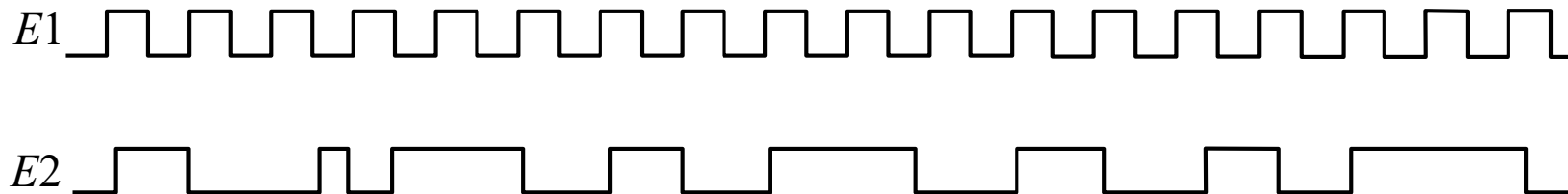
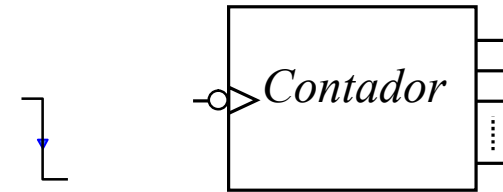
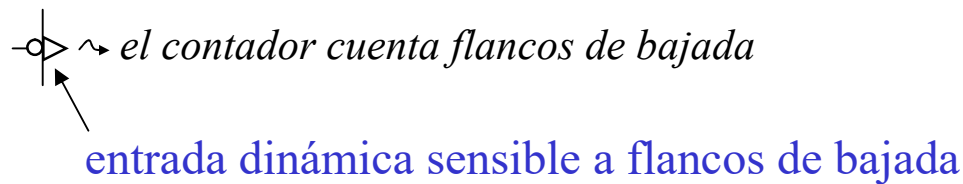
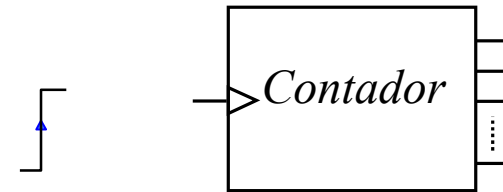
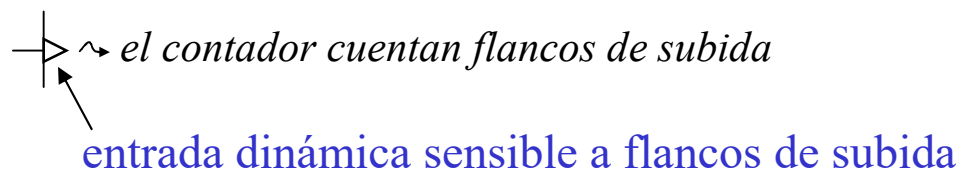


## Tema 5 Parte II

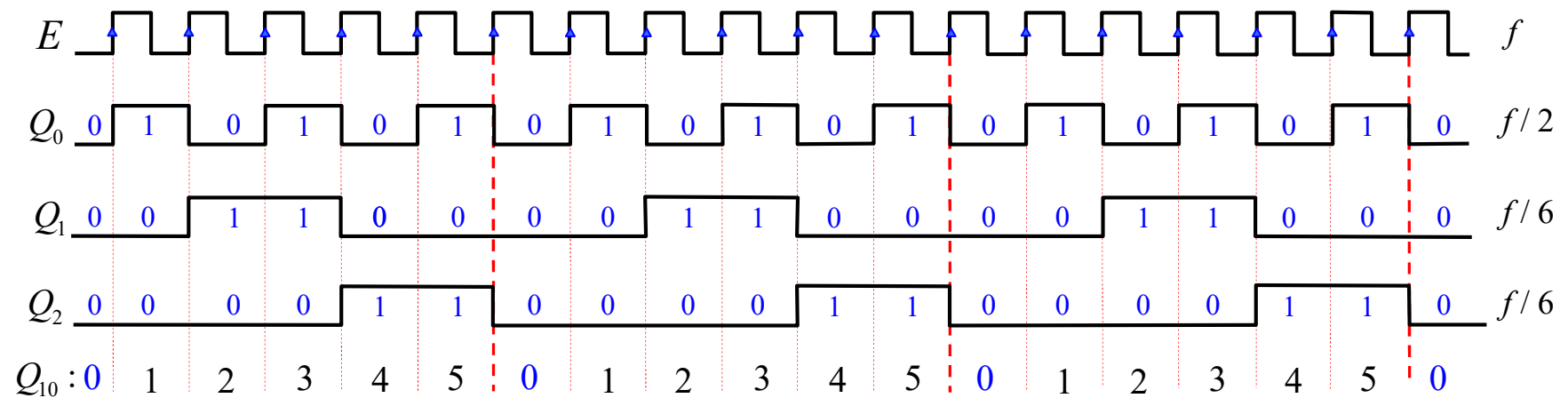
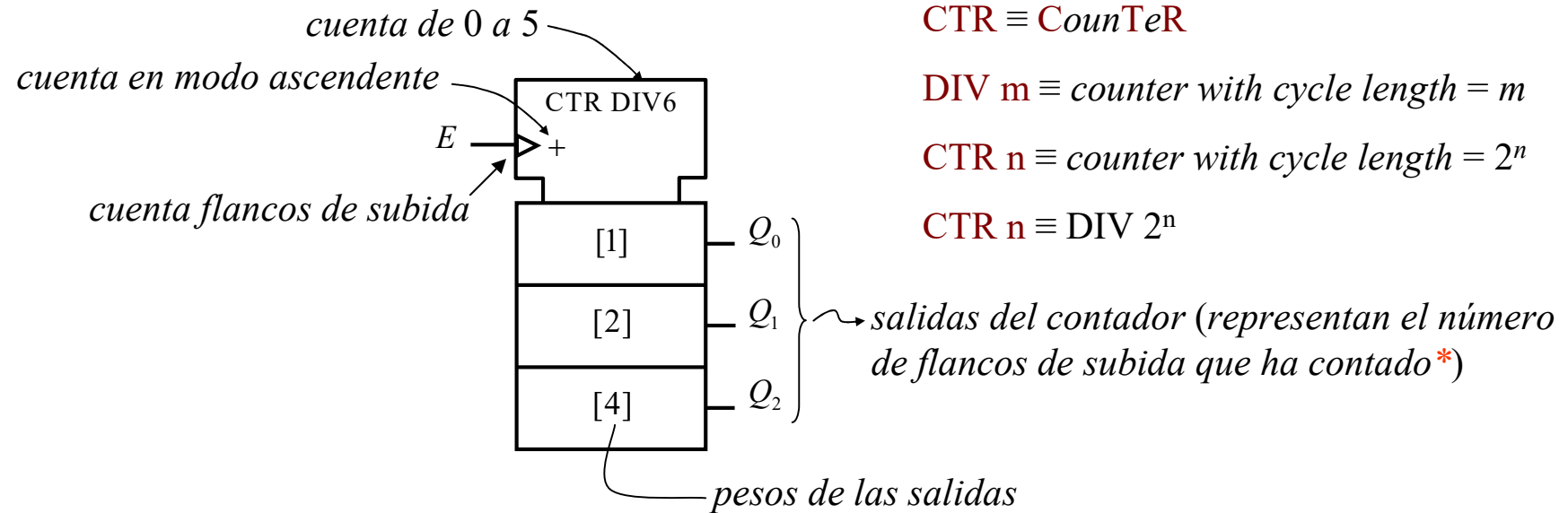
*Bloques funcionales secuenciales (síncronos)*  $\left\{ \begin{array}{l} \textit{Contadores} \\ \textit{Registros} \end{array} \right.$

