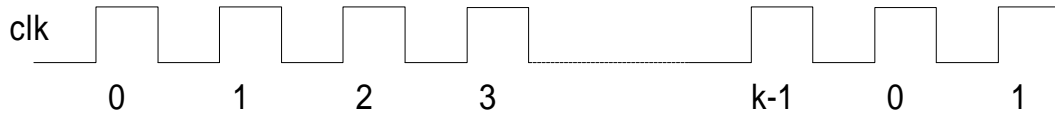


TEMA 8

SUBSISTEMAS SECUENCIALES

1. CONTADORES

Un contador módulo k es un circuito digital capaz de contar k sucesos distintos. Un contador módulo-k, tiene k estados de cuentas distintos, desde el 0, hasta el k-1.

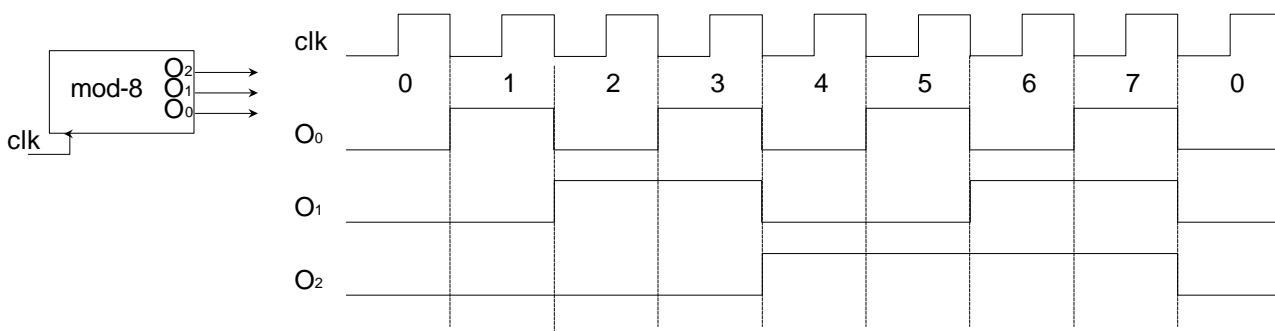


Un contador módulo-k, tiene k estados, y estos se pueden implementar con n biestables, existiendo una relación entre k y n determinada por la expresión:

$$2^{n-1} \leq k \leq 2^n$$

Normalmente los contadores se construyen usando el menor número de biestables posibles y si estos tienen un módulo que es potencia de 2, suelen identificarse también como CONTADORES DE N BITS.

Si las salidas del contador son la codificación binaria del estado de cuenta y el número de estados es una potencia de dos, este suele denominarse divisor de frecuencia. En la siguiente figura se ha representado la salida de un contador módulo 8 (contador de 3 bits o divisor de frecuencia de 3 bits).



1.1 Contadores síncronos

Se puede determinar su estructura interna de dos formas distintas:

- 1) Se obtiene el diagrama de estados correspondiente y a partir de este el circuito.

- 2) Usando un procedimiento no sistemático, pero que se utiliza frecuentemente en la síntesis de circuitos complejos.

