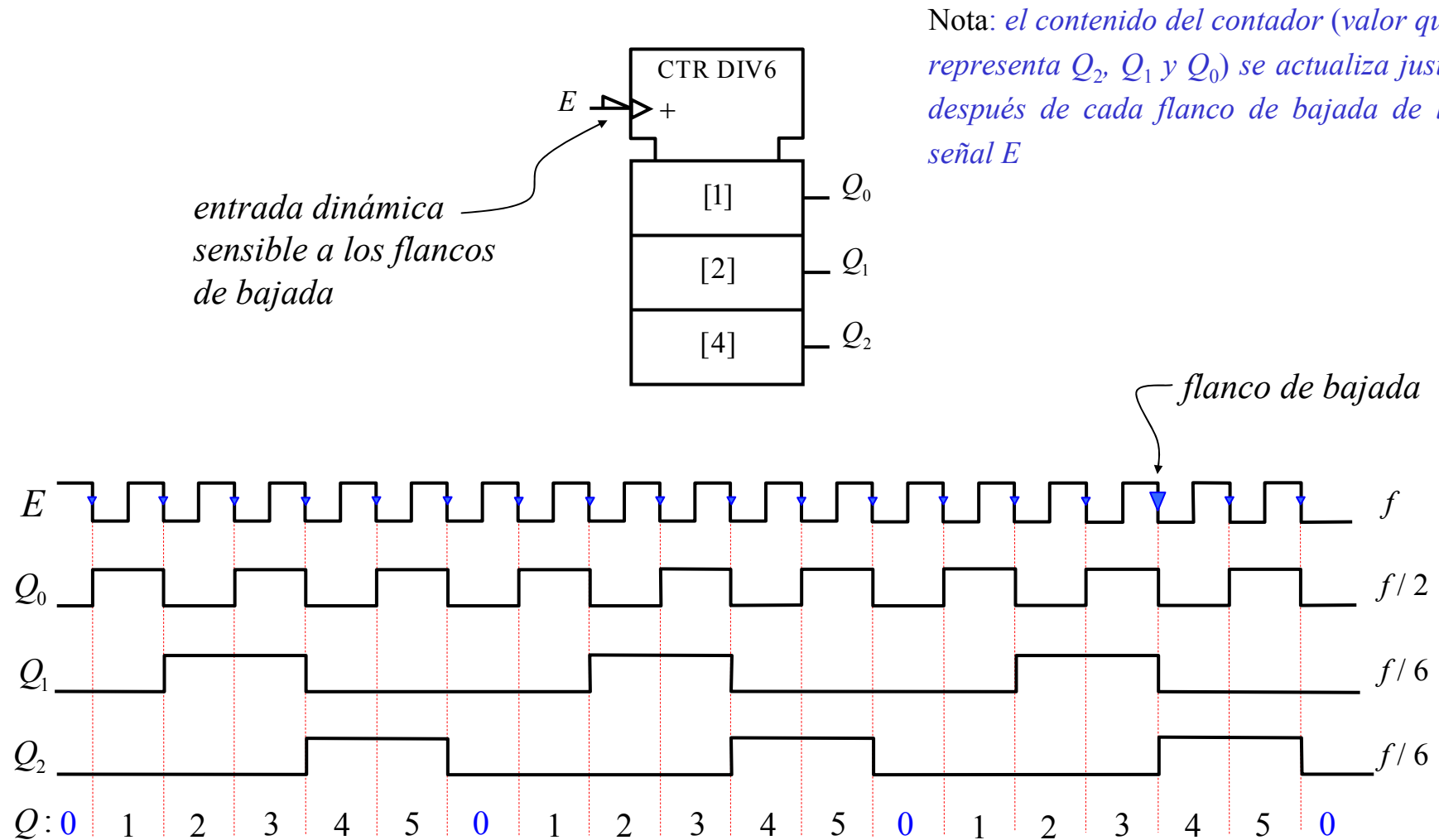
DIV m \equiv counter with cycle length = m

CTR n \equiv counter with cycle length = 2^n

*salidas del contador (representan el número
de flancos de subida que ha contado*)*

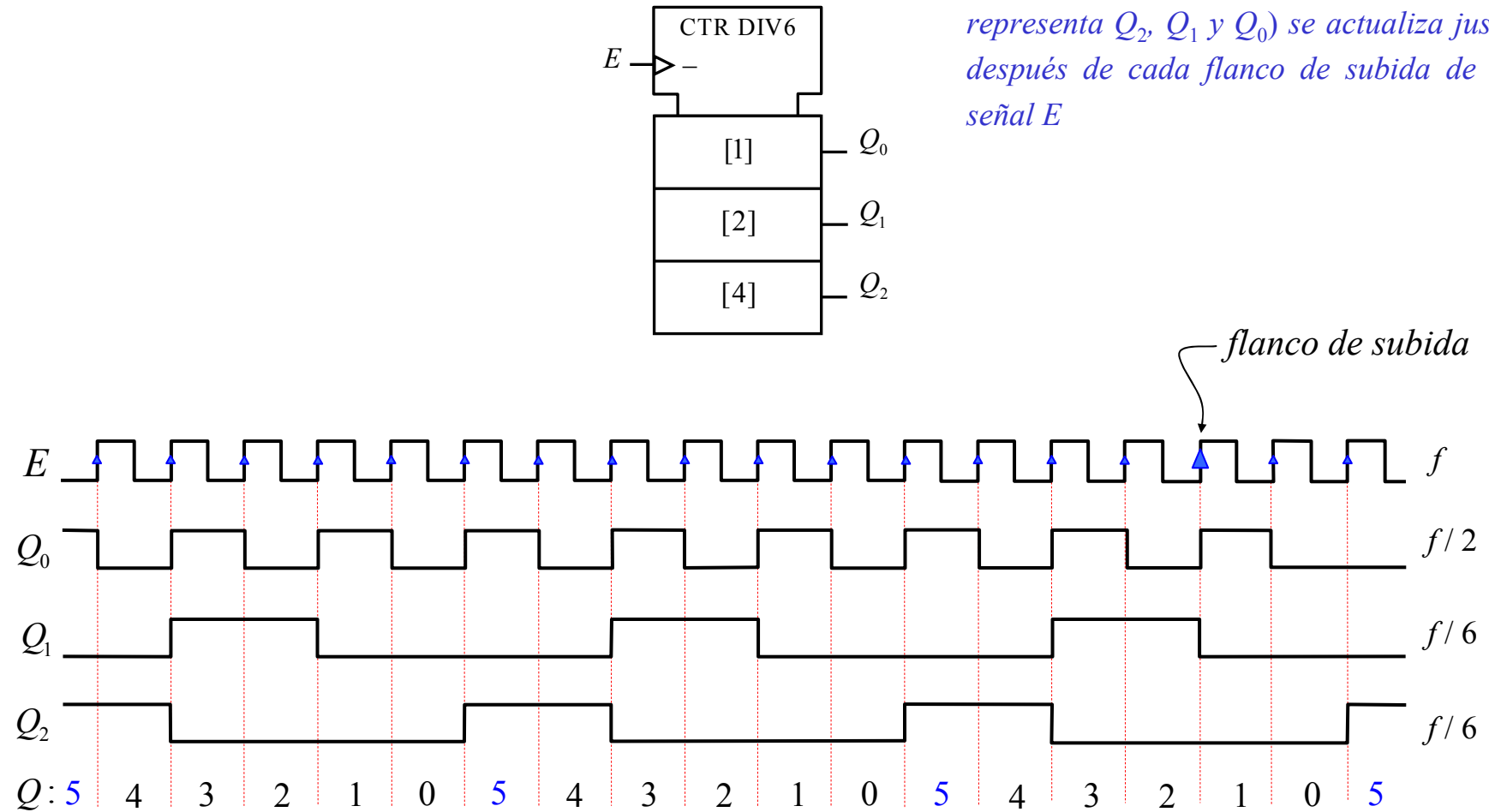


Nota: el contador representado en la parte superior cuenta los flancos de **subida** de la señal E , en modo **ascendente** (+)

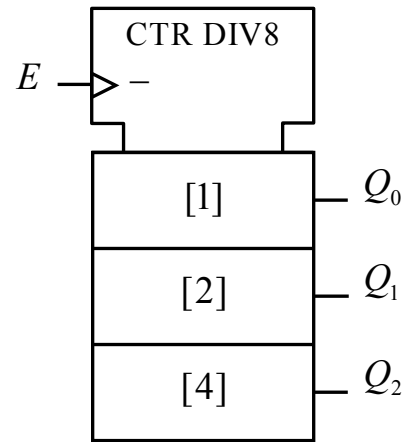


Nota: el contador representado en la parte superior cuenta los flancos de **bajada** de la señal E , en modo **ascendente (+)**

Nota: *el contenido del contador (valor que representa Q_2 , Q_1 y Q_0) se actualiza justo después de cada flanco de subida de la señal E*

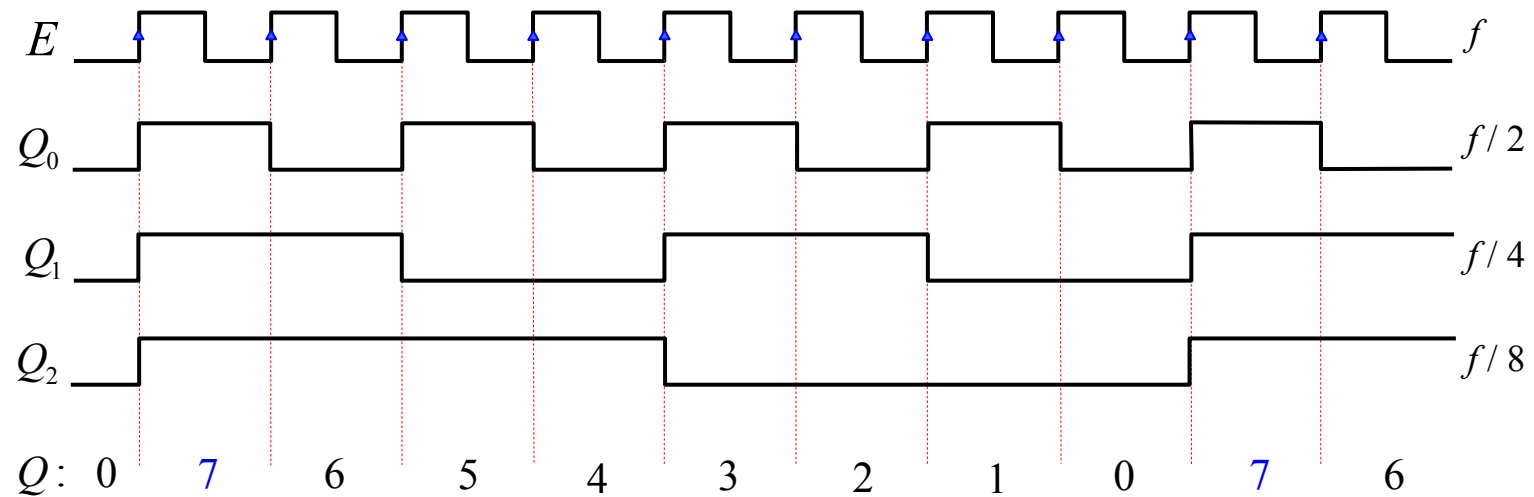


[cuenta flancos de subida en modo descendente (-)]



Nota: *el contenido del contador (valor que representa Q_2 , Q_1 y Q_0) se actualiza justo después de cada flanco de subida de la señal E*

Nota: $\text{CTR DIV } 8 \equiv \text{CTR } 3$



[cuenta flancos de subida en modo descendente (-)]

Propiedad: Un contador es un circuito secuencial que genera una secuencia ordenada de números (\equiv *secuencia de contaje*) de forma repetitiva.

Ejemplos:

0 - 1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 9 - 0 - 1 - 2 - 3 - \cdots (ascendente)

5 - 4 - 3 - 2 - 1 - 0 - 5 - 4 - 3 - \cdots (descendente)

3 - 11 - 15 - 9 - 0 - 5 - 3 - 11 - 15 - \cdots

Nota: en la literatura técnica se dice que “se produce un desbordamiento de un *contador*” cuando su *contenido* pasa del mayor valor de su secuencia de contaje al menor valor. Así, por ejemplo, cuando el contenido de un contador, con una secuencia de contaje 0-1-2-3-4-5-6-7, pasa de 7 a 0 se dice que se ha producido un desbordamiento (*the counter overflows*).

Propiedad: Un *contador* es un sistema secuencial síncrono cuya salida coincide, en general, con su estado interno.

DEF. 2: Se define el *módulo* de un contador como el número de estados distintos que puede tener su estado interno. Dicho de otro modo, el *módulo* de un contador es igual al número de valores que tiene su *secuencia de conteo*.

Ejemplos:

0 - 1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 9 - 0 - 1 - 2 - 3 - \cdots (módulo 10)

5 - 4 - 3 - 2 - 1 - 0 - 5 - 4 - 3 - \cdots (módulo 6)

3 - 11 - 15 - 9 - 0 - 5 - 3 - 11 - 15 - \cdots (módulo 6)

4 - 11 - 7 - 2 - 11 - 5 - 4 - 11 - 7 - 2 - \cdots (módulo 6) (*)

Propiedad: El número n de *flip-flops* que se necesitan para implementar un contador cumple lo siguiente:

$$2^n - 1 \geq \alpha$$

siendo α el mayor valor de su secuencia de conteo.