## Contadores

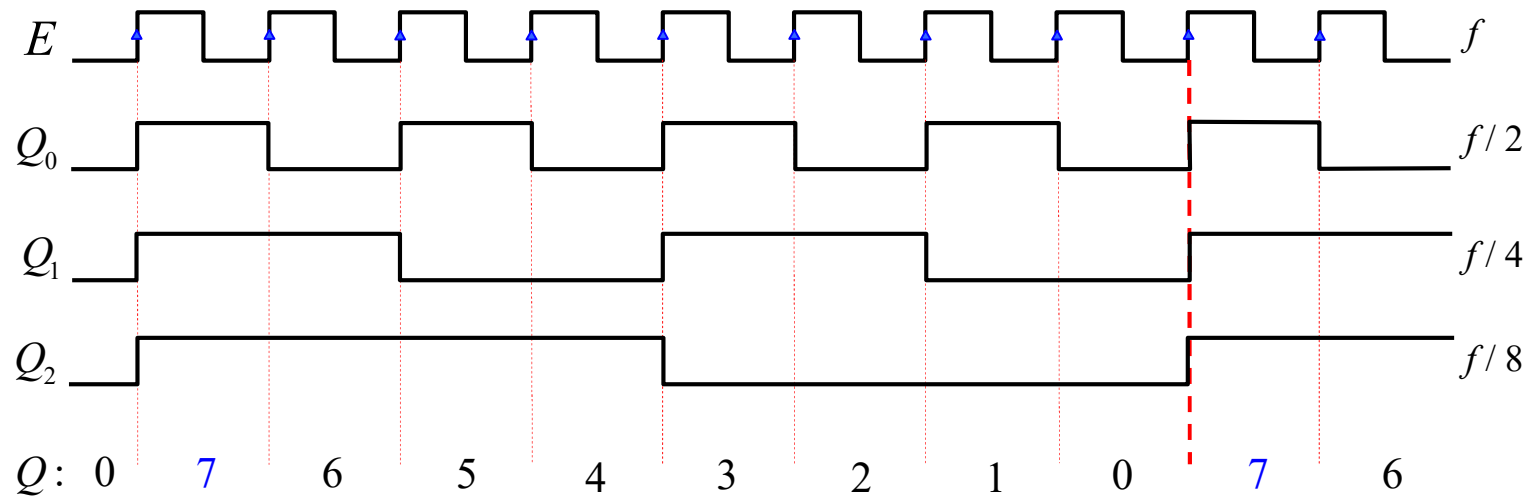
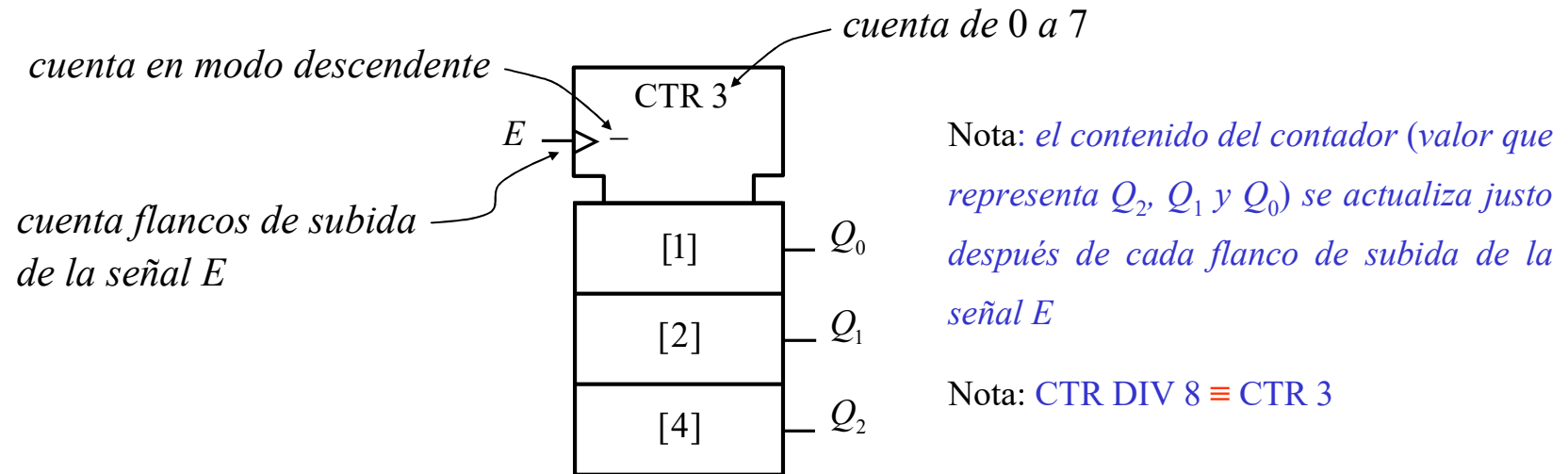
- Un *contador* es un circuito *secuencial síncrono* que cuenta en un código binario los *flancos* de subida o bien de bajada (pero no ambos tipos de flancos) que presenta la señal que se aplica a su entrada dinámica. Cuando se habla de un contador, si no se indica nada, hay que asumir que cuenta en binario.



Ejemplo: contador de los flancos de subida de la señal  $E$ , en modo ascendente (+)

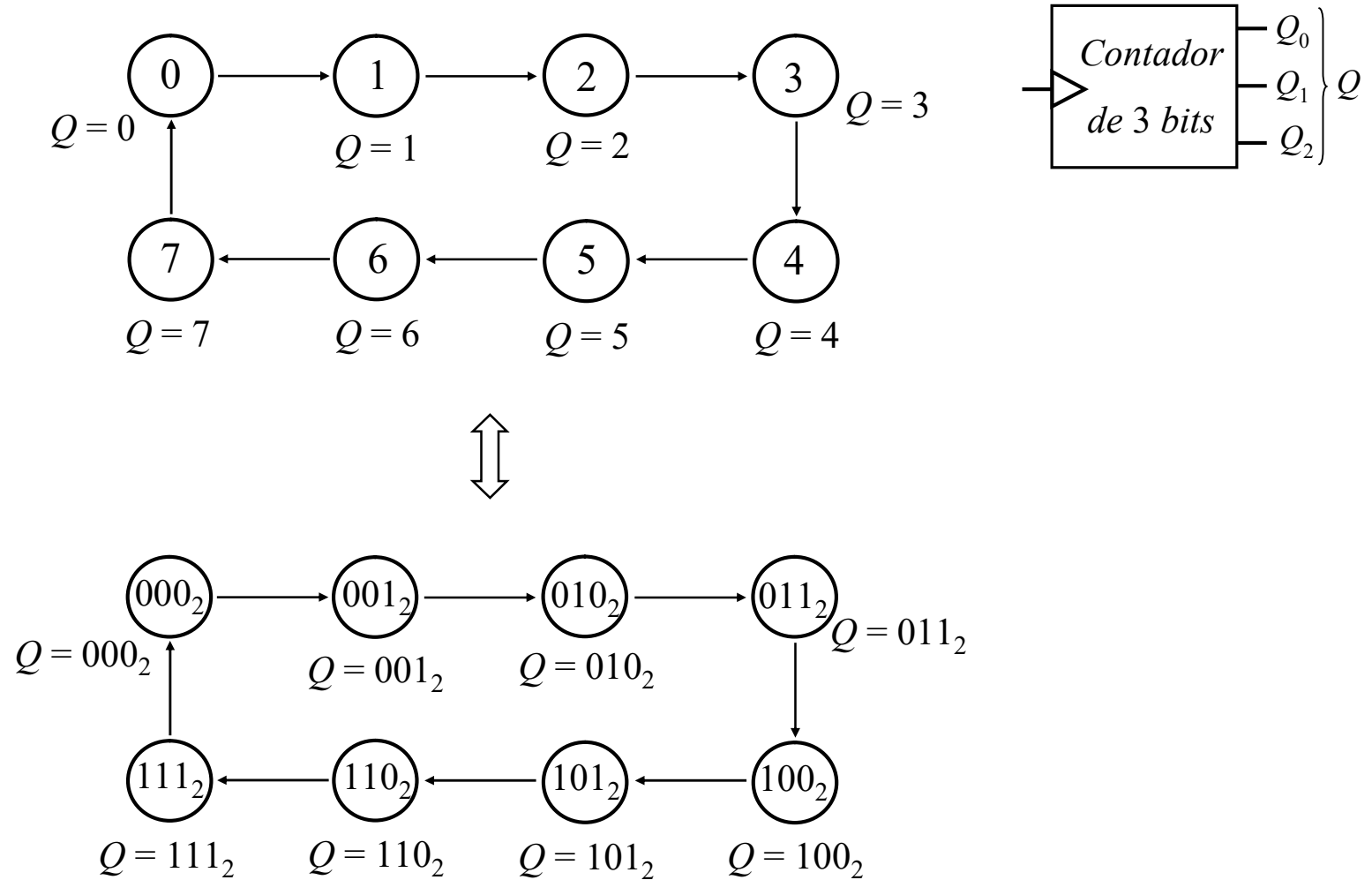


*Ejemplo: contador de los flancos de subida de  $E$ , en modo descendente (-) de 3 bits*



- Un contador de  $n$  bits, con modo de conteo ascendente, puede contar desde 0 hasta  $2^n-1$  (flancos). Es decir, su *secuencia de conteo* va desde 0 hasta  $2^n-1$ .
- Un contador  $\text{DIV}m$ , con modo de conteo ascendente, puede contar desde 0 hasta  $m-1$  (flancos). Es decir, su *secuencia de conteo* va desde 0 hasta  $m-1$ .
- Cuando el contenido de un contador alcanza el valor más alto de su *secuencia de conteo* (modo de conteo ascendente), con el siguiente flanco a contar pasa al valor más bajo de su secuencia de conteo (se produce un *desbordamiento*)
- Los contadores comerciales cuentan en binario (hay contadores que cuentan en otros códigos binarios)

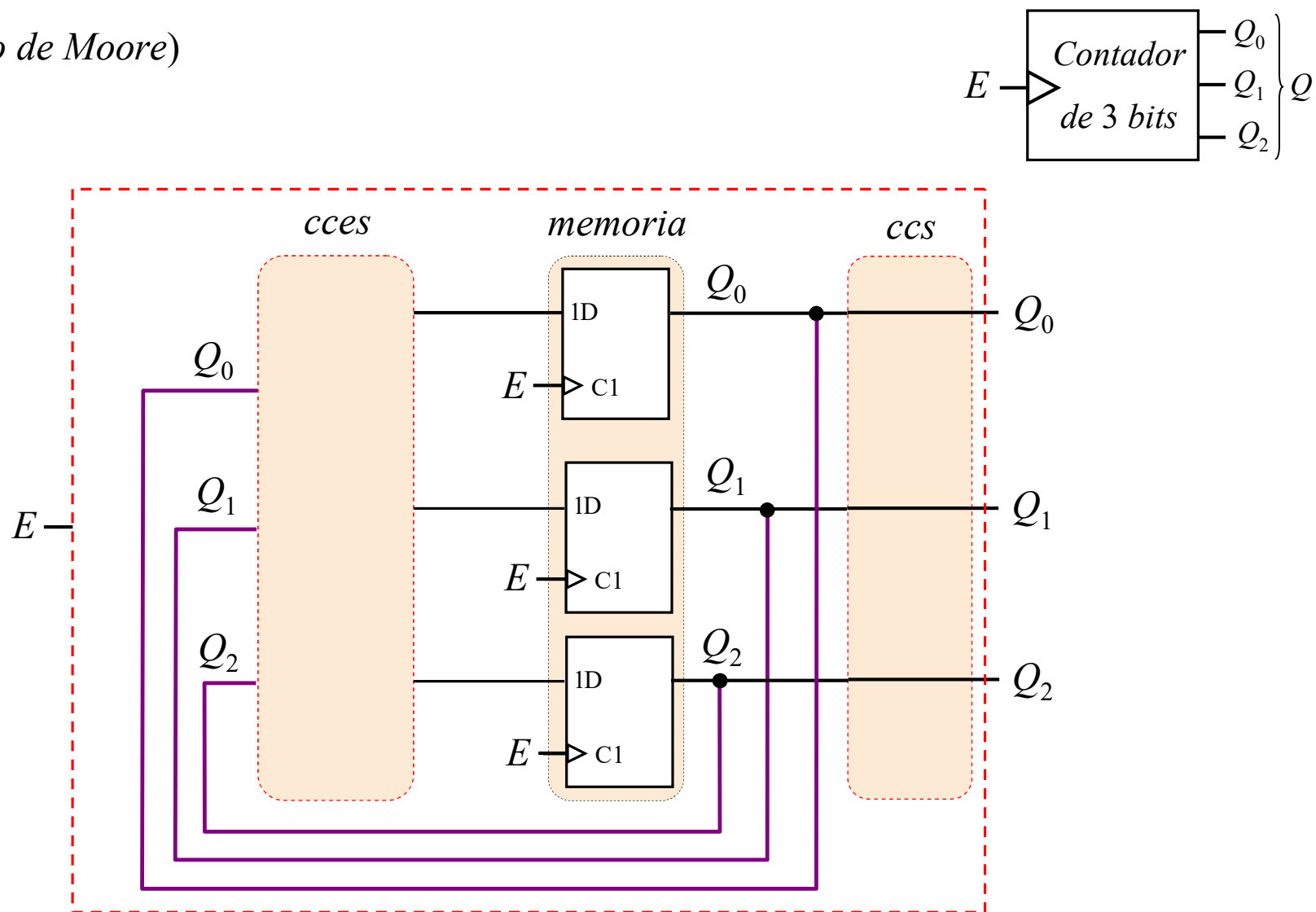
*Ejemplo: diagrama de estados de un contador de módulo 8 que cuenta, en binario, flancos de subida, en modo **ascendente***