$$T_0 = 1$$

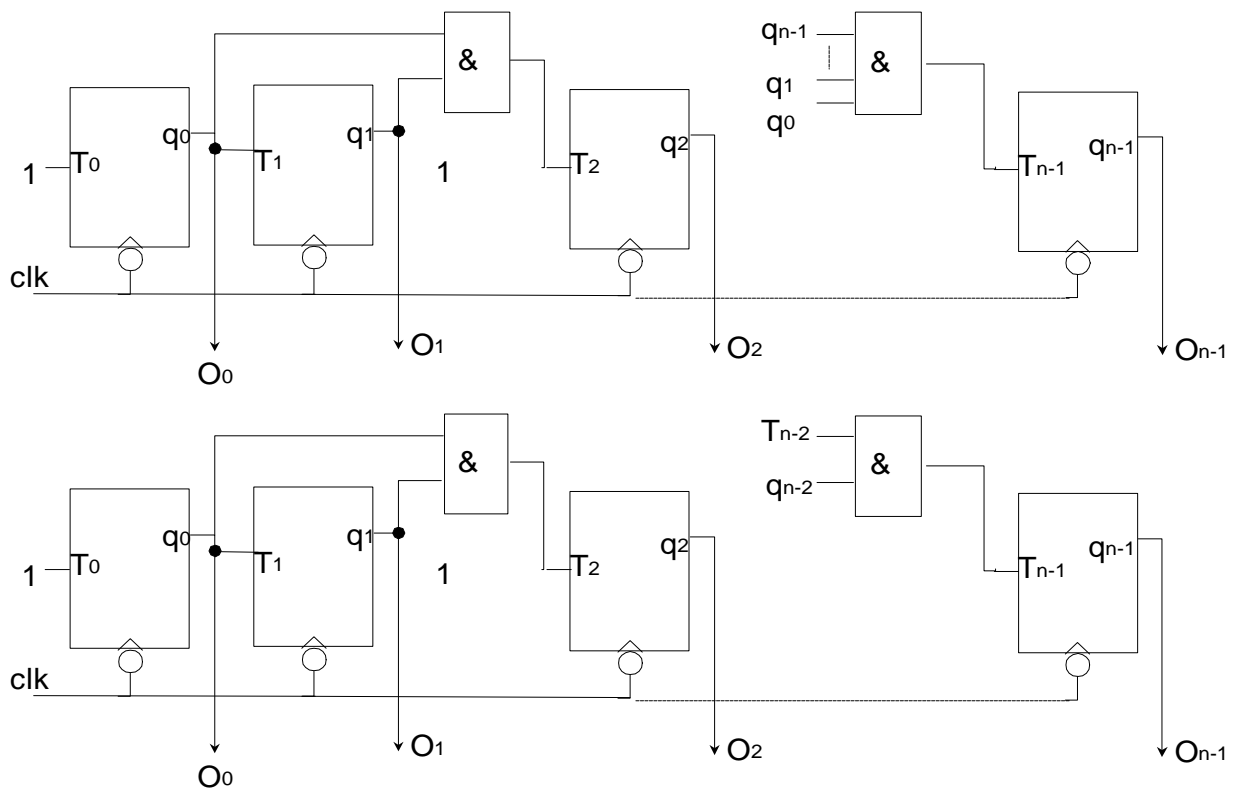
$$T_1 = q_0$$

$$T_2 = q_1 q_0$$

En general, si diseñamos un contador de n bits, la expresión de la entrada del biestable j (donde $j=1,2,3..n-1$) sería

$$T_j = q_0 q_1 \dots q_{j-1}$$

En las siguientes figuras se muestran dos posibles diseños para la estructura de un contador síncrono de n bits