Ejemplos:

0 - 1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 9 - 0 - 1 - 2 - 3 - \cdots ($n = 4$)

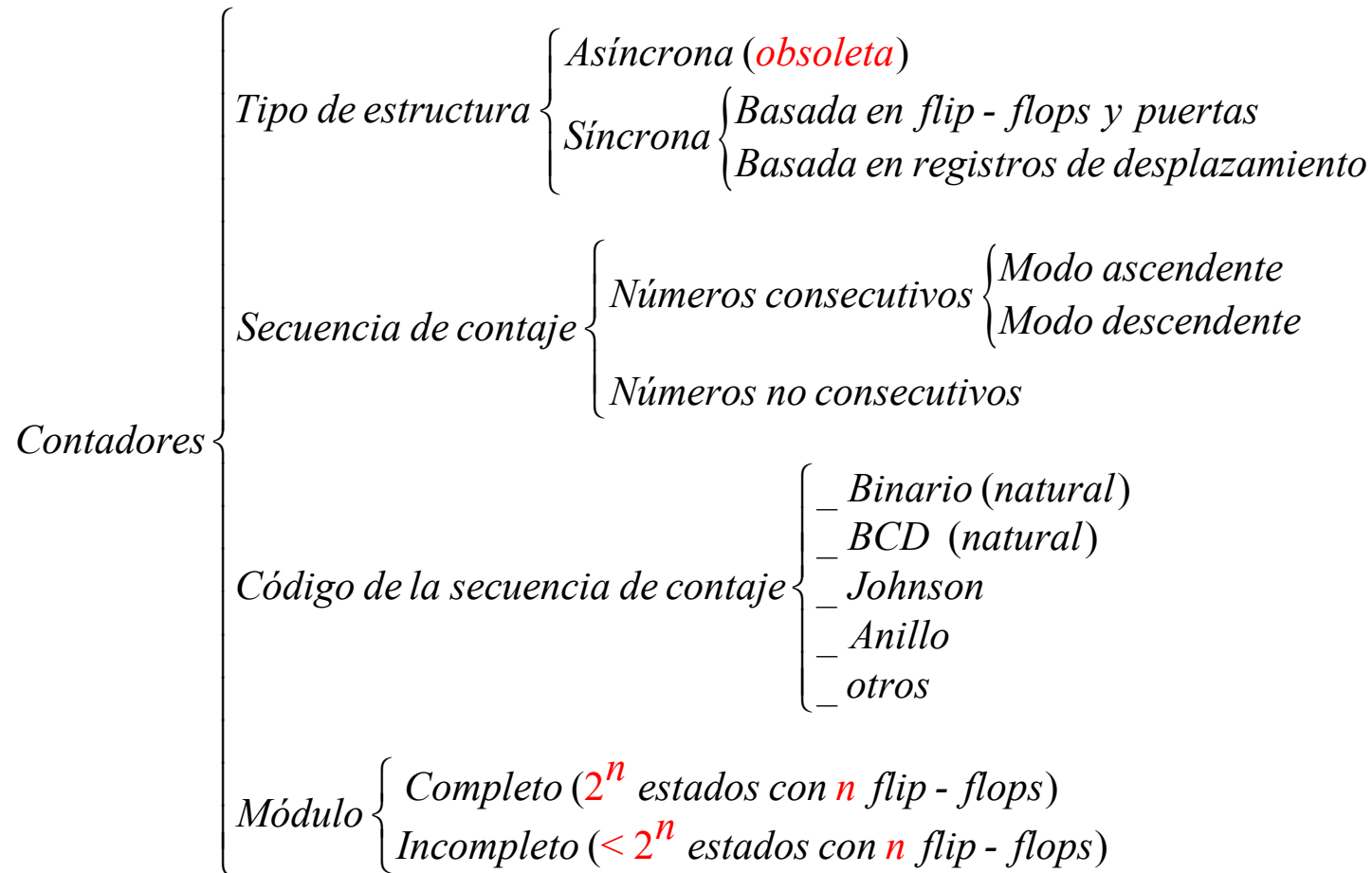
5 - 4 - 3 - 2 - 1 - 0 - 5 - 4 - 3 - \cdots ($n = 3$)

3 - 11 - 15 - 9 - 0 - 5 - 3 - 11 - 15 - \cdots ($n = 4$)

4 - 3 - 7 - 2 - 3 - 5 - 4 - 3 - 7 - 2 - \cdots ($n = 4$) (*)

Propiedad: Para guardar el **estado interno** de un contador se suelen utilizar *flip-flops* de tipo *D* o *J-K*, aunque se puede utilizar cualquier otro tipo de *flip-flop*.

Propiedad: En la práctica, los contadores se eligen en base a propiedades o características muy distintas entre sí:



Contadores síncronos

Características:

- Todos los *flip-flops* comparten la misma señal de sincronismo (*clk*).
- Se puede decir que prácticamente no presentan estados transitorios.
- Se diseñan siguiendo los mismos procedimientos de diseño que se utilizan con cualquier otro sistema secuencial síncrono, con la particularidad de que la entrada de sincronismo de los *flip flops* del bloque de memoria actúa como entrada del contador y que los cambios de estado son *incondicionales*.

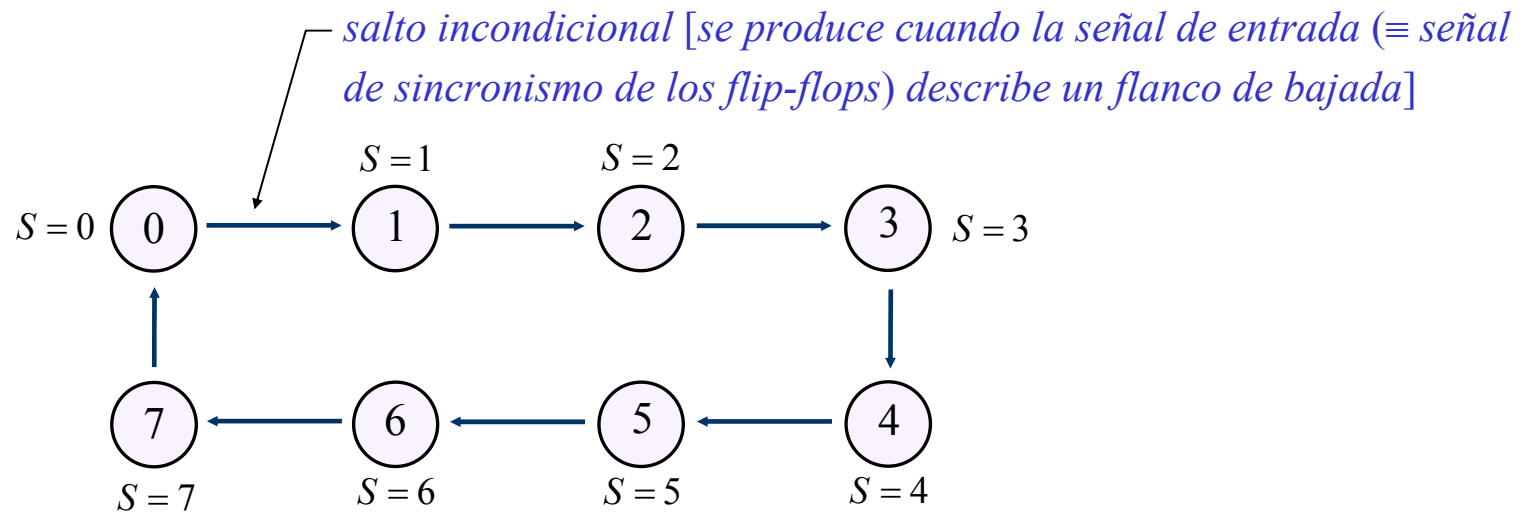
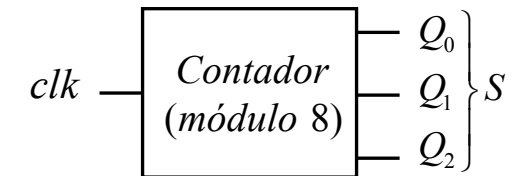
El comportamiento de los contadores síncronos se describe perfectamente mediante el *modelo de Moore*. Cumpliéndose que:

- No existe el circuito combinacional de salida (*ccs*)
- El circuito combinacional del estado siguiente (*cces*) sólo tiene como entradas las variables de estado.
- La entrada de sincronismo de los *flip-flops* también hace la función de entrada del contador

Ejemplo: diseño de un contador síncrono definido por:

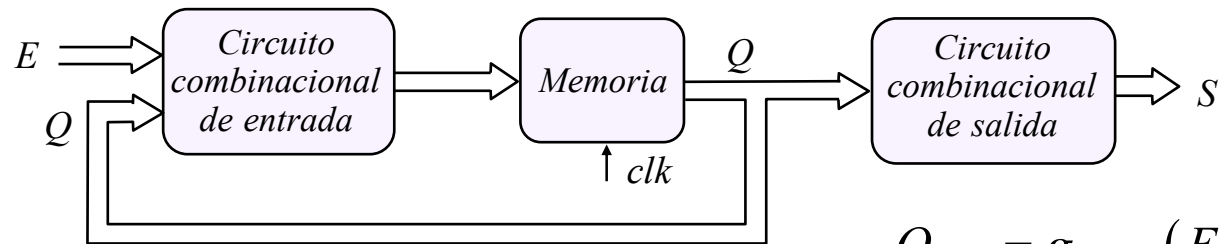
no

- ✓ módulo: 8
- ✓ código de la secuencia de conteo: *binario*
- ✓ modo de conteo: *ascendente* (0-1-2-3-4-5-6-7)
- ✓ cuenta flancos de: *bajada*



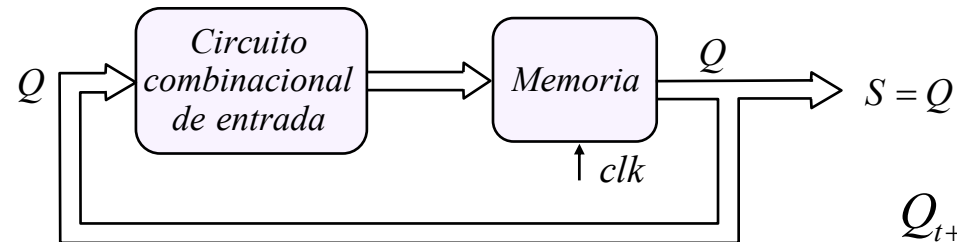
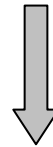
no

Modelo de Moore



$$Q_{t+\Delta t} = g_{Moore}(E_t, Q_t)$$

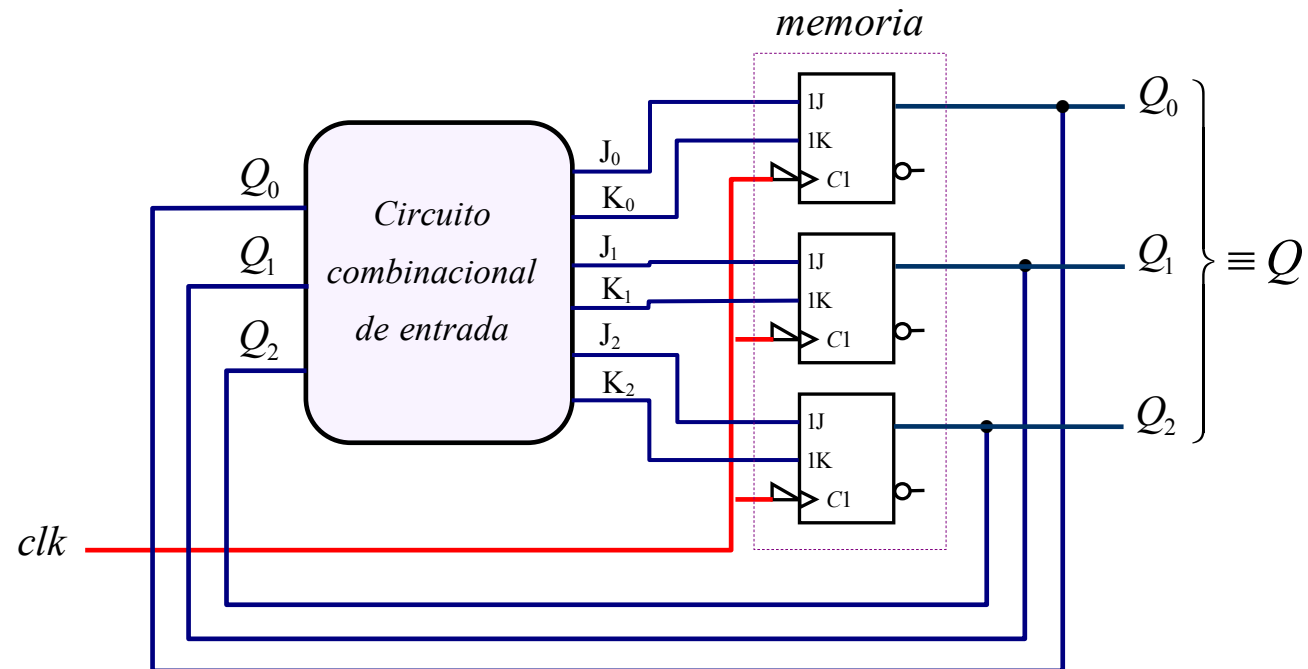
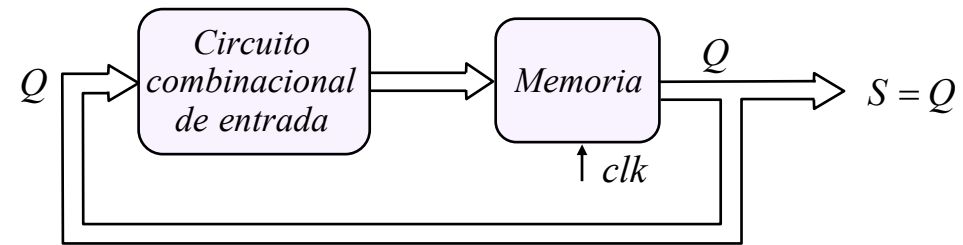
$$S_t = f_{Moore}(Q_t)$$