Del diagrama de estados (*modelo de Moore*) de la página anterior se deduce lo siguiente:

- El diagrama de estados corresponde a un contador de 3 bits que cuenta flancos en binario
- El sistema secuencial descrito por el diagrama de estados de la página anterior no tiene entradas “propiamente dichas”. La señal a la que se le cuentan los flancos (de subida o bien de bajada) es la señal que se aplica en el terminal de sincronismo de los *flip-flops D* que forman el bloque de memoria (ver página siguiente).
- En el diagrama de estados de la página anterior, el valor de las salidas del contador coincide en todo momento con el valor de las variables de estado  $\Rightarrow$  dicho contador no tiene un *circuito combinacional de salida (ccs)*.
- Las transiciones entre estados en el diagrama de estados de la página anterior son saltos incondicionales.

(modelo de Moore)



*Overflow*: en la literatura técnica se dice que “*se produce un desbordamiento de un contador que cuenta en binario, en modo ascendente*”, cuando su *contenido* pasa del mayor valor de su secuencia de conteo al menor valor.

Así, por ejemplo, cuando el contenido de un contador de 3 bits, que cuenta en modo ascendente, pasa de 7 a 0 se dice que se ha producido un desbordamiento (*the counter overflows*)

Se define el *módulo* de un contador como el número de valores que tiene su *secuencia de conteo*.

*Ejemplos:*

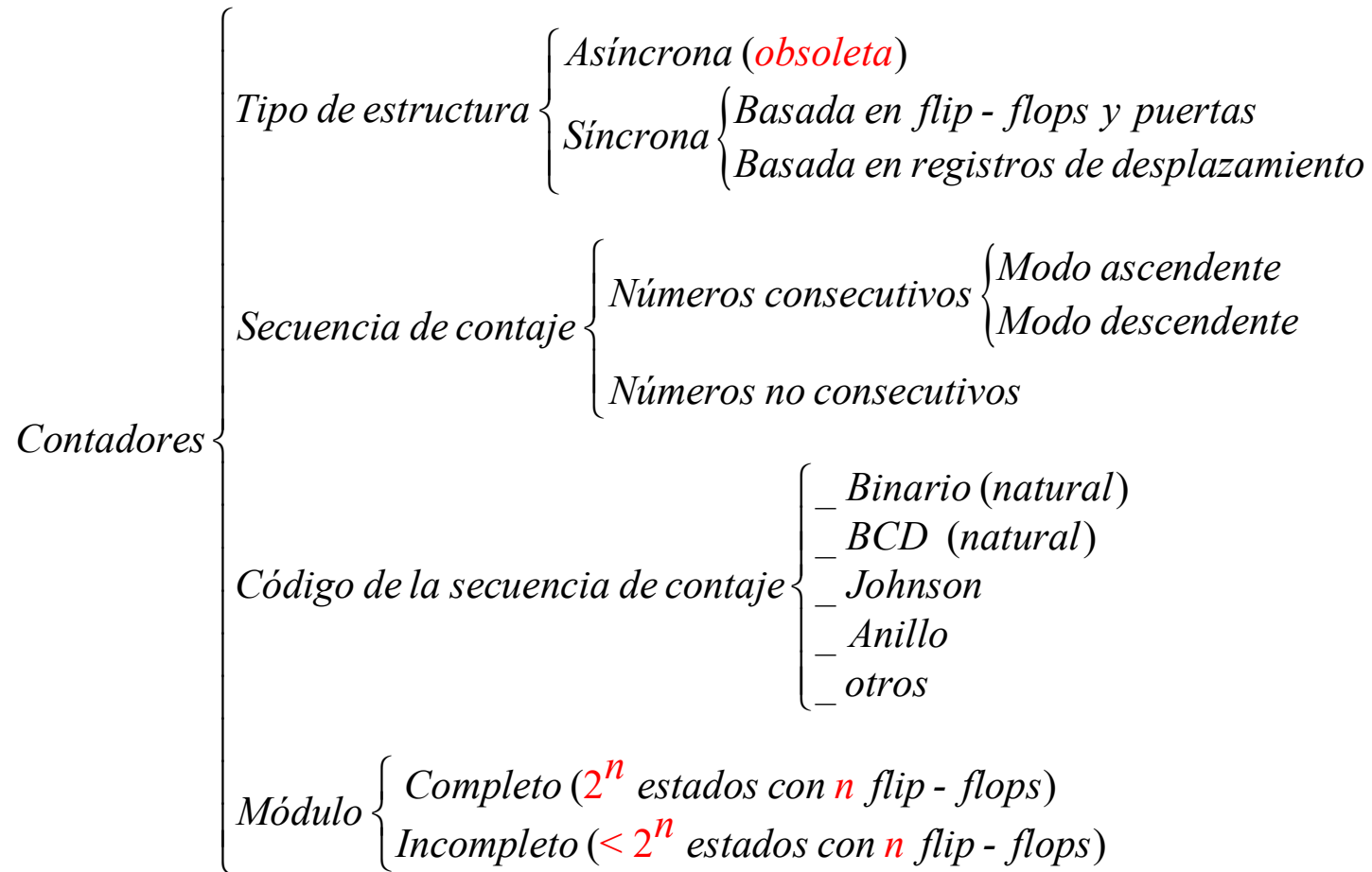
0 - 1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 9 - 0 - 1 - 2 - 3 -  $\dots$  (módulo 10, DIV10)

5 - 4 - 3 - 2 - 1 - 0 - 5 - 4 - 3 -  $\dots$  (módulo 6, DIV6)

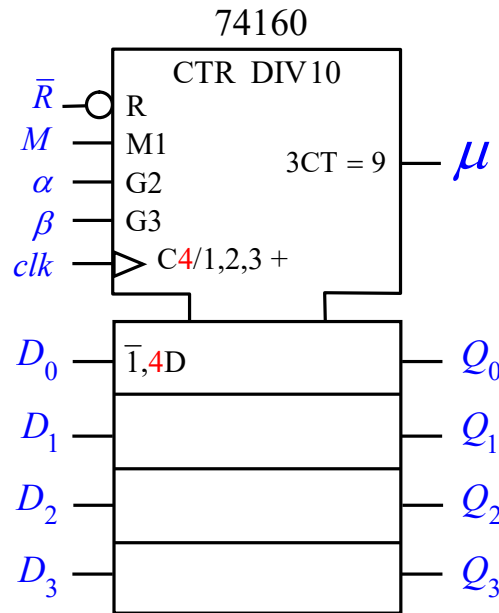
3 - 11 - 15 - 9 - 0 - 5 - 3 - 11 - 15 -  $\dots$  (módulo 6)

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

## Clasificación de los contadores:



## Contadores comerciales (cuentan en binario):



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 del modo de funcionamiento:

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

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

$\alpha$ : entrada de inhibición del contaje

$\beta$ : 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

$CT \equiv content \equiv contenido$

Características del contador 74xxx160:

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

\_ *reset: asíncrono, activo a nivel bajo*

\_ *modo de conteo: ascendente*

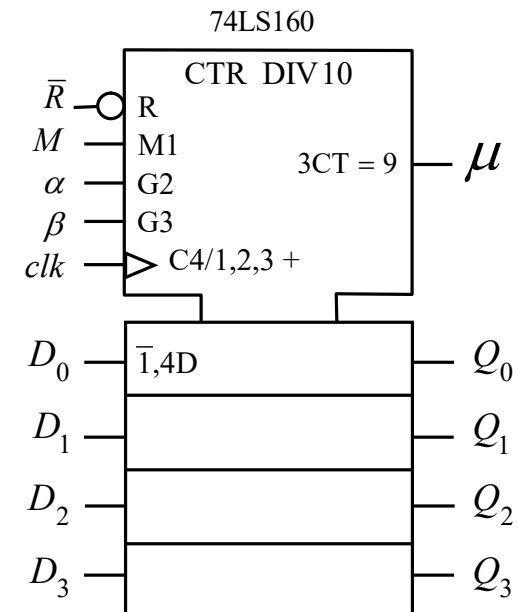
\_ *secuencia de conteo: 0 – 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 9*

\_ *cuenta en binario*

Nota: en el momento en el que se activa la entrada de *reset* el *contenido* del contador se pone a *cero* ( $Q_3, Q_2, Q_1, Q_0 = 0000_2$ )

**Importante:** en los sistemas secuenciales la función de *reset* siempre es prioritaria frente a cualquier otra función (*conteo, carga en paralelo, etc.*). Esto es así por diseño (los sistemas secuenciales se diseñan para que esto sea siempre así).