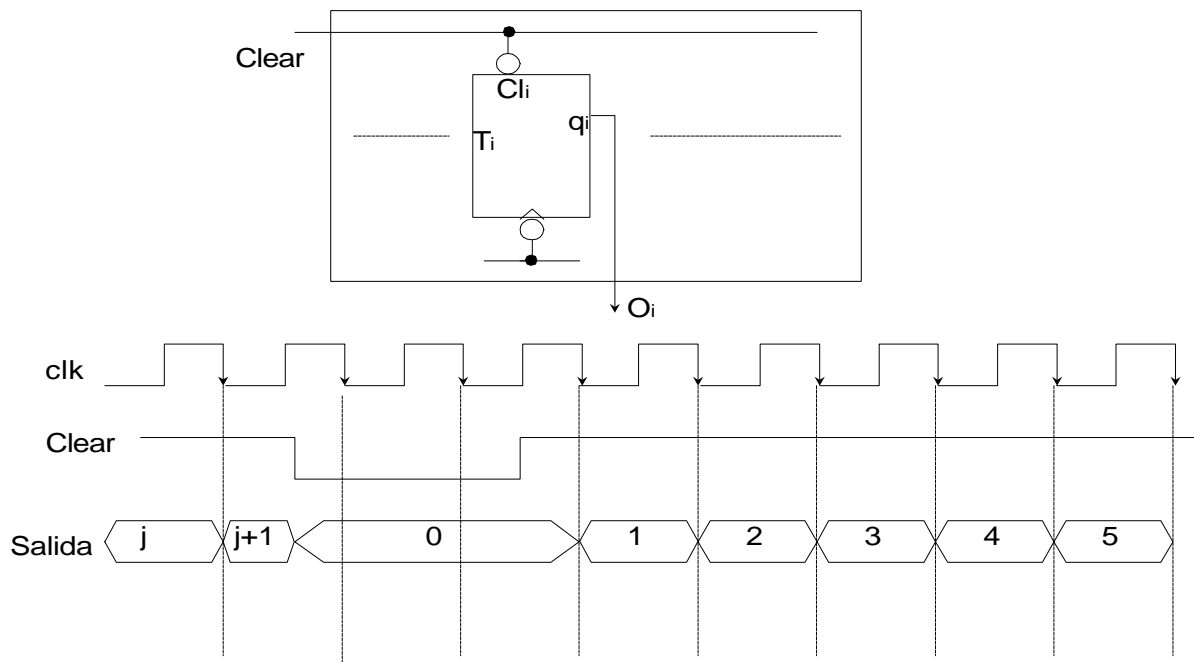


1.1.1 Reset(Clear) y Preset

La línea de Reset o Clear permite inicializar el contador a su estado de cuenta 0 cuando es activada. La línea de Preset permite inicializar el contador a su valor de cuenta más alto (aquel en que todos los biestables están a 1). Estas líneas pueden ser activas en alto o en bajo, y existen dos modalidades desde el punto de vista funcional: asíncrona y síncrona.

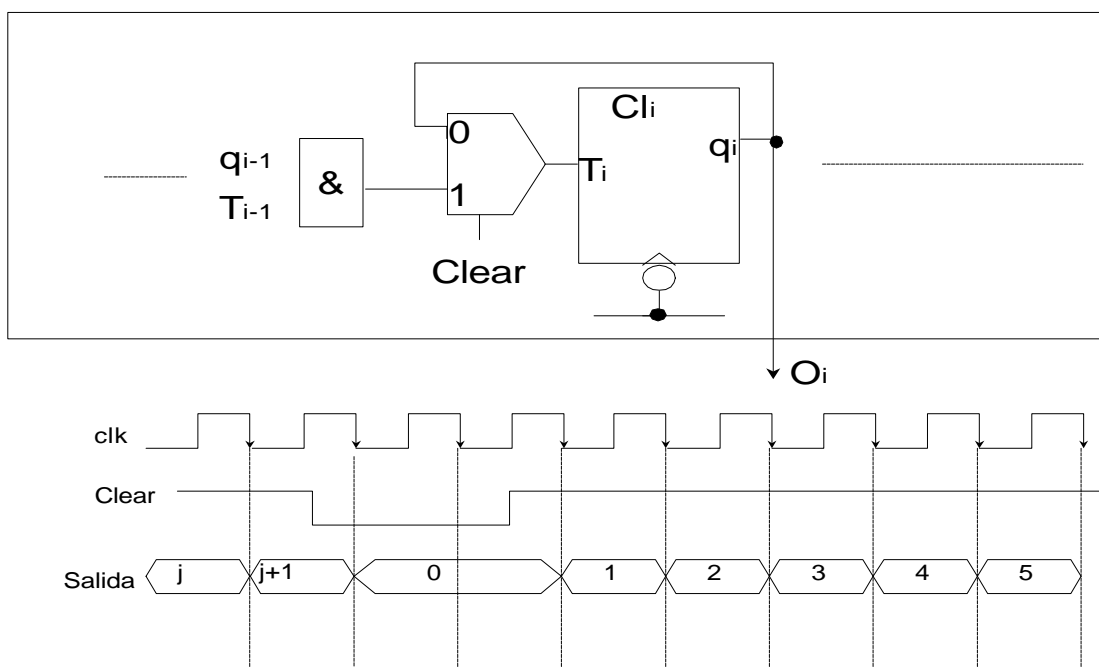
Clear o Reset asíncrono

De forma general diremos que todas las líneas de control que operen de modo asíncrono lo hacen usando las entradas asíncronas de los biestables.