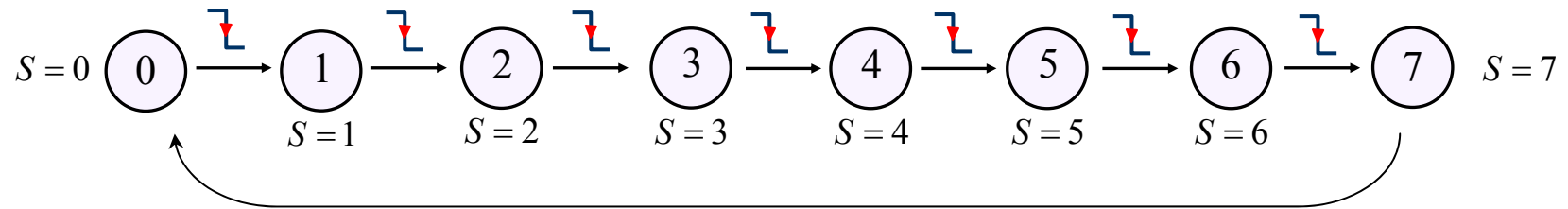
$$Q_{t+\Delta t} = g_{Moore}(Q_t)$$

$$S_t = Q_t$$

no

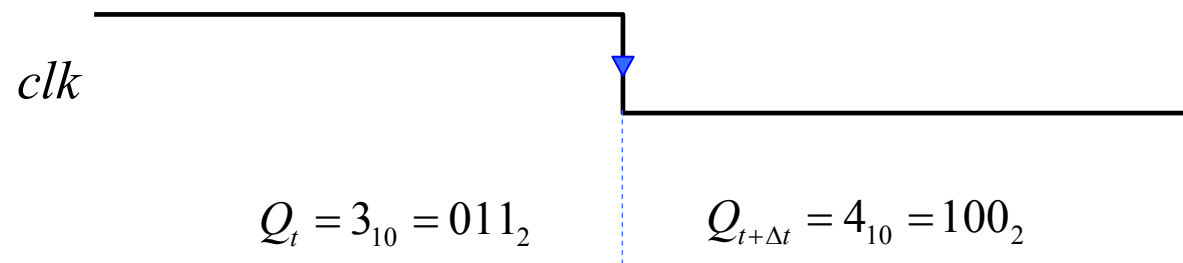
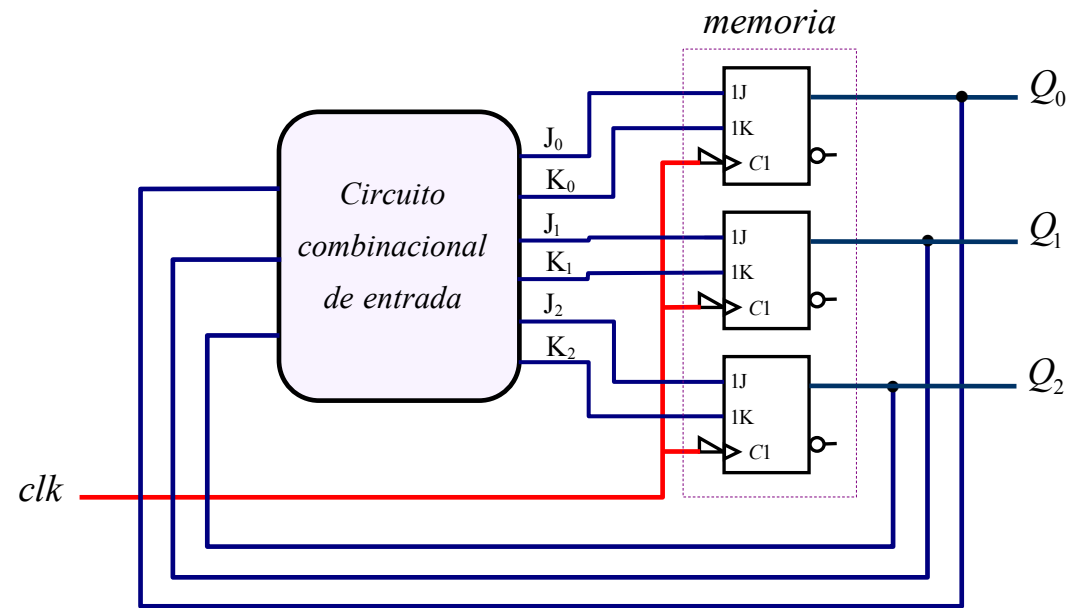


no



Q_{2t}	Q_{1t}	Q_{0t}	$Q_{2t+\Delta t}$	$Q_{1t+\Delta t}$	$Q_{0t+\Delta t}$
0	0	0	0	0	1
0	0	1	0	1	0
0	1	0	0	1	1
0	1	1	1	0	0
1	0	0	1	0	1
1	0	1	1	1	0
1	1	0	1	1	1
1	1	1	0	0	0

no



$J_2, K_2, J_1, K_1, J_0, K_0$?

no









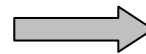
clk	J_t	K_t	Q_t	$Q_{t+\Delta t}$
	0	0	0	0
	0	0	1	1
<hr/>				
	0	1	0	0
	0	1	1	0
<hr/>				
	1	0	0	1
	1	0	1	1
<hr/>				
	1	1	0	1
	1	1	1	0
<hr/>				

Tabla de funcionamiento



$Q_t \rightarrow Q_{t+\Delta t}$	J_t	K_t
$0 \rightarrow 0$	0	x
$0 \rightarrow 1$	1	x
$1 \rightarrow 0$	x	1
$1 \rightarrow 1$	x	0

Tabla de excitación

no

$Q_t \rightarrow Q_{t+\Delta t}$	J_t	K_t
$0 \rightarrow 0$	0	x
$0 \rightarrow 1$	1	x
$1 \rightarrow 0$	x	1
$1 \rightarrow 1$	x	0

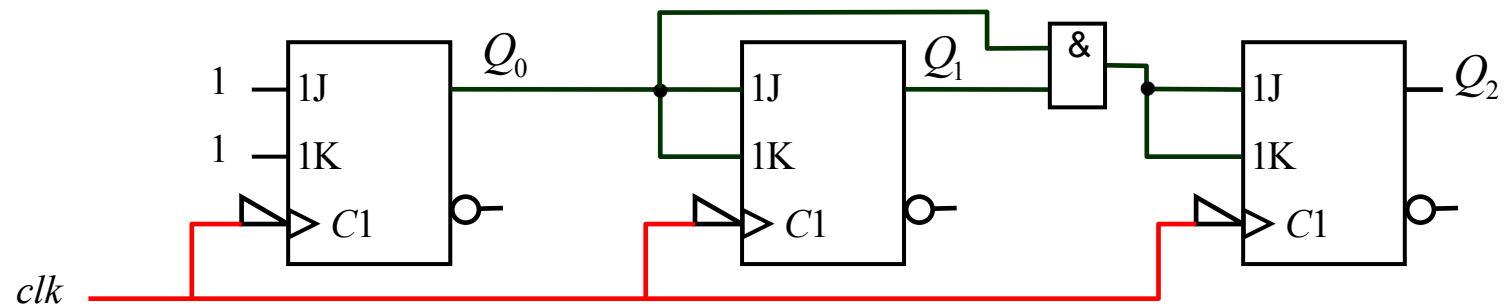
Q_{2t}	Q_{1t}	Q_{0t}	$Q_{2t+\Delta t}$	$Q_{1t+\Delta t}$	$Q_{0t+\Delta t}$	J_2	K_2	J_1	K_1	J_0	K_0
0	0	0	0	0	1	0	x	0	x	1	x
0	0	1	0	1	0	0	x	1	x	x	1
0	1	0	0	1	1	0	x	x	0	1	x
0	1	1	1	0	0	1	x	x	1	x	1
1	0	0	1	0	1	x	0	0	x	1	x
1	0	1	1	1	0	x	0	1	x	x	1
1	1	0	1	1	1	x	0	x	0	1	x
1	1	1	0	0	0	x	1	x	1	x	1
<i>antes de los flancos</i>			<i>después de los flancos</i>			<i>durante los flancos</i>					

no

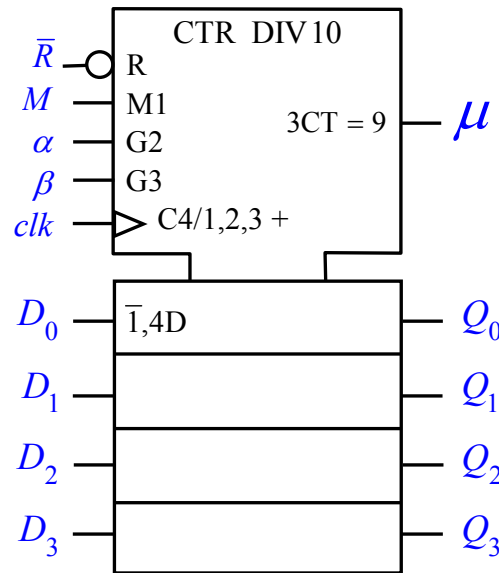
$$J_2(Q_2, Q_1, Q_0) = K_2(Q_2, Q_1, Q_0) = Q_1 Q_0$$

$$J_1(Q_2, Q_1, Q_0) = K_1(Q_2, Q_1, Q_0) = Q_0$$

$$J_0(Q_2, Q_1, Q_0) = K_0(Q_2, Q_1, Q_0) = 1$$



Contadores comerciales:



Notas sobre simbología:

DIV10 \equiv su módulo es igual a 10 (cuenta desde 0 hasta 9)

$\bar{R} \equiv$ entrada de reset: sirve para poner a 0 el contenido del contador
En este caso, es una entrada asíncrona, activa a nivel bajo.

$M \equiv$ entrada de selección de modo de funcionamiento:

_ con $M = 0$ se selecciona el modo de “carga en paralelo”

_ con $M = 1$ se selecciona el modo de “contaje”

α : entrada de inhibición del contaje

β : entrada de inhibición del contaje y de la salida de fin de cuenta.

clk (entrada dinámica): en este caso sincroniza tanto el contaje como la carga en paralelo (4).

$\mu \equiv$ salida que indica el fin de cuenta: se pone a 1 cuando $\beta = 1$ y el contenido del contador es igual a 9_{10} ($Q_3, Q_2, Q_1, Q_0 = 1001_2$)

D_3, D_2, D_1, D_0 : entradas en las que se aplica el valor a cargar en paralelo

Q_3, Q_2, Q_1, Q_0 : en estas salidas se indica en todo momento el contenido del contador en binario

Características del contador 74xxx160:

_ *módulo: 10 (contador de décadas)*

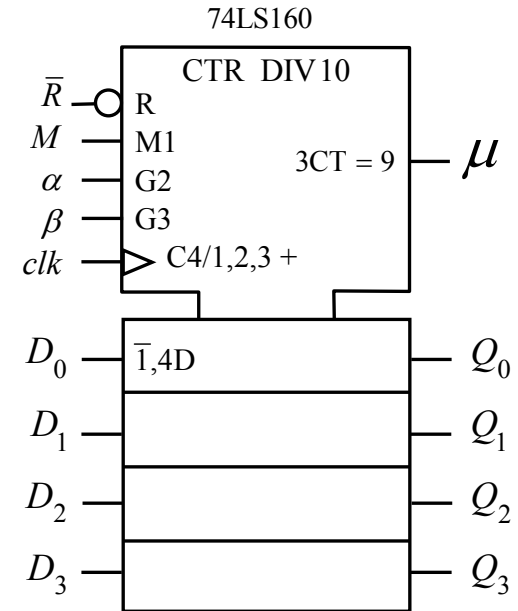
_ *reset: asíncrono, activo a nivel bajo*

_ *modo de conteo: ascendente*

_ *secuencia de conteo: 0 – 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 9*

_ *cuenta en binario*

Nota: en el momento en el que se activa la entrada de *reset* el contenido del contador se pone a cero.