**Importante:** cuando una salida (o salidas) cambia de valor siempre lo hace con un cierto retraso  $t_p$  con respecto al instante en el que cambia de valor la entrada que provoca el cambio de la salida (o salidas).



Características del contador 74xxx162:

\_ *módulo: 10 (contador de décadas)*

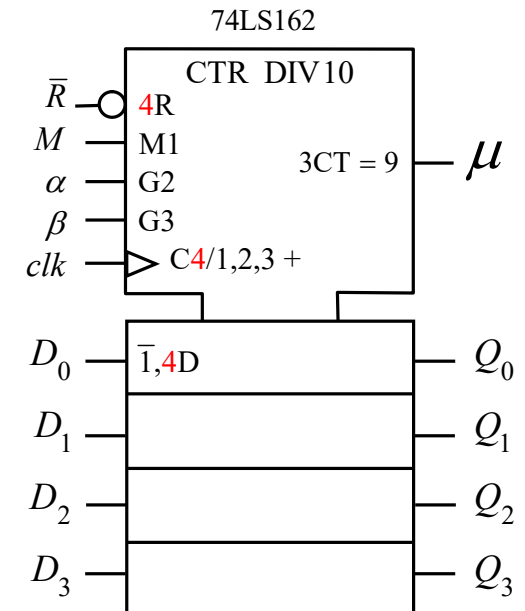
\_ *reset: **síncrono**, activo a nivel bajo*

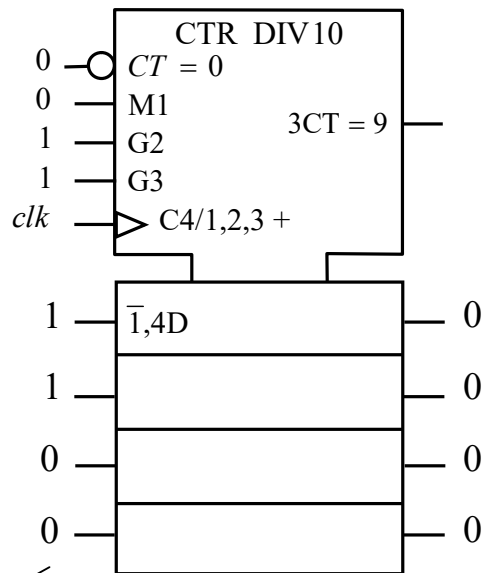
\_ *modo de conteo: ascendente*

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

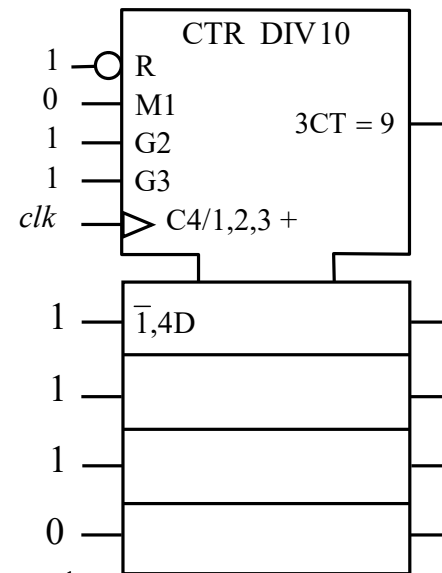
\_ *cuenta en binario*

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

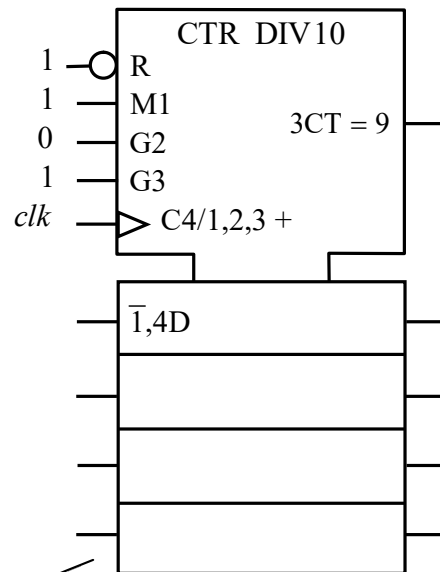




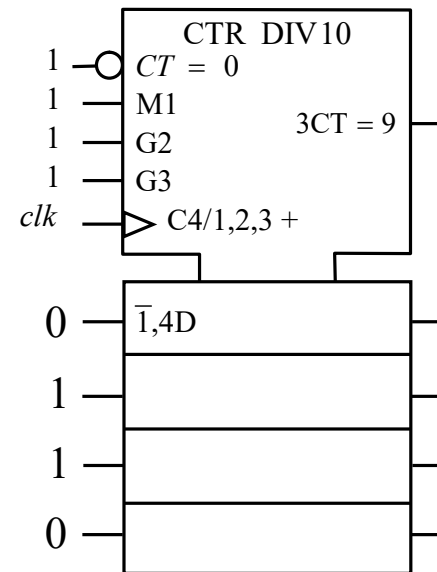
está seleccionado el modo de carga en paralelo, pero el contenido del contador está a cero porque la entrada de *reset* está activa.



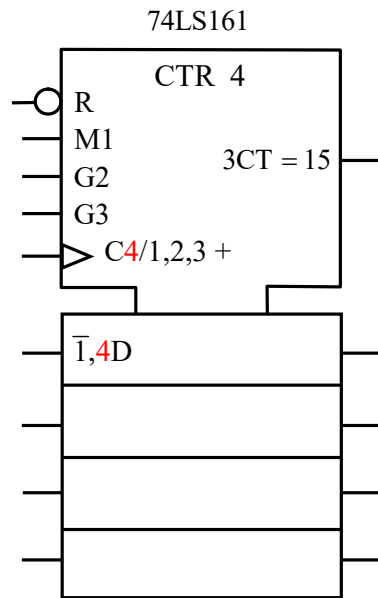
está seleccionado el modo de carga en paralelo. Cuando la señal *clk* describa un flanco de subida, el contenido del contador pasará a ser  $0111_2$ .



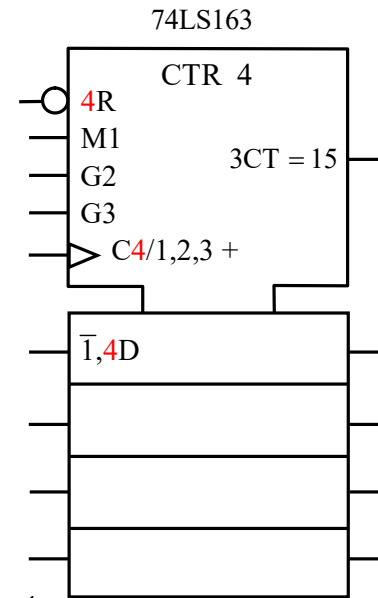
Está seleccionado el modo de *contaje*, pero dicho modo de funcionamiento está *inhibido* ( $G2 = 0$ )



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.

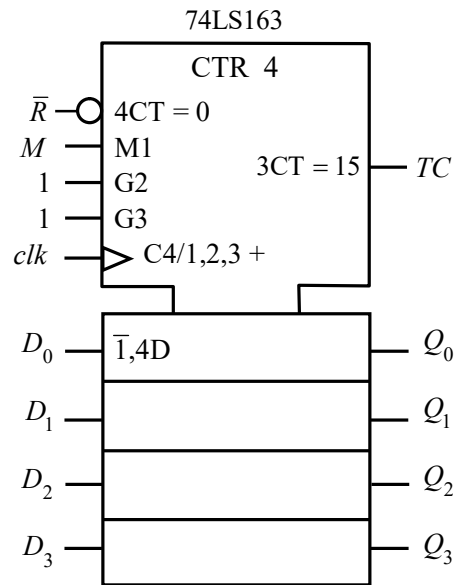


- \_ Contador de 4 bits (módulo igual a  $2^4 = 16$ )
- \_ Reset *asíncrono*, activo a nivel bajo.
- \_ Cuenta desde 0 hasta 15
- (CTR 4  $\equiv$  CTR DIV 16)