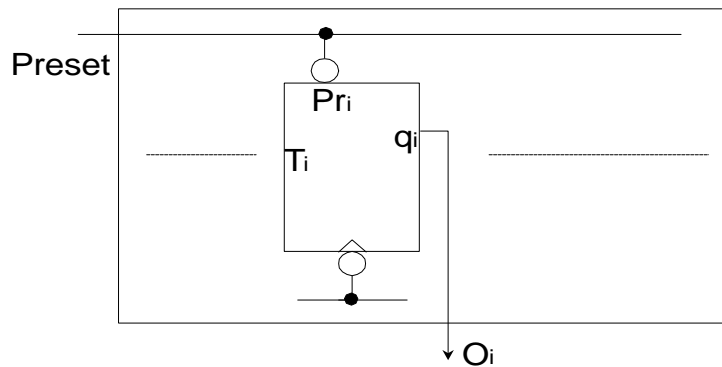
Clear o Reset síncrono

En el clear o reset síncrono no se utilizan las entradas asíncronas de los biestables. Se trata en este caso de introducir por la entrada T de cada uno de los biestables el valor adecuado para que cuando se reciba un flanco activo en **clk**, cada uno de los biestables se ponga a 0.

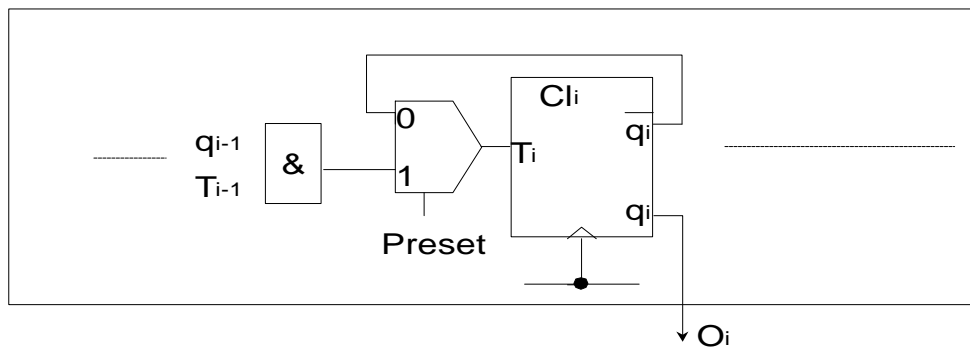


Preset asíncrono

Es idéntico al clear asíncrono salvo que las entradas asíncronas de los biestables a usar es Pr.