Características del contador 74xxx162:

_ *módulo: 10* (*contador de décadas*)

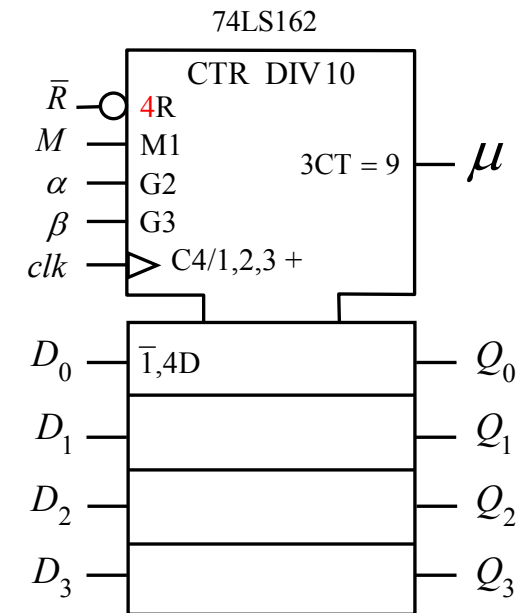
_ *reset: síncrono*, activo a nivel bajo

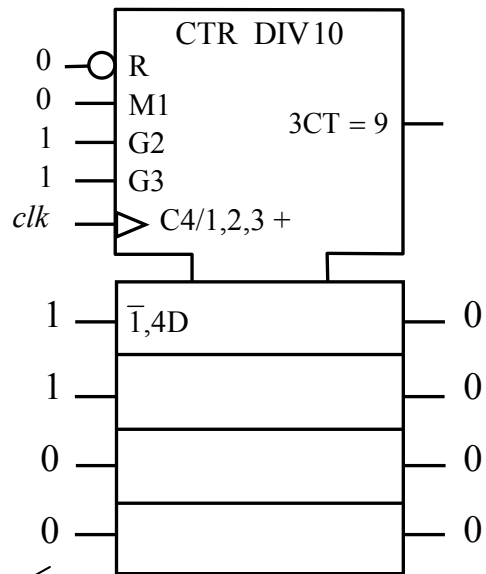
_ *modo de conteo: ascendente*

_ *secuencia de conteo: 0 – 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 9*

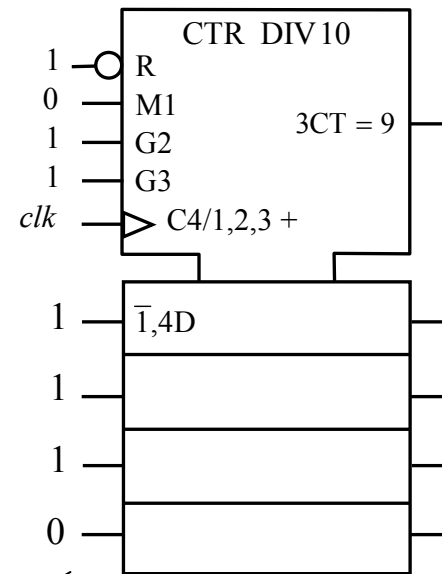
_ *cuenta en binario*

Nota: el contenido del contador se pone a cero si estando activa la entrada de reset, la señal *clk* describe un flanco de subida.

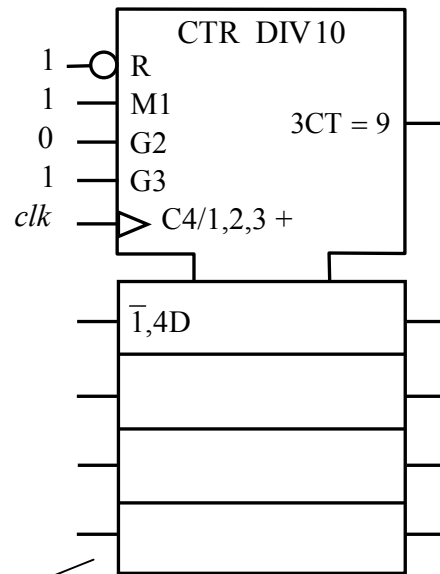




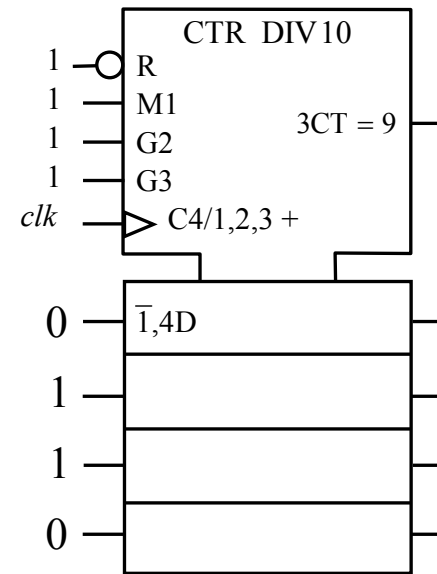
está seleccionado el modo de carga en paralelo, pero el contenido del contador está a cero, porque la entrada de *reset* está activa.



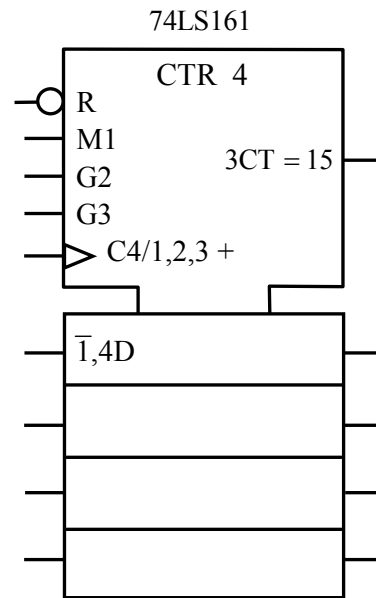
está seleccionado el modo de carga en paralelo. Cuando la señal *clk* describa un flanco de subida, el contenido del contador pasará a ser 0111_2 .



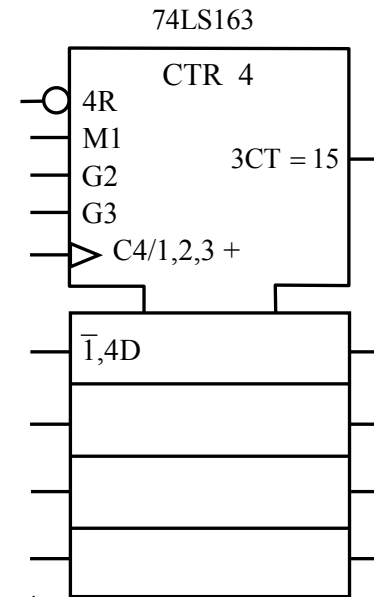
Está seleccionado el modo de *contaje*, pero dicho modo de funcionamiento está *inhibido*.



Está seleccionado el modo de *contaje*.
Cada vez que la señal *clk* describa un flanco de *subida*, el contenido del contador se incrementará en una unidad. Cuando el contenido del contador valga 1001_2 , con el siguiente flanco de subida de *clk*, el contenido del contador pasará a valer 0.

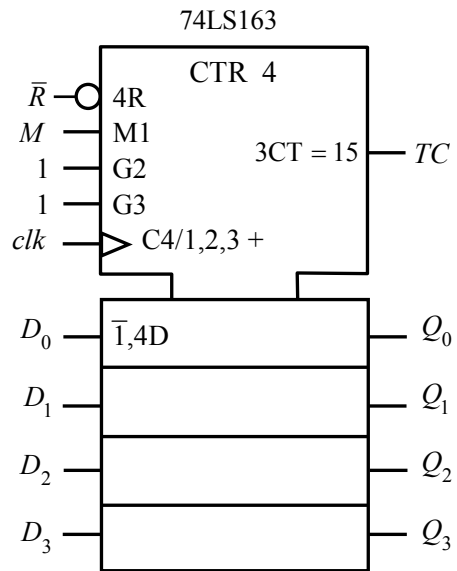


- _ Contador de 4 bits (módulo igual a 16)
- _ Reset *asíncrono*, activo a nivel bajo.
- _ Cuenta desde 0 hasta 15



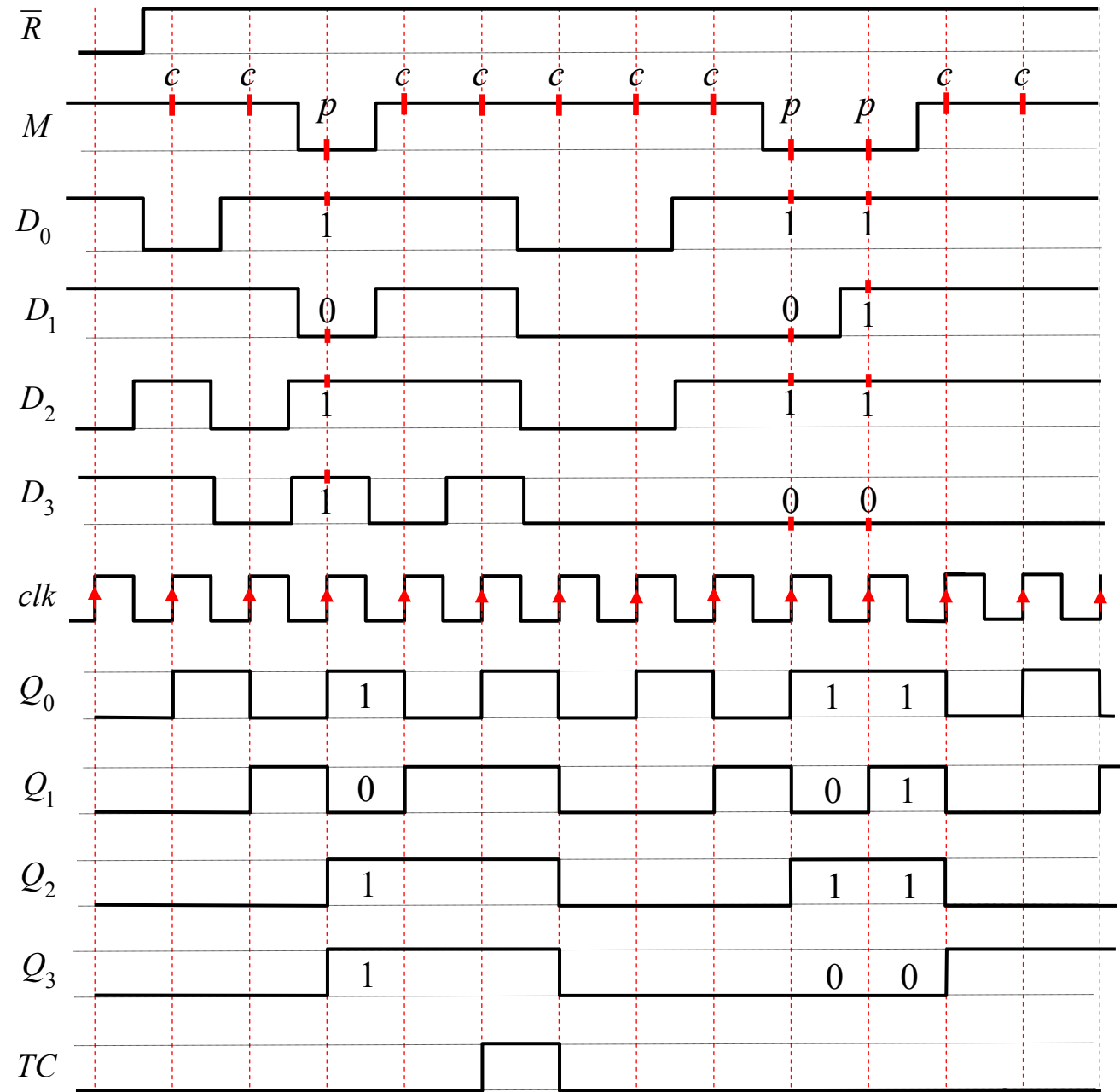
- _ Contador de 4 bits (módulo igual a 16)
- _ Reset *síncrono*, activo a nivel bajo.
- _ Cuenta desde 0 hasta 15

Ejemplo (cronograma)

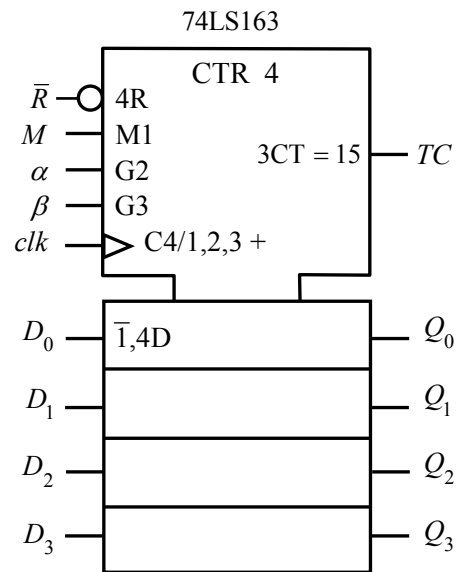


$M = 1$: cuenta (*c*)

$M = 0$: carga en paralelo (*p*)