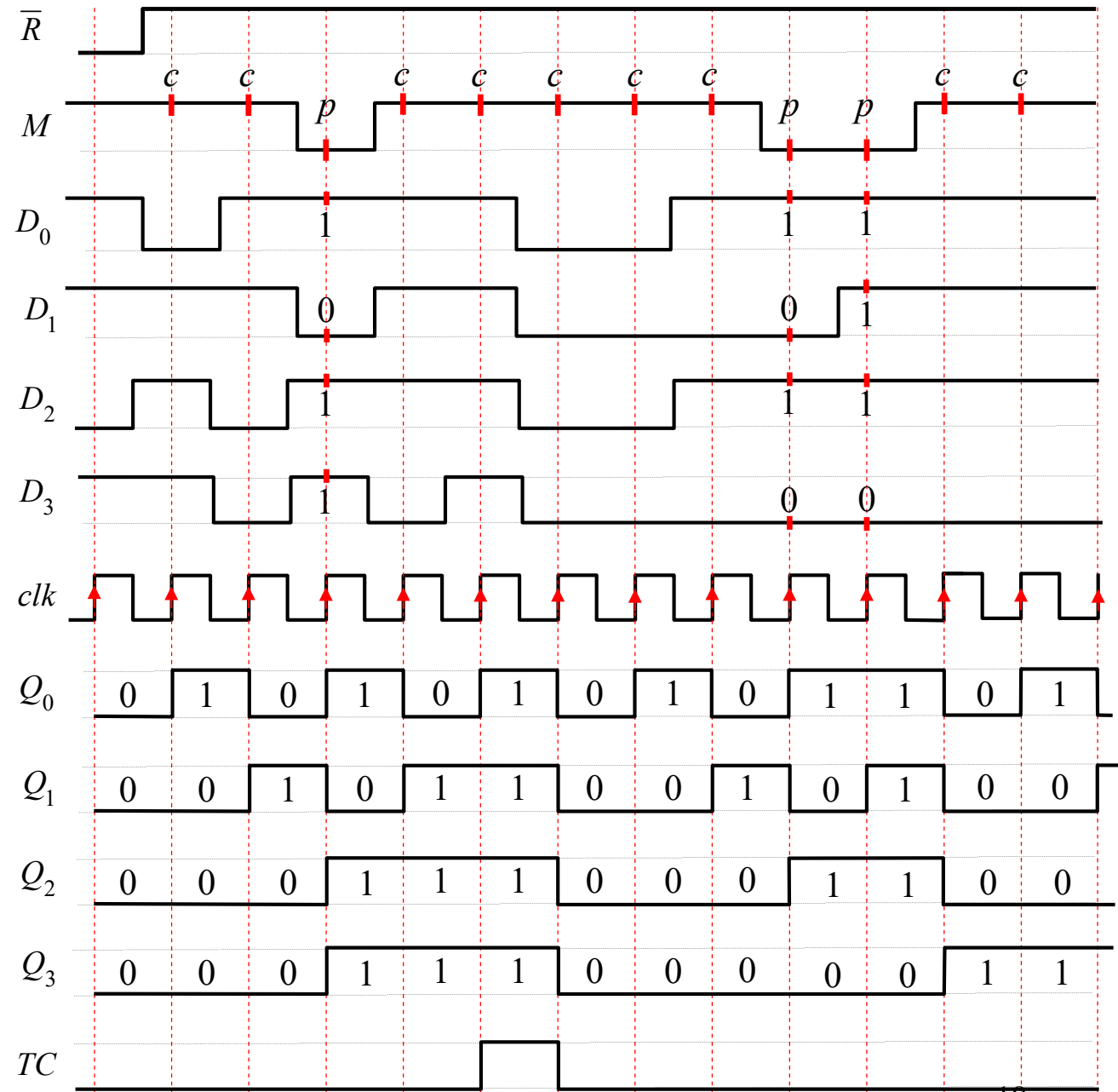
- \_ Contador de 4 bits (módulo igual a 16)
- \_ Reset *síncrono*, activo a nivel bajo.
- \_ Cuenta desde 0 hasta 15

## Ejemplo (cronograma)

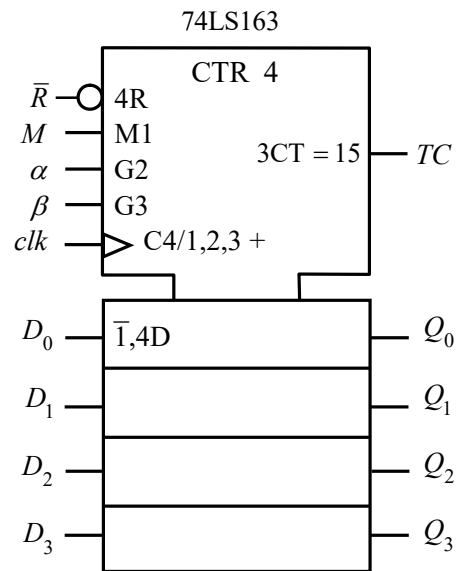


$M = 1$  : cuenta (*c*)

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

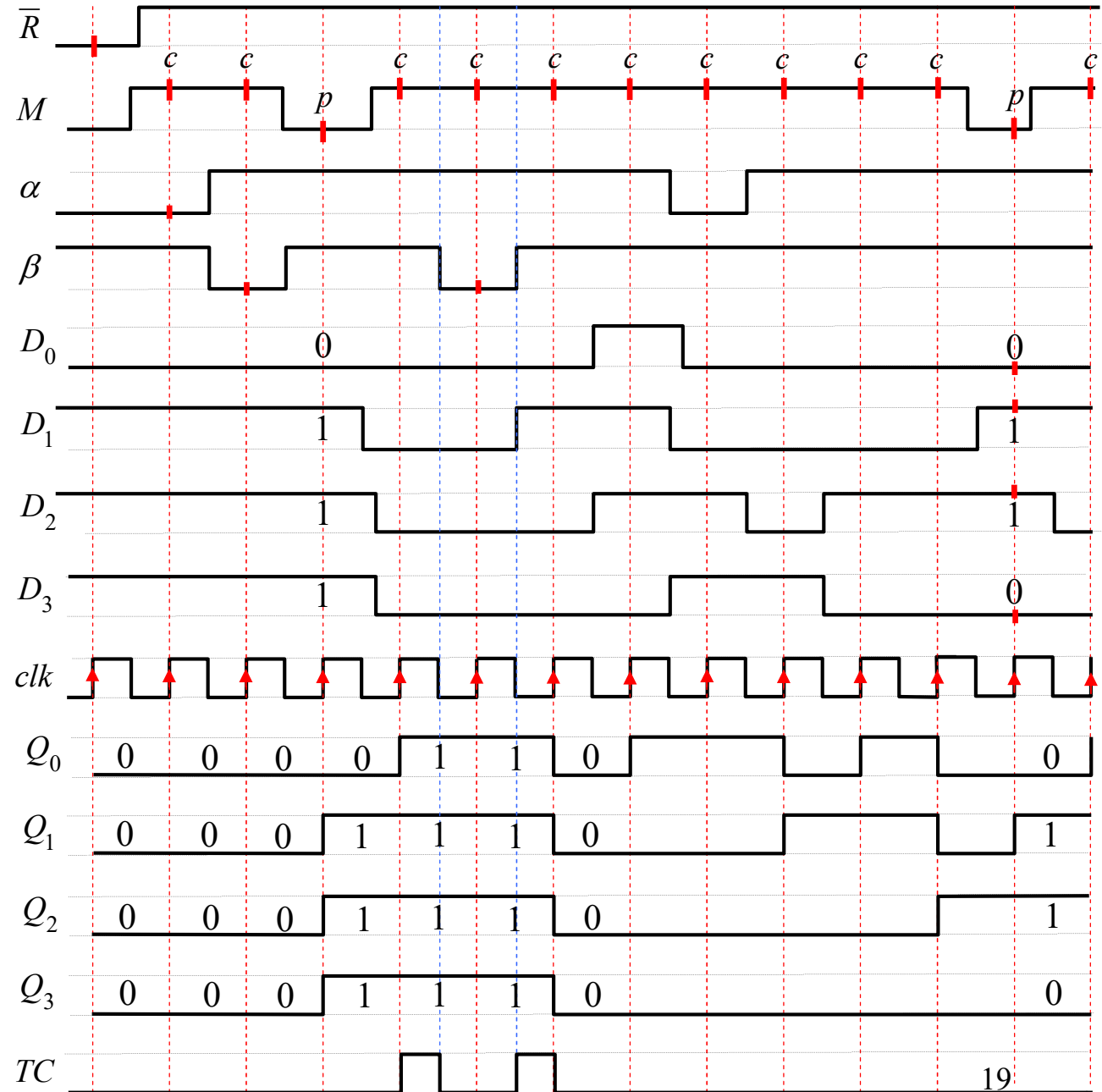


# Ejemplo (cronograma)



$M = 1$  : cuenta (c)

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

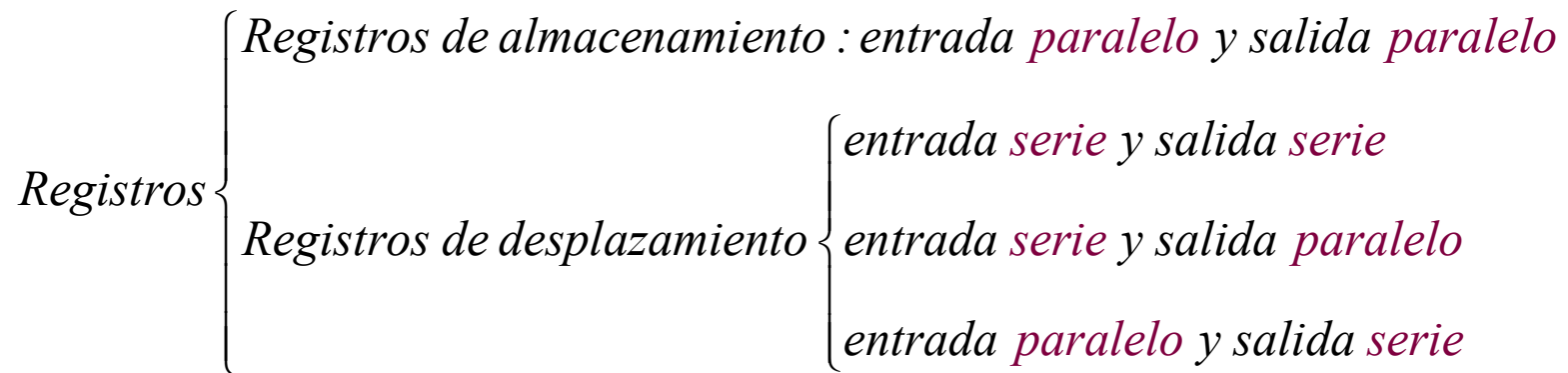


## Registros

### Definición y características:

- 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*  $D$  adecuadamente interconectados entre sí.
- Los registros no tienen una secuencia de estados característica.

- Los registros se pueden clasificar, de acuerdo con los formatos de entrada y de salida de los bits, de la siguiente manera:



*Importante:* en los sistemas secuenciales la función de *reset* siempre es prioritaria frente a cualquier otra función (*contaje, carga en paralelo, etc.*). Esto es así por diseño (los sistemas secuenciales se diseñan para que esto sea así).