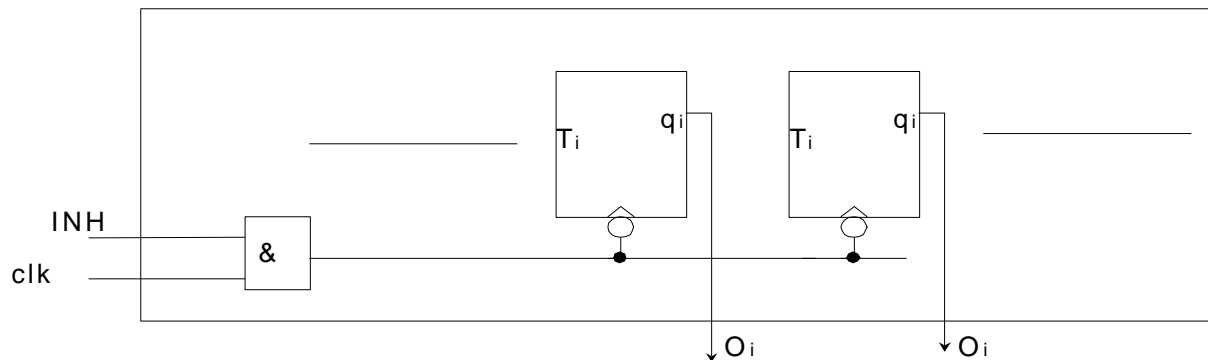


Preset síncrono

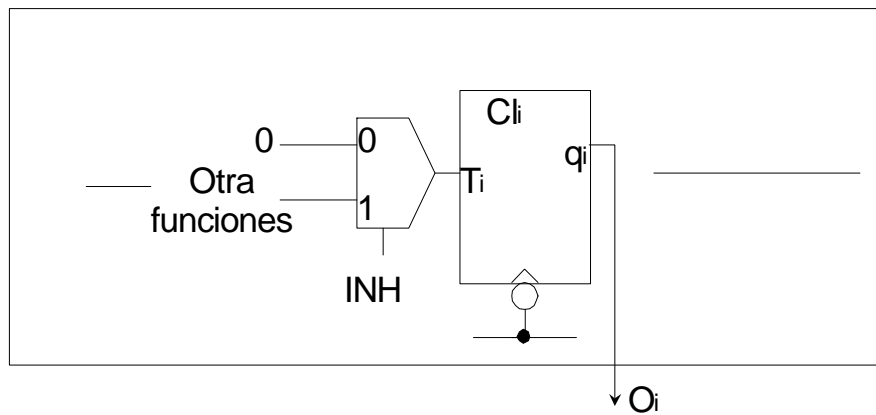


1.1.2 Inhibición

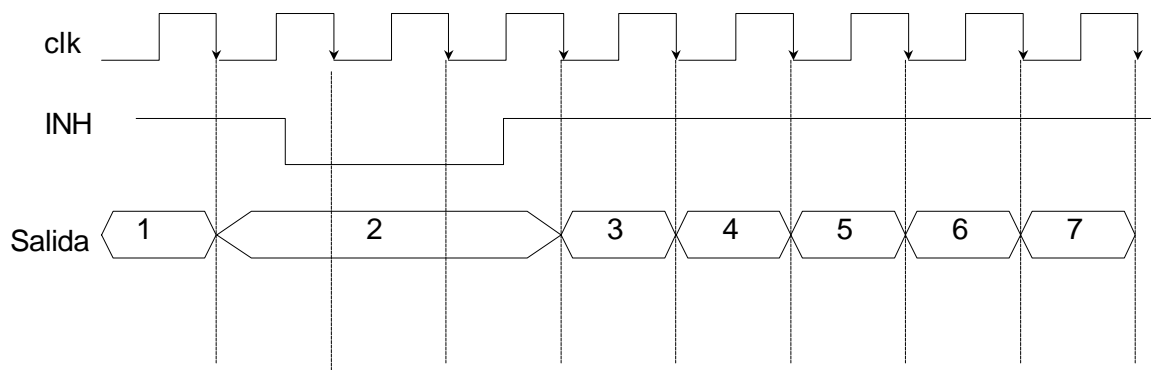
Es una línea de control (activa en alta o en baja) cuya misión es detener el proceso de cuenta del contador.



No obstante este tipo de implementación, que se suele denominar asíncrona, presenta problemas a la hora de activar la inhibición de cuenta, ya que en función del nivel de la señal de reloj, se pueden producir incrementos no deseados.



En la siguiente figura se representa el cronograma para la inhibición síncrona.