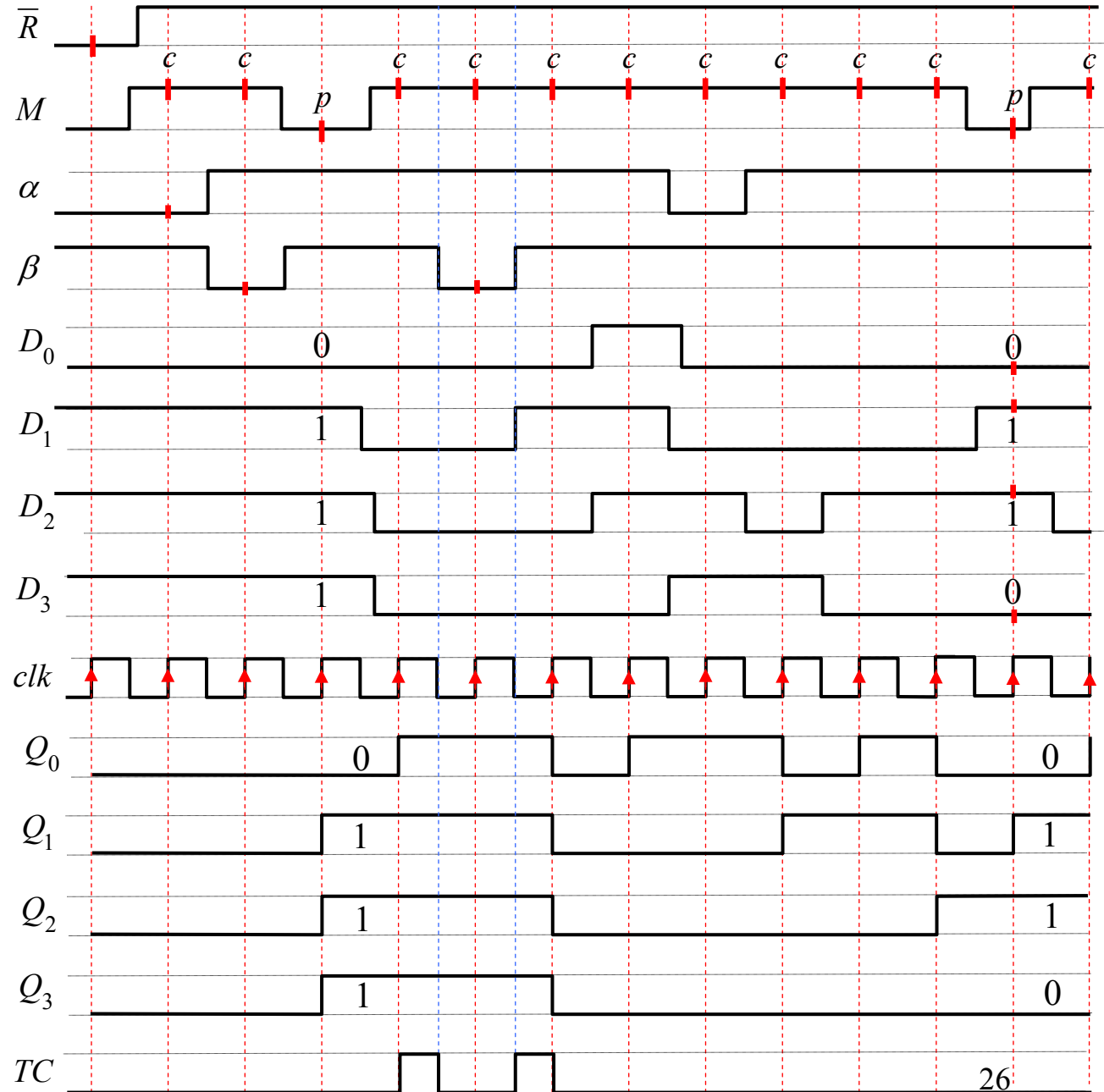


Ejemplo (cronograma)

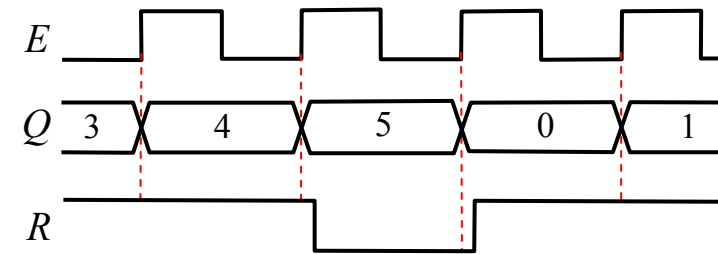
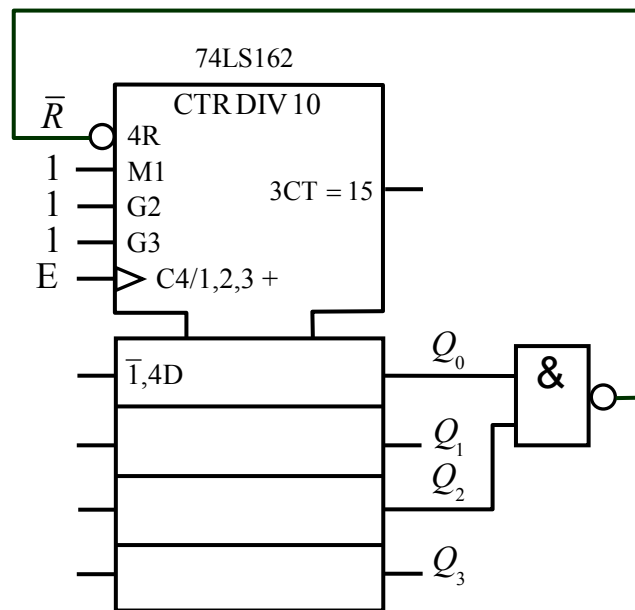


M = 1 : cuenta (c)

M = 0 : carga en paralelo (p)



Diseño de un contador de módulo 6 (0-1-2-3-4-5) a partir de un contador de módulo 10

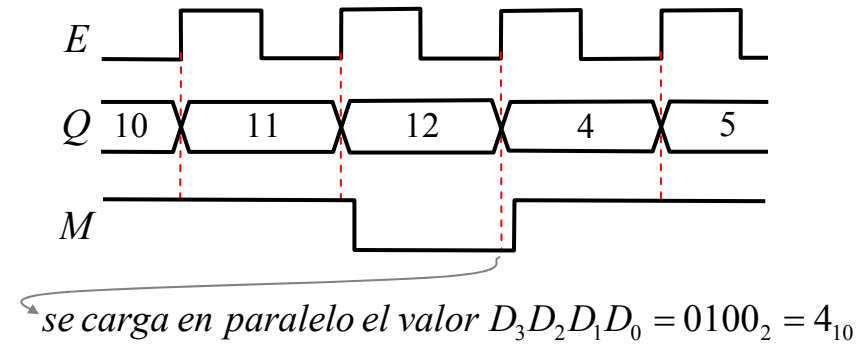
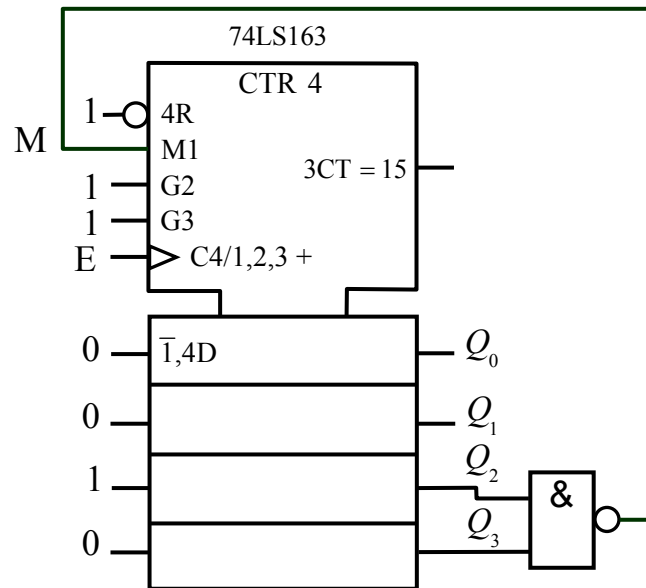


se pone a cero el contenido del contador.

Q_1Q_0	00	01	11	10
Q_3Q_2	1 ₀	1 ₁	1 ₃	1 ₂
01	1 ₄	0 ₅	x ₇	x ₆
11	x ₁₂	x ₁₃	x ₁₅	x ₁₄
10	x ₈	x ₉	x ₁₁	x ₁₀

$$\bar{R}(Q_3Q_2Q_1Q_0) = \bar{Q}_2 + \bar{Q}_0 = \overline{Q_2Q_0}$$

Diseño de un contador de módulo 9 (4-5-6-7-8-9-10-11-12) a partir de un contador de módulo 16



Q_1Q_0		00	01	11	10
Q_3Q_2	00	x_0	x_1	x_3	x_2
	01	1 ₄	1 ₅	1 ₇	1 ₆
	11	0 ₁₂	x_{13}	x_{15}	x_{14}
	10	1 ₈	1 ₉	1 ₁₁	1 ₁₀

$$M(Q_3Q_2Q_1Q_0) = \overline{Q_3} + \overline{Q_2} = \overline{Q_3Q_2}$$

Registros

Definición y características:

DEF.: Se entiende por **registro** de n bits a todo circuito capaz de guardar un dato de n bits.

- ✓ Los registros se utilizan para guardar datos temporalmente.
- ✓ Algunos tipos de registros se pueden utilizar para *cambiar el formato de la información (serie \leftrightarrow paralelo)*.
- ✓ Básicamente, un registro de n bits está formado por n *flip-flops* adecuadamente interconectados entre sí.
- ✓ No tienen una secuencia de estados característica.

✓ Se dice que los formatos de **entrada** y de **salida** de la información son de tipo:

_ **Entrada *serie***: si los bits a guardar en el registro se **introducen** uno a uno por una misma línea.

_ **Entrada *paralelo***: si los bits a guardar en el registro se **introducen** todos a la vez (se utiliza una línea independiente para cada bit).

_ **Salida *paralelo***: si los bits guardados en el registro se **pueden leer** (salen) todos a la vez (se utiliza una línea independiente para cada bit).

_ **Salida *serie***: si los bits guardados en el registro sólo se **pueden leer** (salen) uno a uno por una misma línea.

- ✓ De acuerdo con los formatos de entrada y de salida de la información, los registros se pueden clasificar de la siguiente manera:

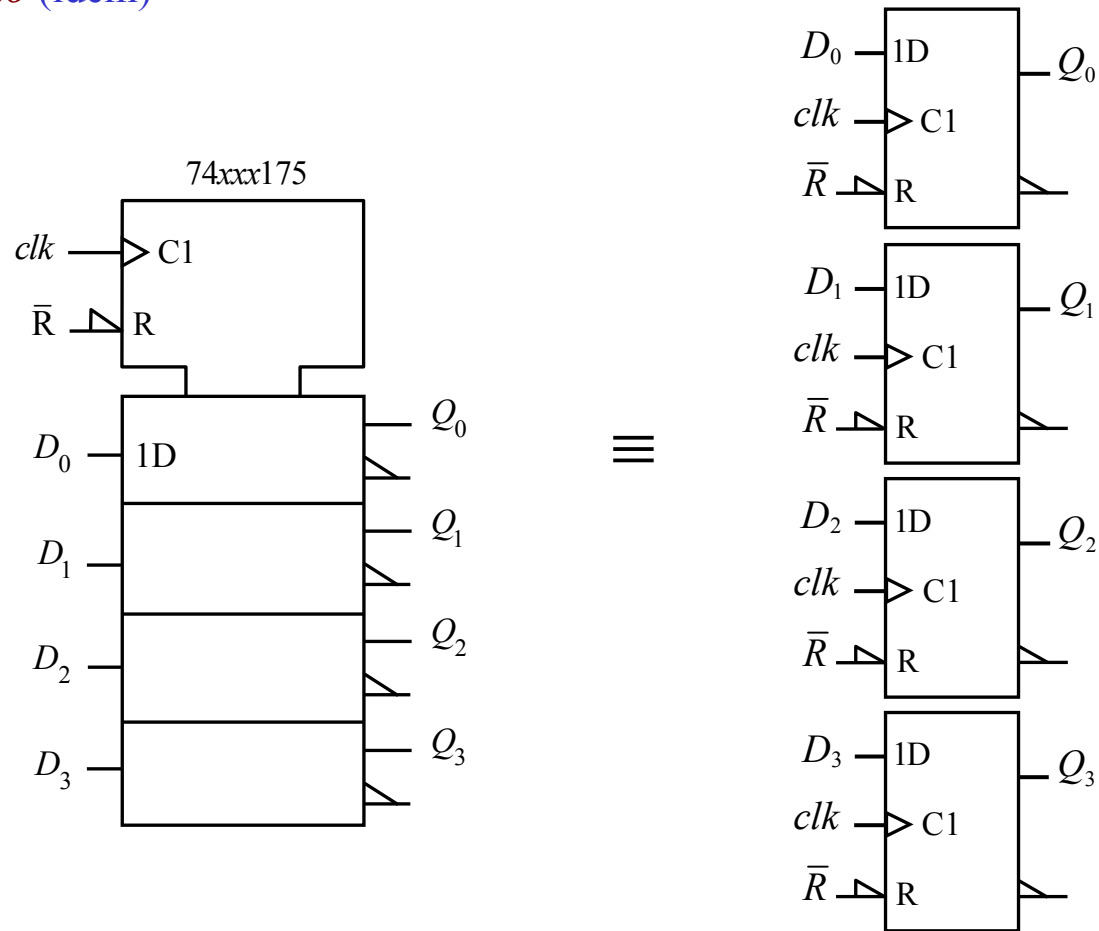
$$\text{Registros} \left\{ \begin{array}{l} \text{Registros de almacenamiento : entrada } \textcolor{red}{\text{paralelo}} \text{ y salida } \textcolor{red}{\text{paralelo}} \\ \text{Registros de desplazamiento} \left\{ \begin{array}{l} \text{entrada } \textcolor{red}{\text{serie}} \text{ y salida } \textcolor{red}{\text{serie}} \\ \text{entrada } \textcolor{red}{\text{serie}} \text{ y salida } \textcolor{red}{\text{paralelo}} \\ \text{entrada } \textcolor{red}{\text{paralelo}} \text{ y salida } \textcolor{red}{\text{serie}} \end{array} \right. \end{array} \right.$$