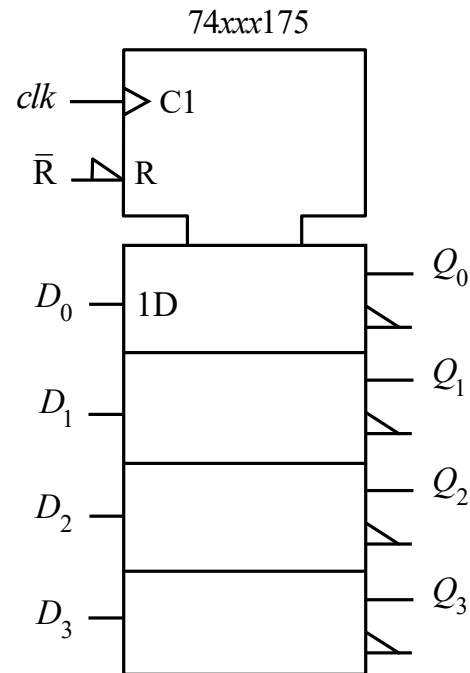
✓ Registro paralelo de  $n$  bits

Registro de almacenamiento de  $n$  bits

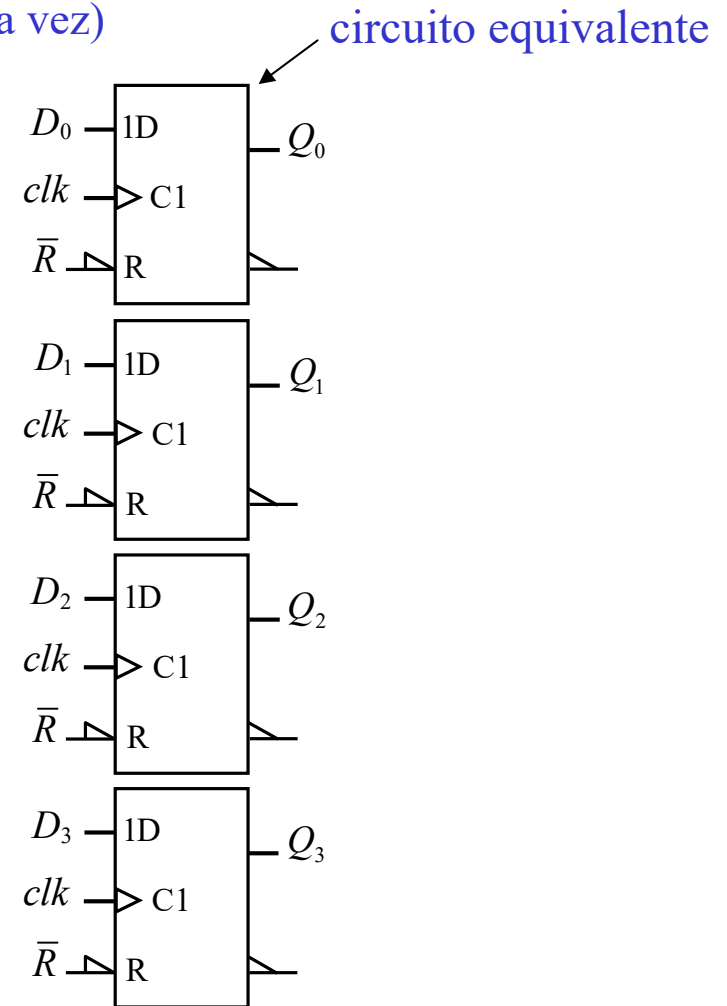
entrada: *paralelo* (los  $n$  bits se guardan a la vez)

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

*Ejemplo*: registro paralelo de 4 bits

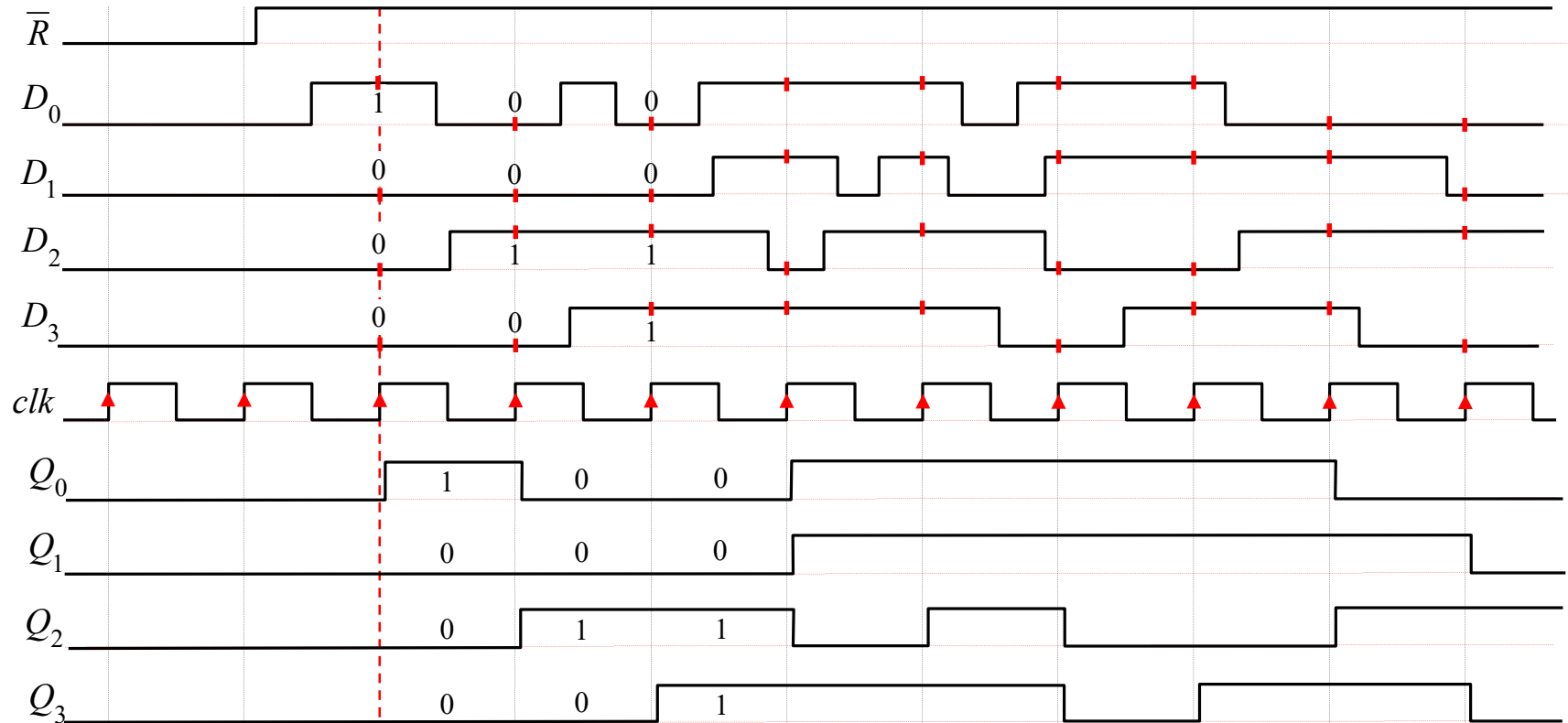
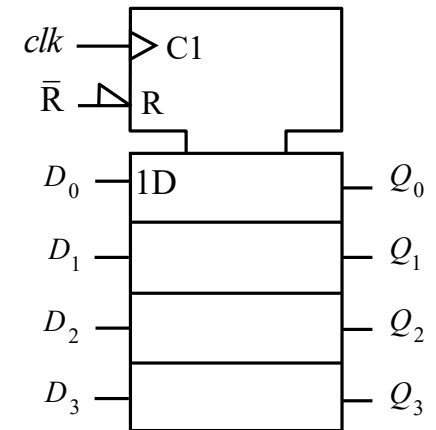


≡



### Ejemplo (cronograma)

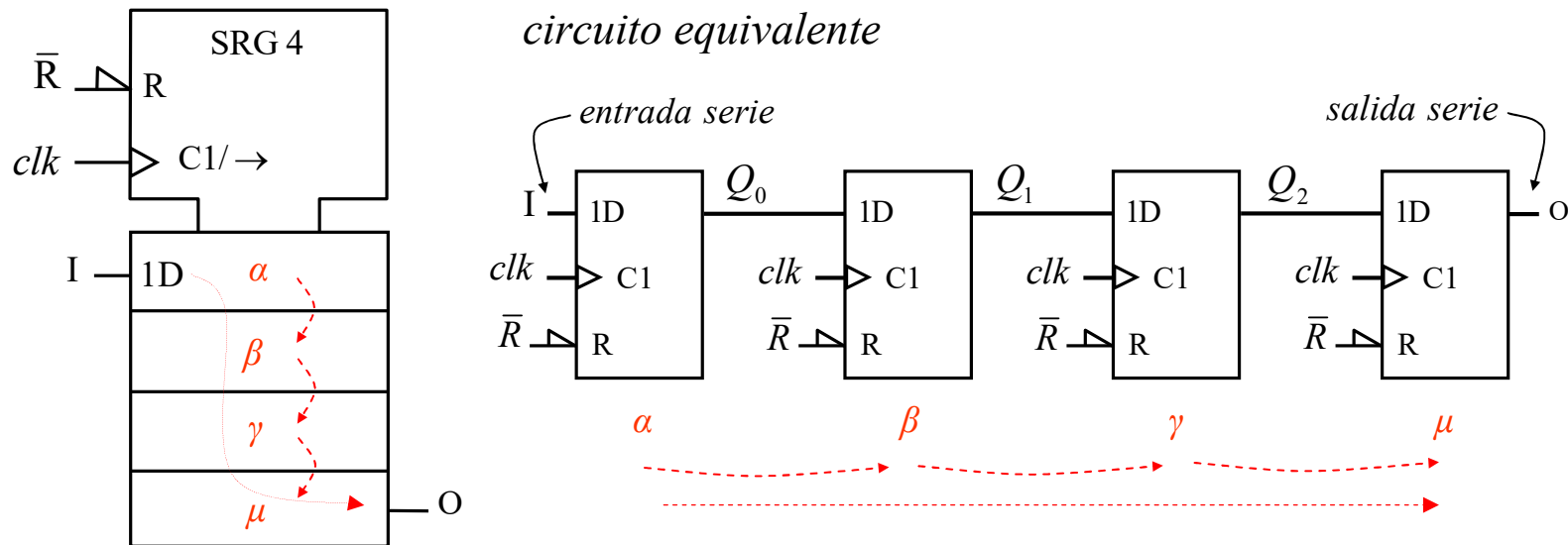
Nota: cada vez que la señal  $clk$  describe un flanco de subida, en el registro se guardan los valores que haya en las entradas  $D_3$ ,  $D_2$ ,  $D_1$  y  $D_0$ . A partir del momento en el que se activa la entrada de reset ( $R$ ) en el registro se guarda el valor 0000. El contenido del registro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