1.1.3 Load (Carga en paralelo)

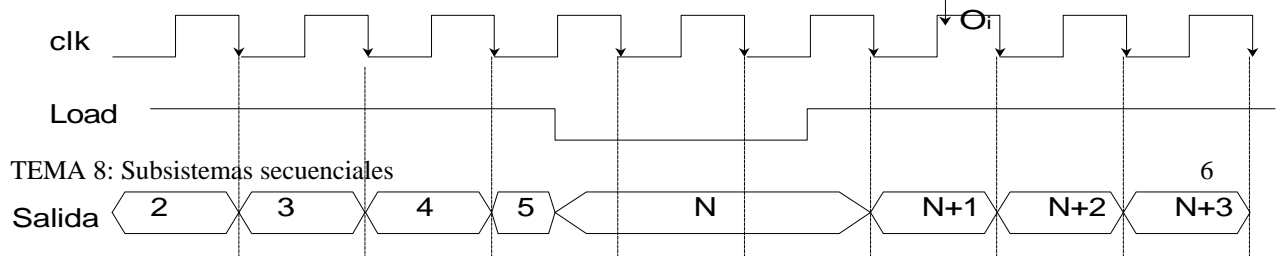
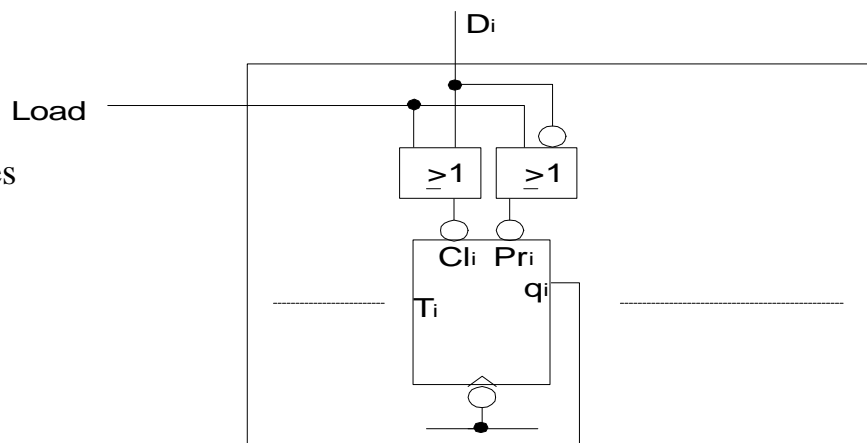
En determinadas aplicaciones es interesante cargar al contador con un valor de cuenta inicial. Esto se consigue con la línea de Load.

Load asíncrono

$$Cl_i = Load + D_i$$

$$Pr_i = Load + D_i'$$

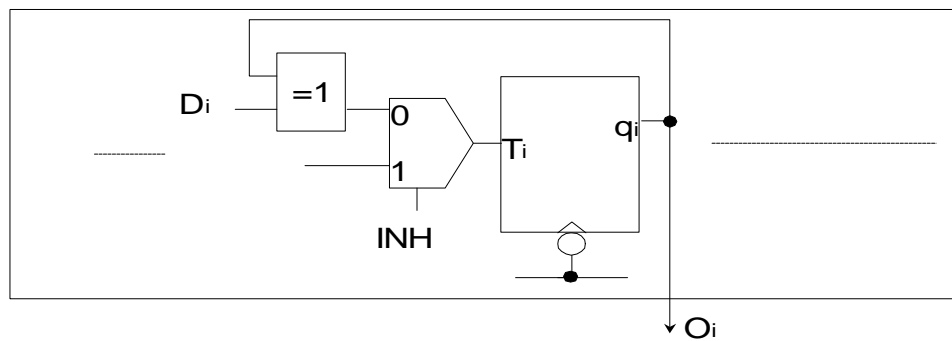
El circuito equivalente es



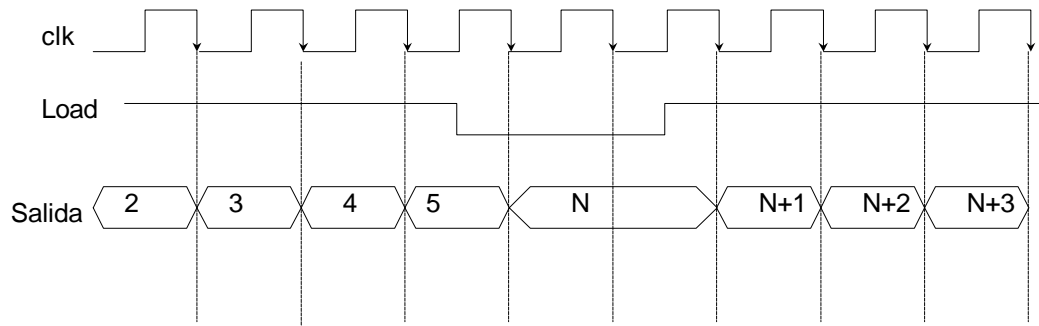
Load síncrono

$$T_i = q_i \oplus D_i$$

La siguiente figura muestra la estructura de la etapa i de un contador que incorpora la carga síncrona.