✓ Registro paralelo de n bits

Registro de almacenamiento de n bits

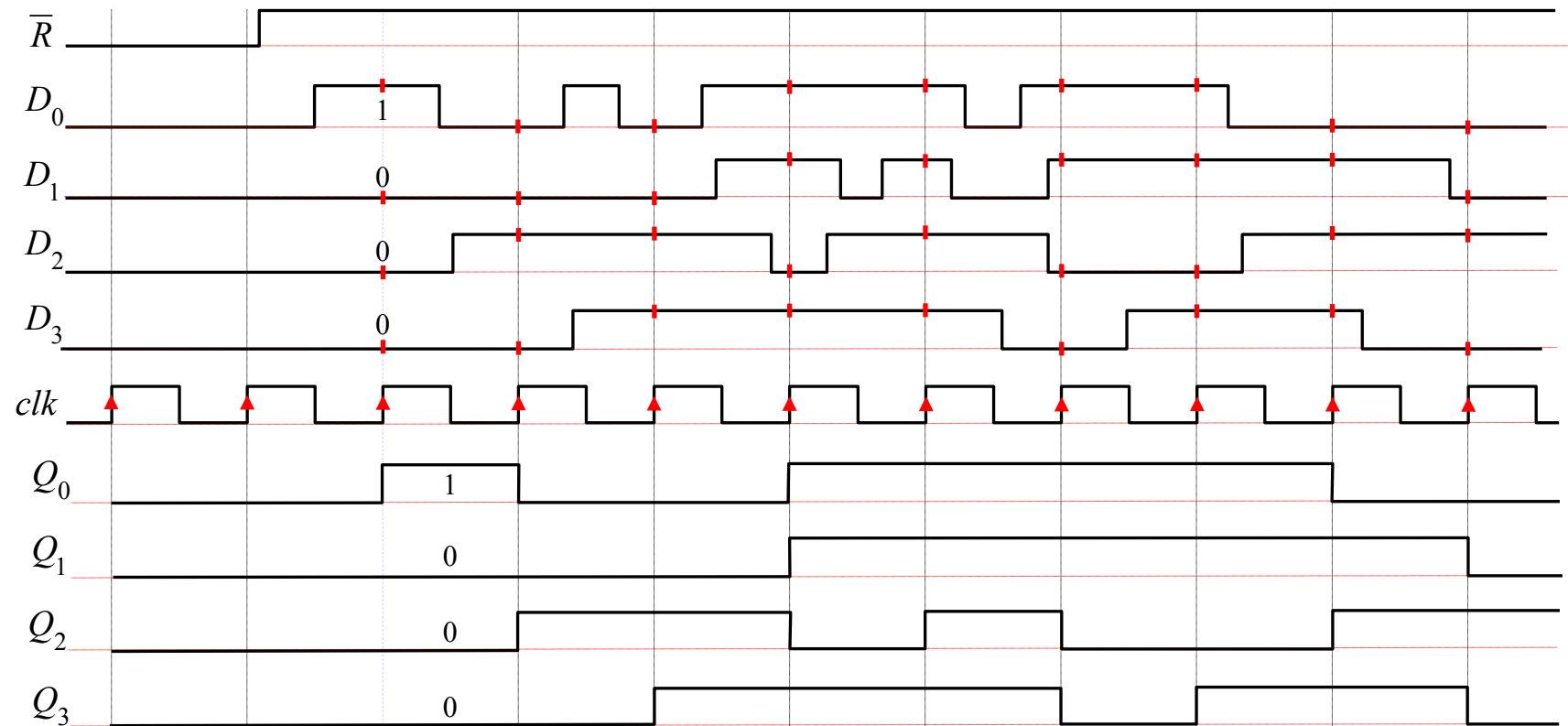
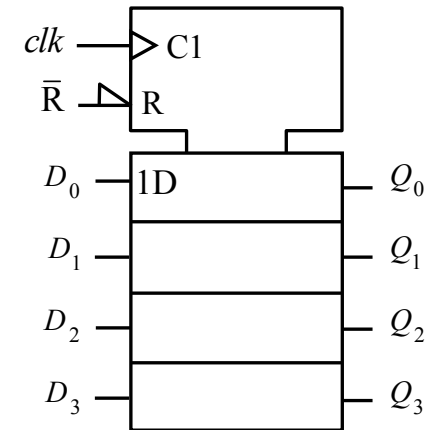
entrada: *paralelo* (los n bits se guardan a la vez y se tiene acceso a los n bits guardados a la vez)

salida: *paralelo* (ídem)



Ejemplo (cronograma)

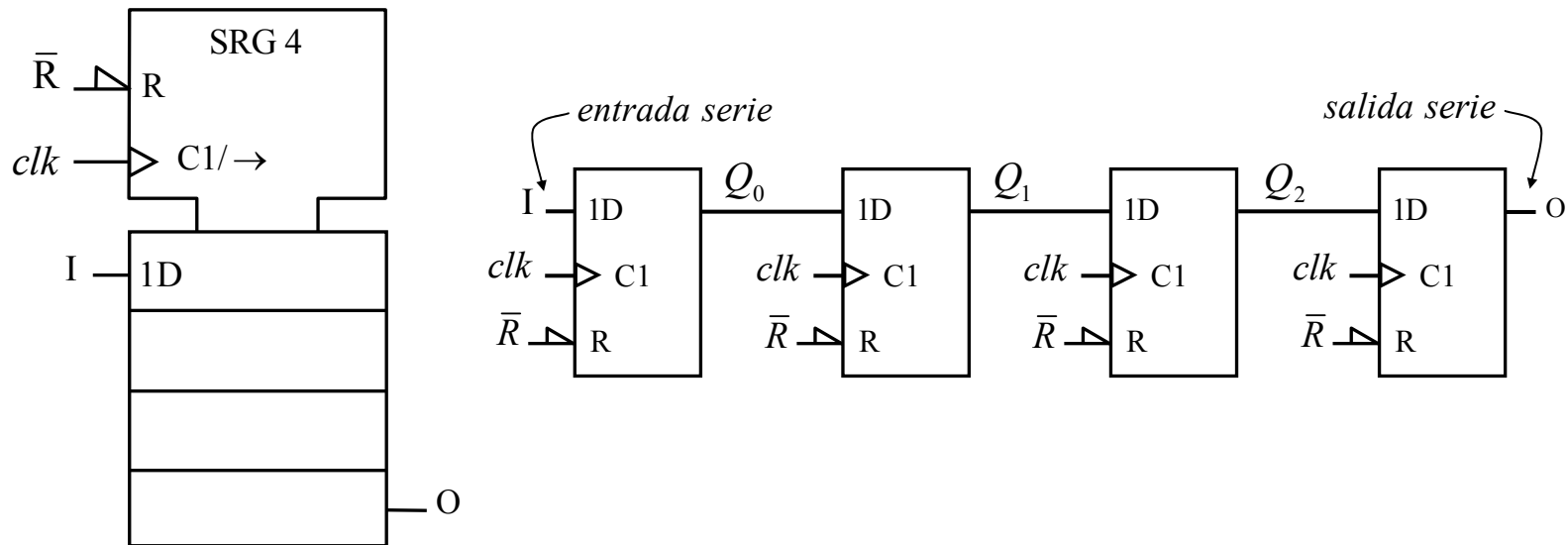
Nota: cada vez que la señal clk describe un flanco de subida, en el registro se guardan los valores que haya en las entradas D_3 , D_2 , D_1 y D_0 . A partir del momento en el que se activa la entrada de reset (R) en el registro se guarda el valor 0000. El contenido del registro se mantiene a cero (0000) mientras la entrada de reset esté activa.



✓ Registro de desplazamiento (*shift register*) de n bits

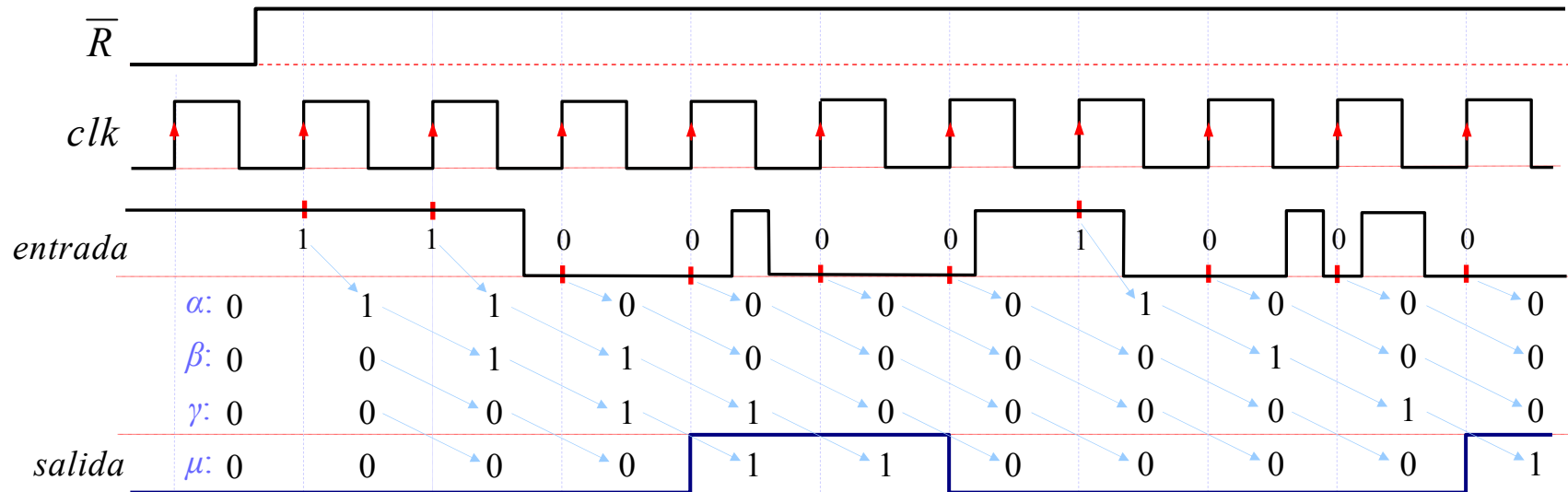
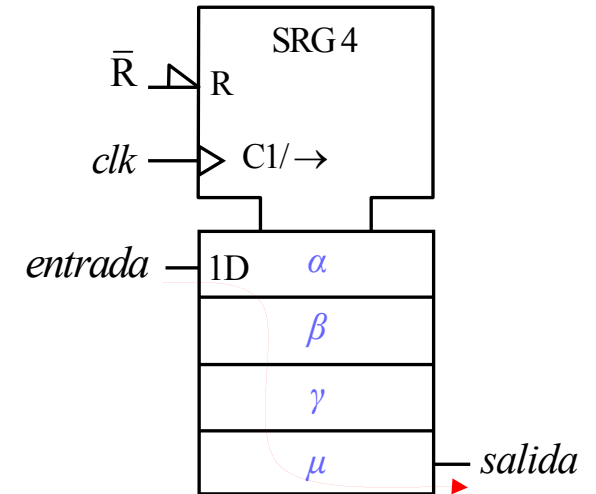
entrada: *serie* (los n bits se guardan de uno en uno)

salida: *serie* (se tiene acceso a los n bits guardados de uno en uno)



Ejemplo (cronograma)

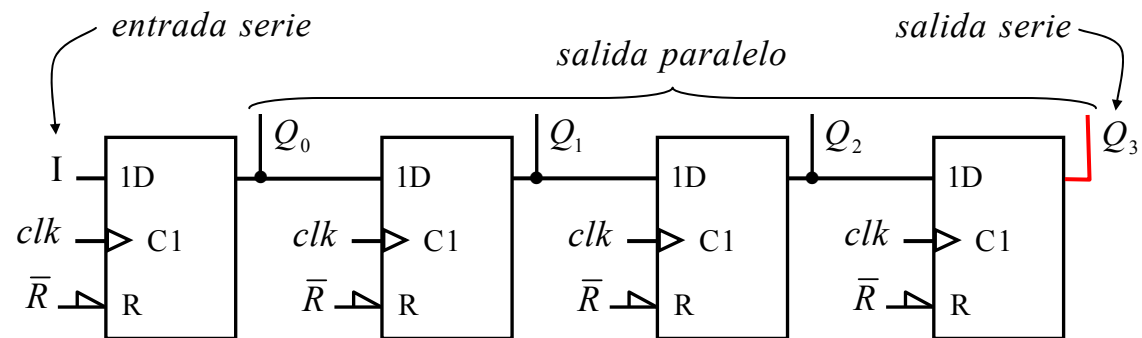
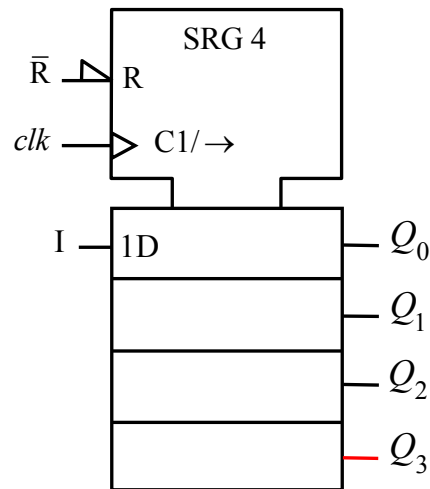
Nota: con cada flanco de subida de la señal clk se produce un desplazamiento de los valores guardados en el registro, a la vez que se introduce (en el registro) el valor que haya en la entrada. A partir del momento en el que se activa la entrada de reset (R), los valores de los 4 bits guardados en el registro pasan a ser 0000. El contenido del registro se mantiene a cero (0000) mientras la entrada de reset (R) esté activa.