✓ 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)

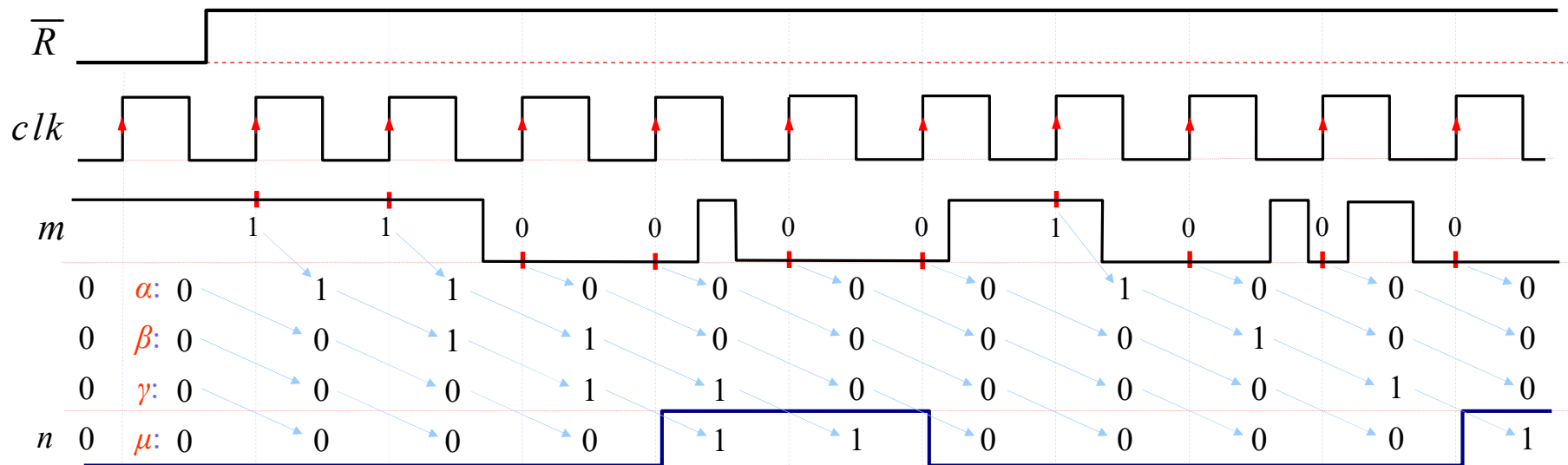
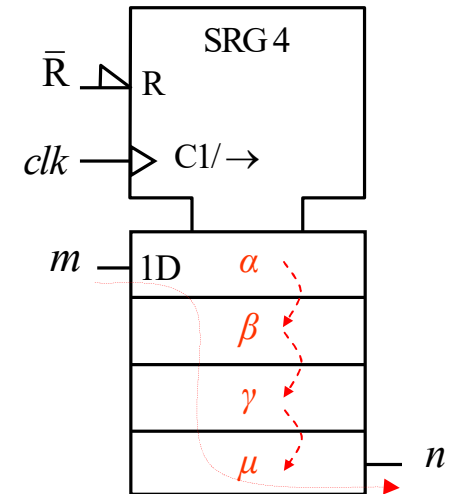


*Importante:* un flip-flop  $D$  guarda el valor que hay en su entrada de información ( $D$ ) durante un flanco de sincronismo un poco después de que se produzca dicho flanco de sincronismo. Esto hace que, por ejemplo, después de que la señal  $clk$  describa un flanco de sincronismo el flip-flop  $\mu$  guarde el valor que guardaba el flip-flop  $\gamma$  antes de producirse dicho flanco de sincronismo y que el flip-flop  $\gamma$  guarde el valor que guardaba el flip-flop  $\beta$  antes del flanco de sincronismo.

## 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  $\alpha$ . 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.

**Importante:** un flip-flop  $D$  guarda el valor que hay en su entrada  $D$  durante un flanco de sincronismo un poco un poco después de que se produzcan dicho flanco de sincronismo.

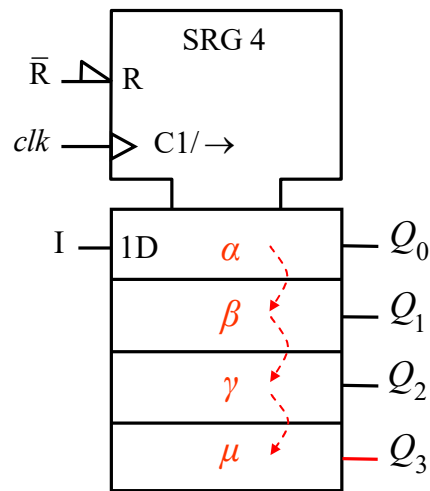


Nota: después de que ocurra un flanco de sincronismo, el flip-flop  $\beta$  guarda el valor que guardaba el flip-flop  $\alpha$  antes de producirse dicho flanco de sincronismo.

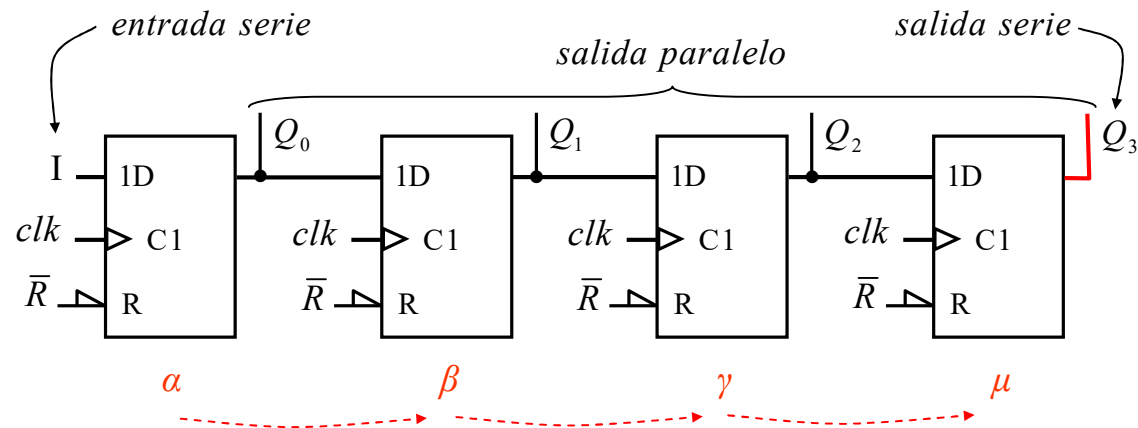
✓ Registro de desplazamiento (*shift register*) de  $n$  bits

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)

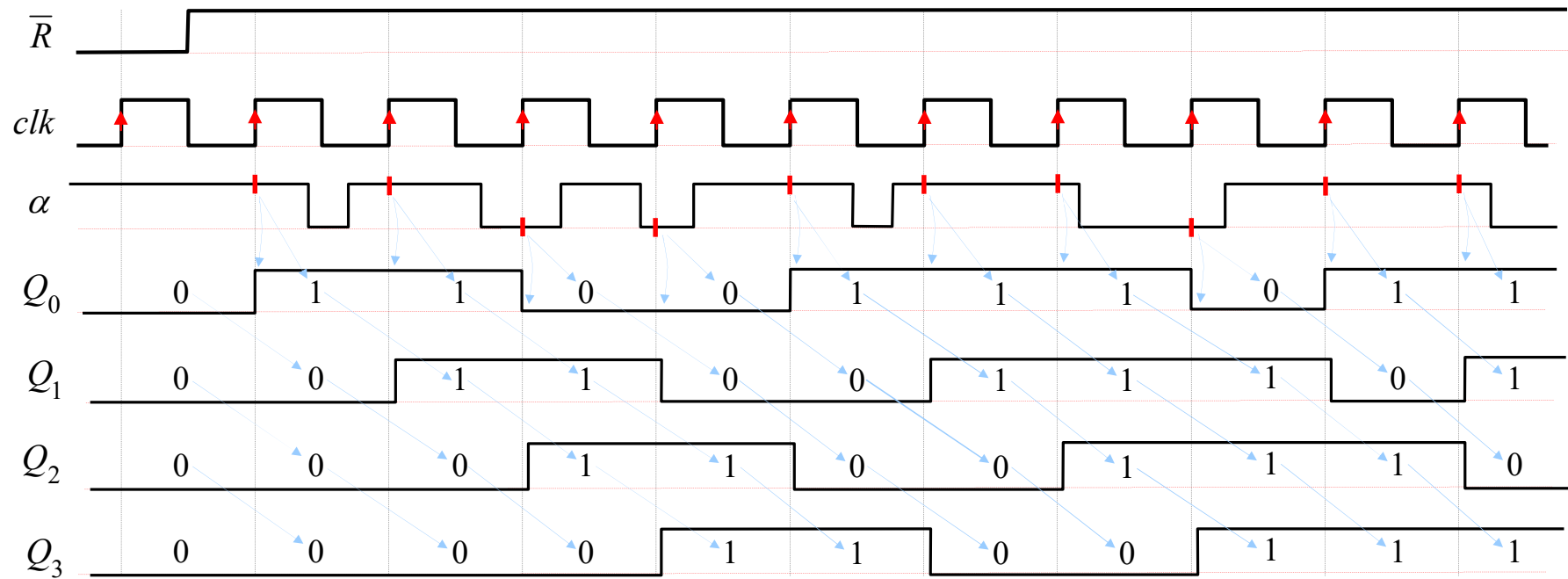
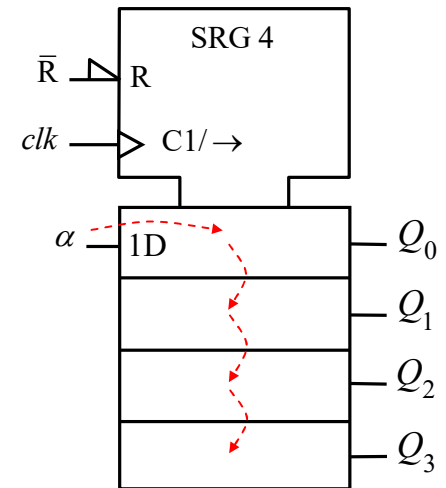


*circuito equivalente*



*Recuerda:* un flip-flop  $D$  guarda un poco después de que se produzca un flanco de sincronismo el valor que hay en su entrada de información ( $D$ ) durante el flanco de sincronismo. Esto hace que, por ejemplo, después de que la señal  $clk$  describa un flanco de sincronismo el flip-flop  $\mu$  guarde el valor que guardaba el flip-flop  $\gamma$  antes de producirse dicho flanco de sincronismo y que el flip-flop  $\gamma$  guarde el valor que guardaba el flip-flop  $\beta$  antes del flanco de sincronismo.