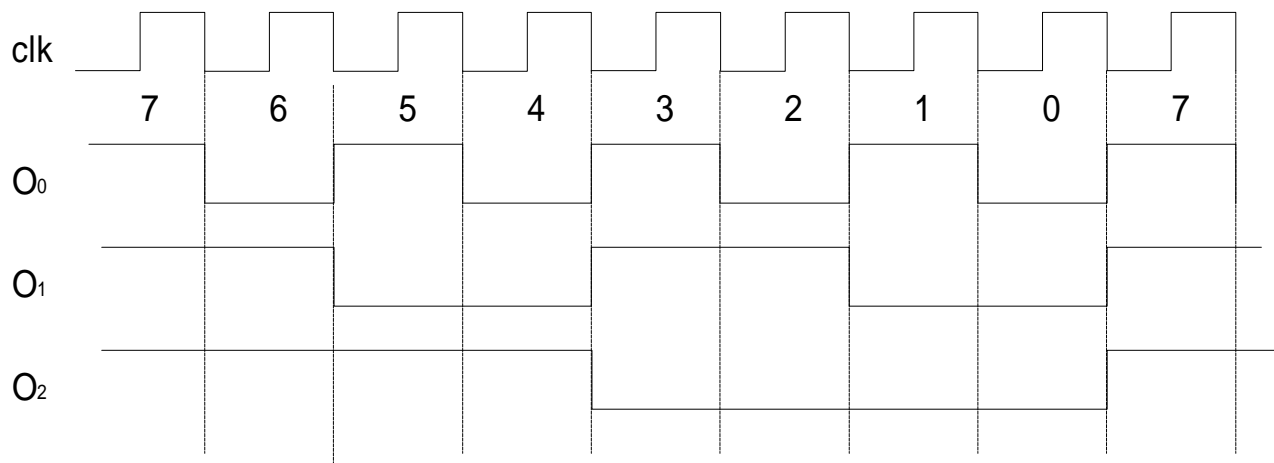


En la siguiente figura se muestra un cronograma donde se observa el funcionamiento de este tipo de operación.



1.1.4 Contadores reversibles (UP/DOWN)

Los contadores estudiados hasta ahora sólo tienen capacidad de cuenta ascendente, existen otros que tienen capacidad de cuenta descendente o incluso ambas (ascendente y descendente). Estos últimos son los contadores reversibles. La siguiente figura muestra la salida de un contador descendente de módulo 8.



O_0 cambia de valor en cada ciclo de reloj. $T_0=1$;

O_1 cambia en los flancos de bajada de clk si $O_0=0$, en caso contrario no. Esto implica que $T_1 = q_0'$

O_2 cambia en los flancos de bajada de clk si $O_0=0$ y $O_1=0$, en caso contrario no. Esto implica que $T_2 = q_1' q_0'$.

$T_i = q_0' q_1' \dots q_{i-1}'$ ($i=1, 2, \dots, n-1$)

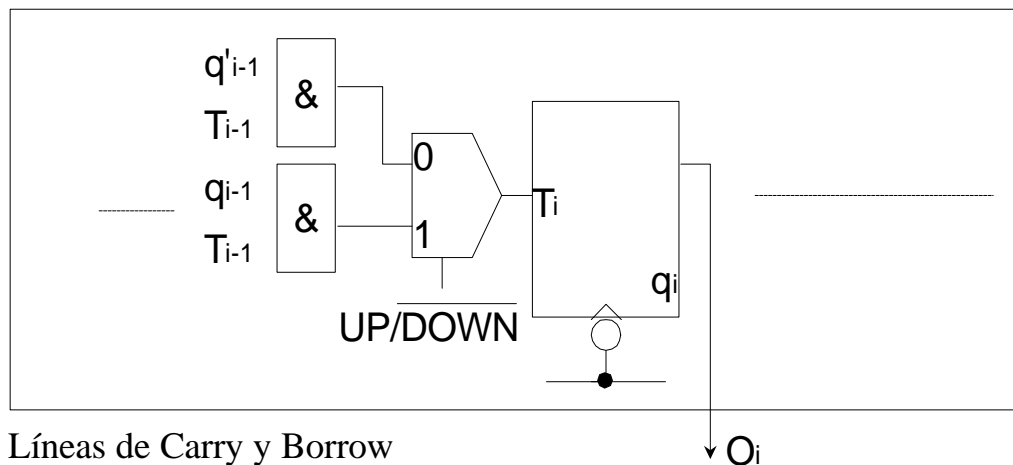
De forma equivalente, la expresión anterior se puede escribir como