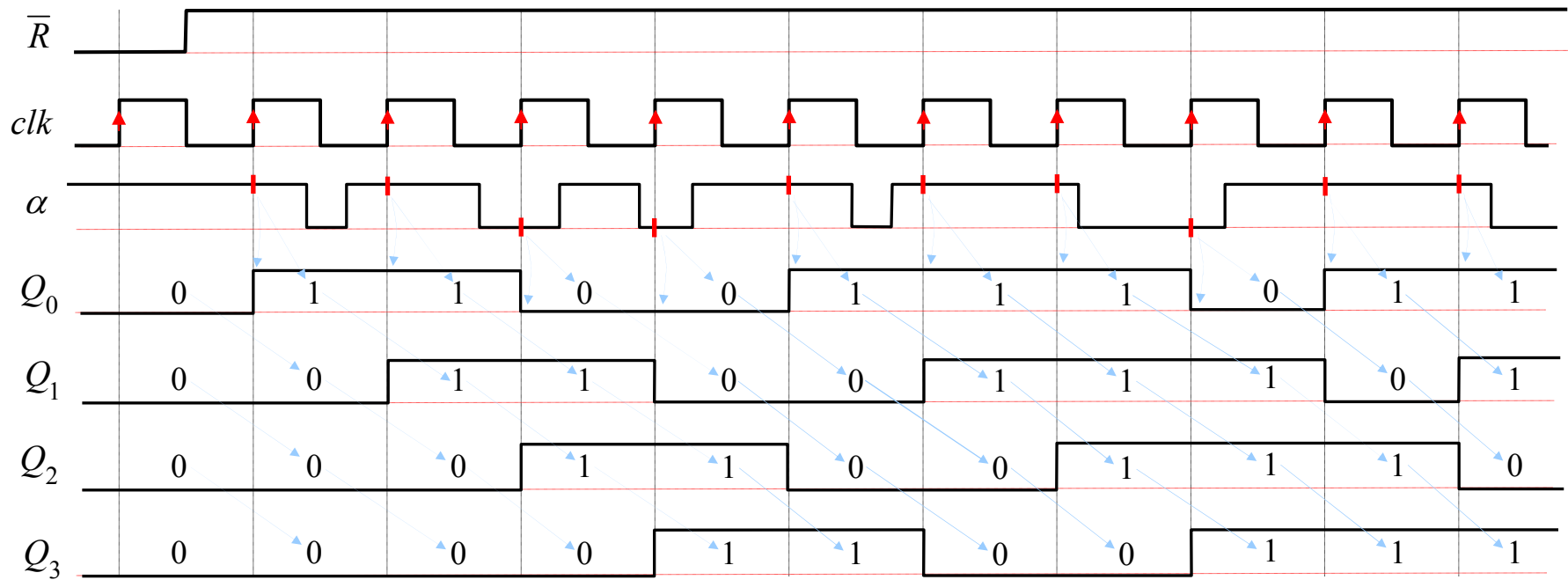
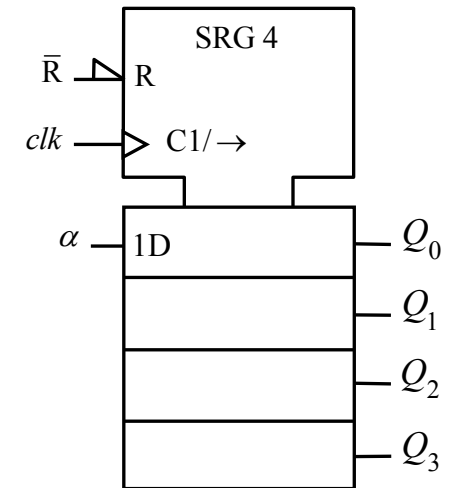
✓ Registro de desplazamiento (*shift register*) de n bits

entrada: *serie* (los n bits se guardan de uno en uno)

salida: *paralelo* o *serie* (se tiene acceso a la vez a los n bits guardados)



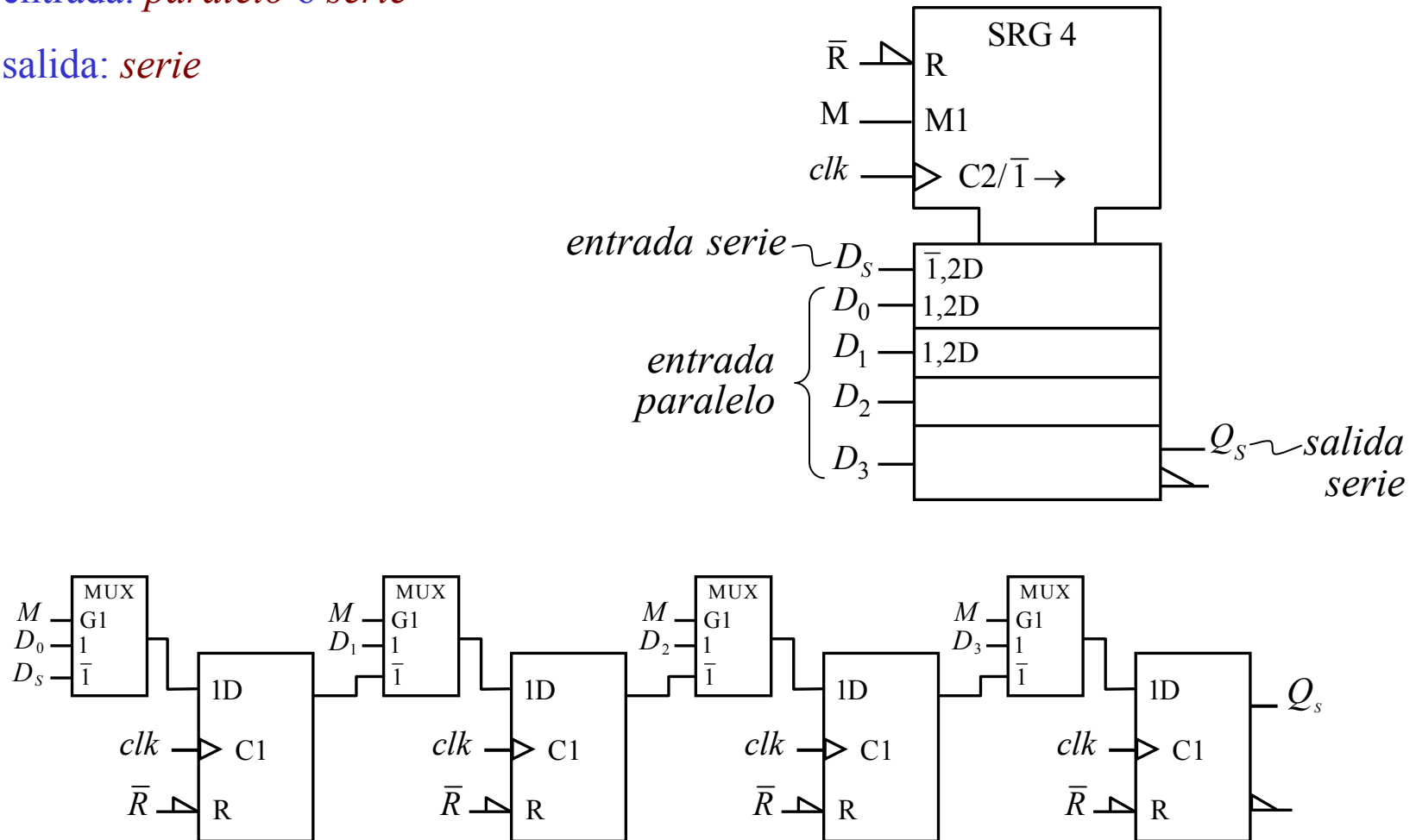
Ejemplo (cronograma)



✓ Registro de desplazamiento (*shift register*) de n bits

entrada: *paralelo o serie*

salida: *serie*

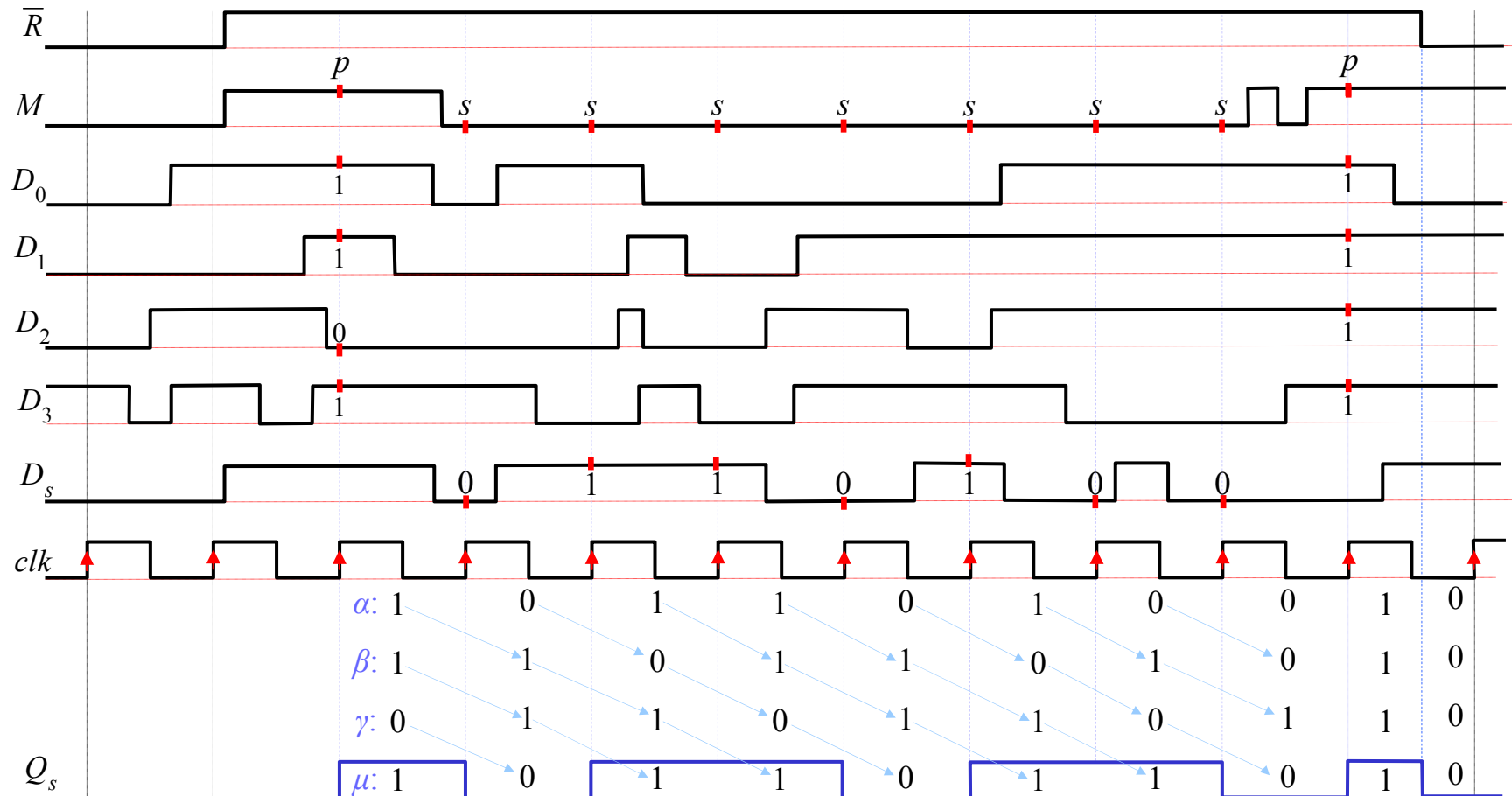
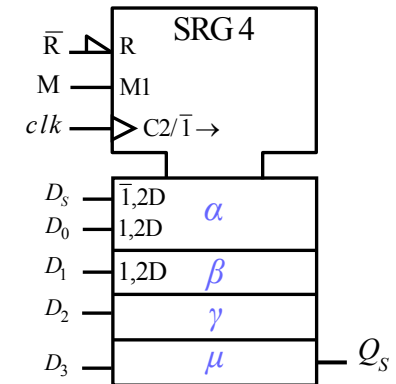


Ejemplo (cronograma)

$M = 1$: entrada paralelo (p)

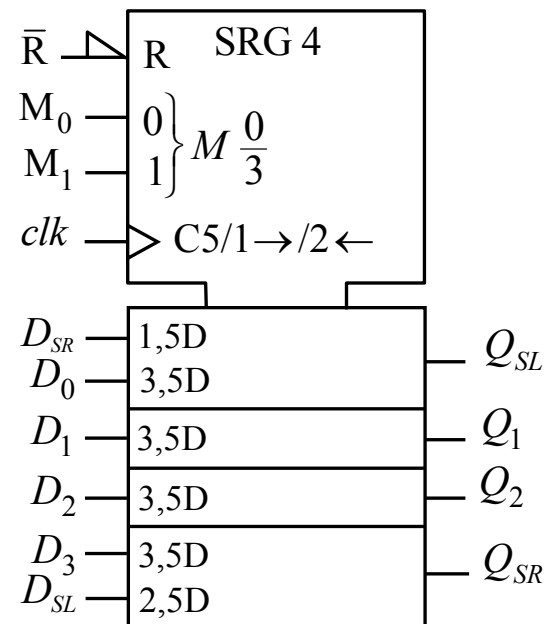
$M = 0$: entrada serie y desplazamiento hacia la derecha (hacia abajo) (s)

Reset: asíncrono



✓ Registros universales (*universal register*): pueden realizar cualquiera de las funciones analizadas anteriormente.