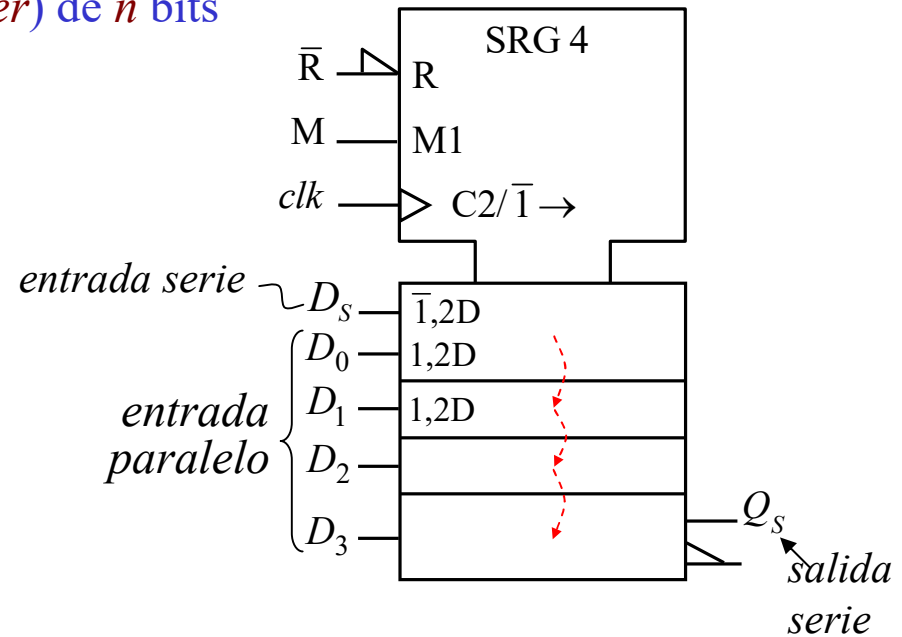
## Ejemplo (cronograma)



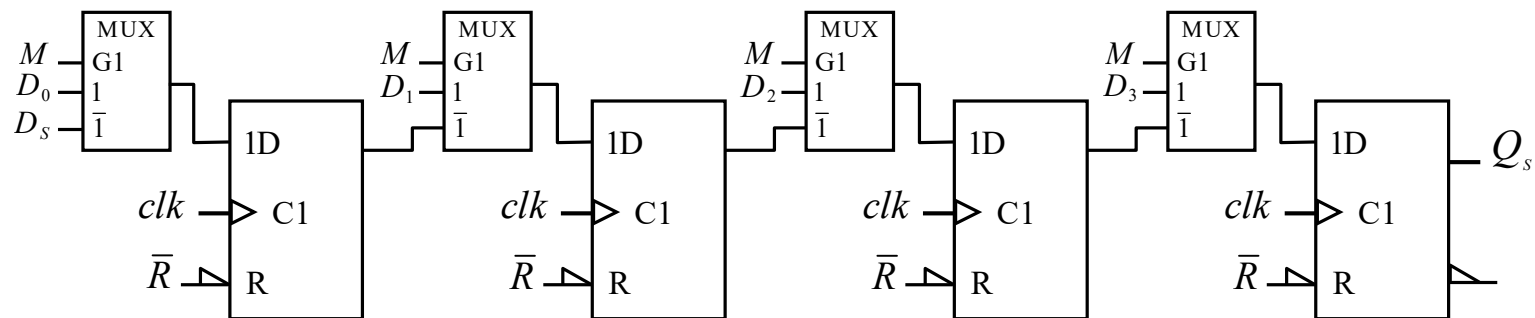
✓ Registro de desplazamiento (*shift register*) de  $n$  bits

entrada: *paralelo* o *serie*

salida: *serie*



Circuito equivalente:



$M = 1$  : entrada en *paralelo*

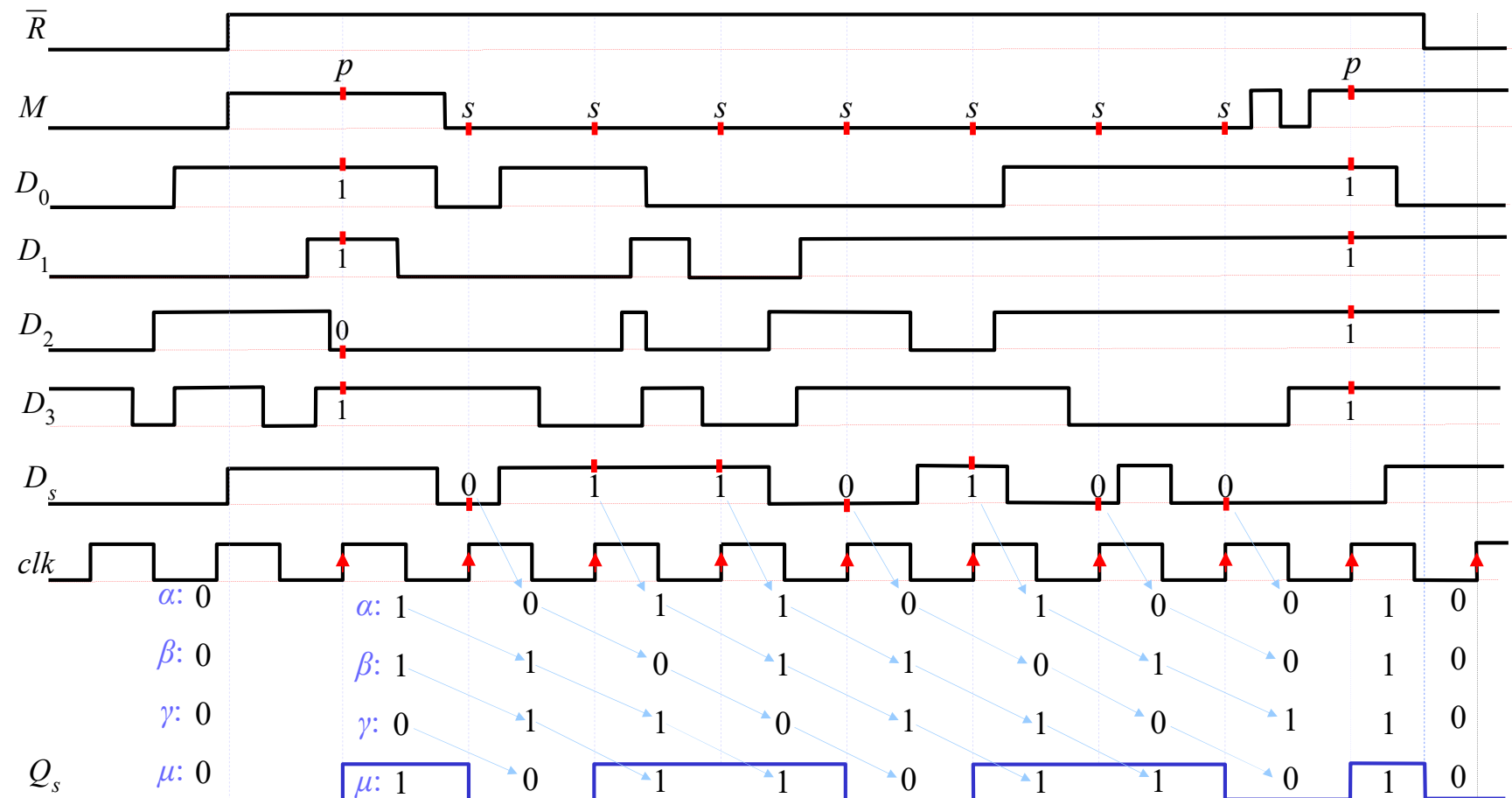
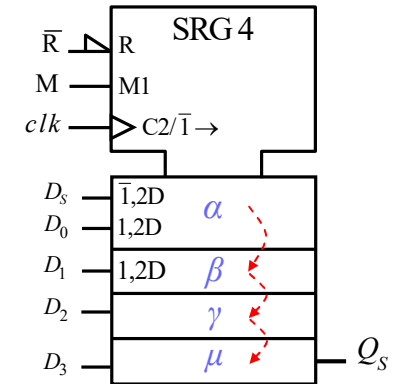
$M = 0$  : entrada *serie* y desplazamiento hacia la derecha (hacia abajo en el símbolo)

## Ejemplo (cronograma)

$M = 1$  : entrada paralelo ( $D_0 D_1 D_2 D_3$ ) (*p*)

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

Reset: asíncrono



### Aplicaciones de los registros:

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

## *Tema 6: Memorias semiconductoras*