Existen unos registros denominados *bidireccionales* que disponen de dos entradas serie con las que se puede desplazar la información por su interior en ambos sentidos. Algunos registros universales son bidireccionales, otros no.

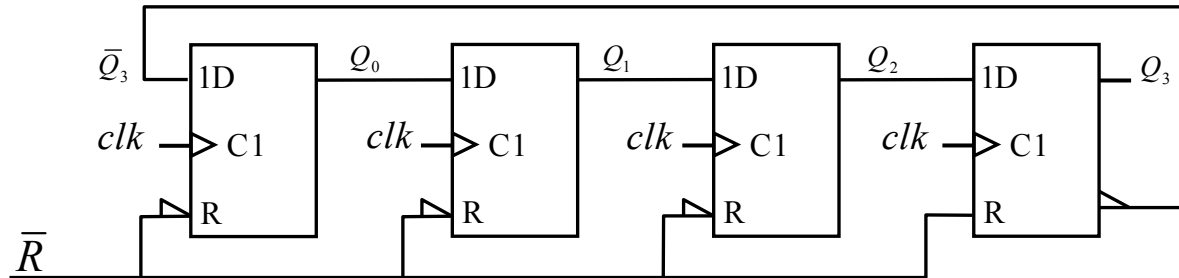


Aplicaciones de los registros:

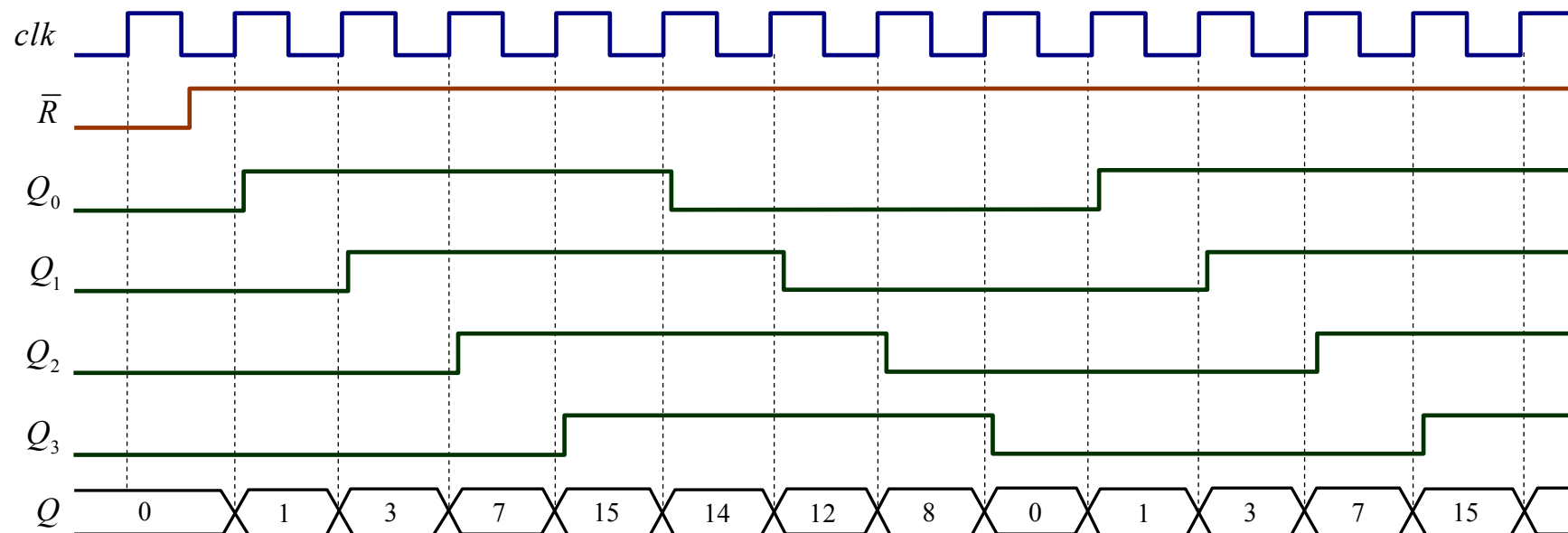
- ✓ Guardar datos temporalmente
- ✓ Cambiar el formato de la información (serie ↔ paralelo)
- ✓ Implementación de contadores que cuentan en códigos especiales
- ✓ etc.

no

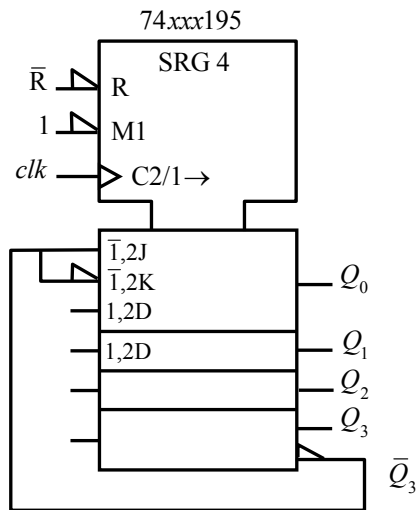
- Contador de 4 bits en código *Johnson* implementado con *flip-flops D*



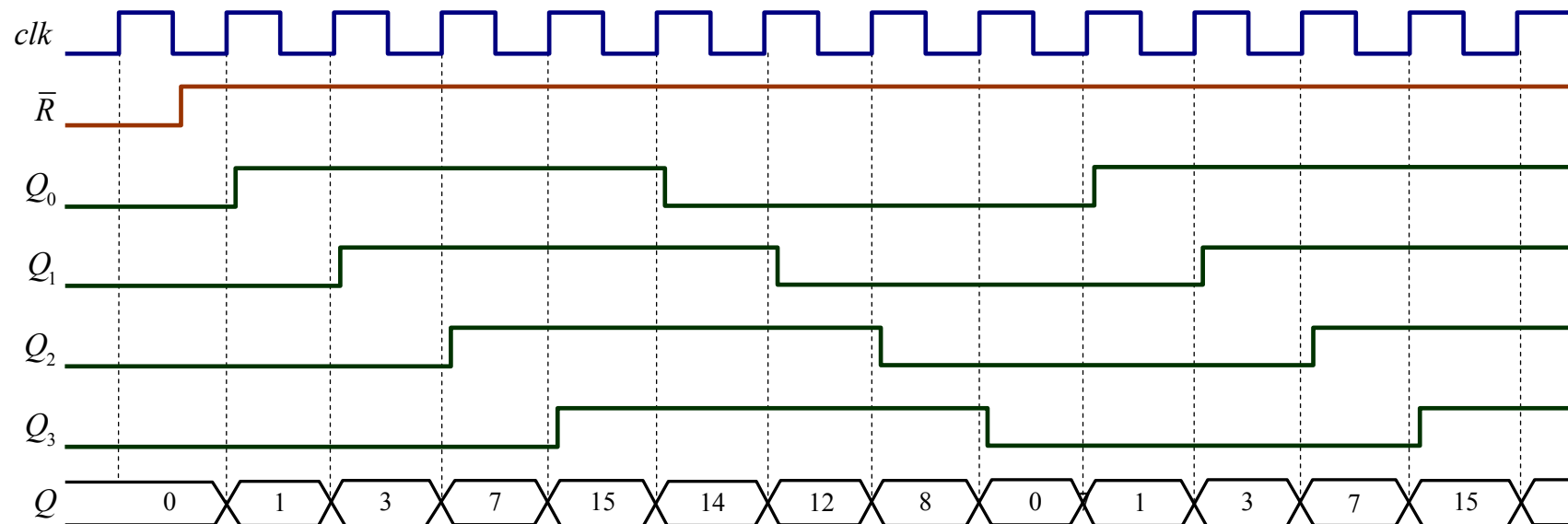
Q_3	Q_2	Q_1	Q_0
0	0	0	0
0	0	0	1
0	0	1	1
0	1	1	1
1	1	1	1
1	1	1	0
1	1	0	0
1	0	0	0



- Contador de 4 bits en código *Johnson* implementado con un registro 74xxx195

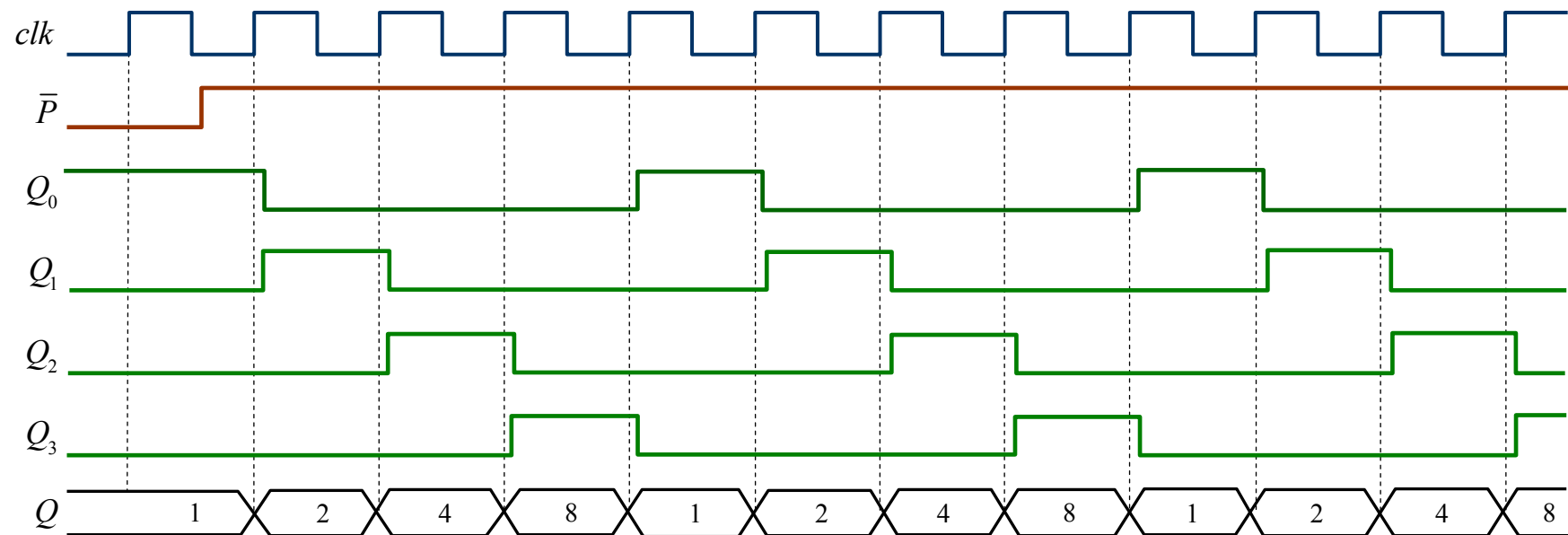
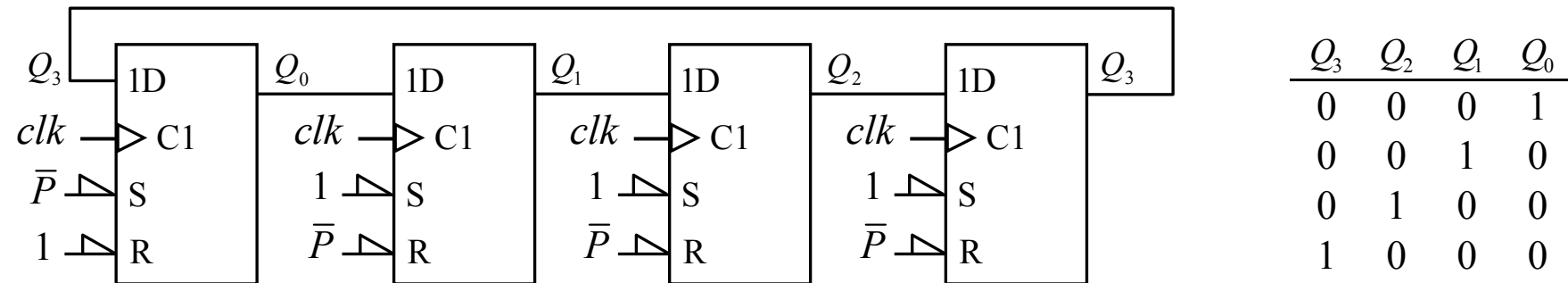


Q_3	Q_2	Q_1	Q_0
0	0	0	0
0	0	0	1
0	0	1	1
0	1	1	1
1	1	1	1
1	1	1	0
1	1	0	0
1	0	0	0



no

- Contador de 4 bits en código *Anillo* implementado con *flip-flops D*



- Contador de 4 bits en código *Anillo* implementado con un registro 74xxx195

no