$$T_i = q_{i-1}' T_{i-1}$$

Por tanto la entrada T_i de cada biestable del contador reversible debe tener la siguiente expresión

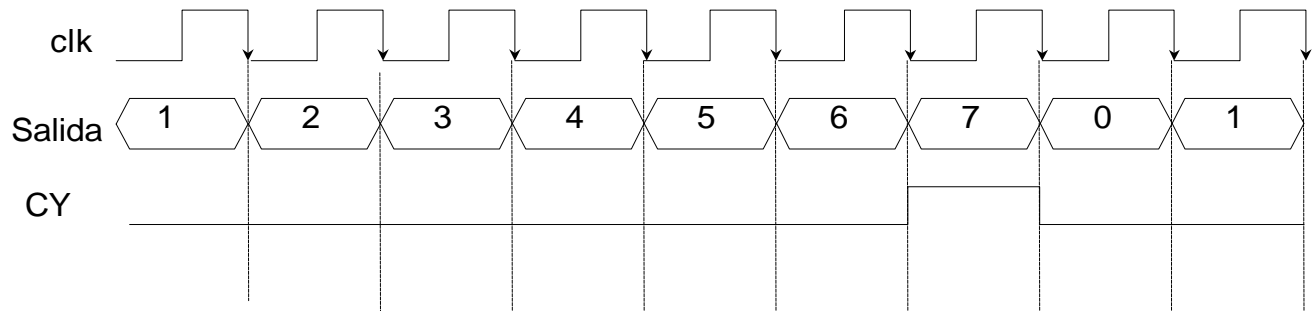
$$T_i = q_{i-1} T_{i-1} \text{ UP/DOWN} + q_{i-1}' T_{i-1} \text{ UP/DOWN}'$$

Y el circuito de la etapa i se muestra en la siguiente figura

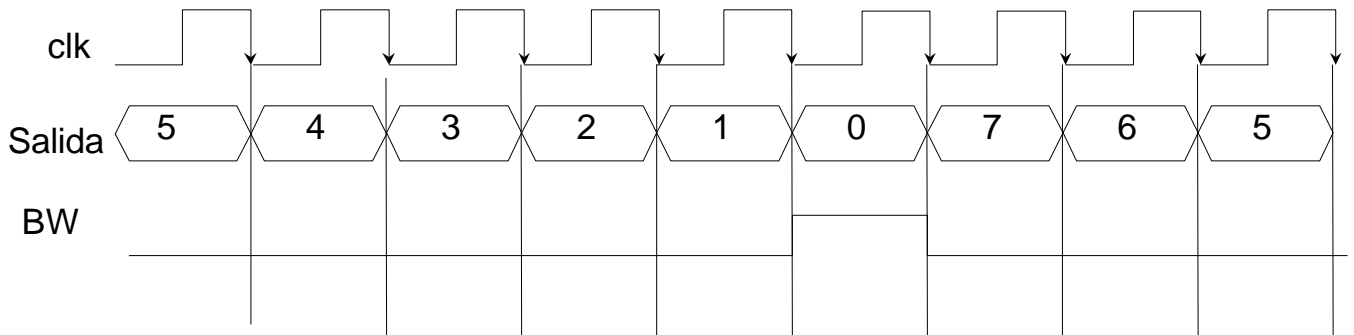


1.1.5 Líneas de Carry y Borrow

Son salidas del contador que informan de la llegada al último estado de cuenta del mismo. Para contadores ascendentes, se usa la salida Carry(CY) y para descendentes Borrow(Bw).



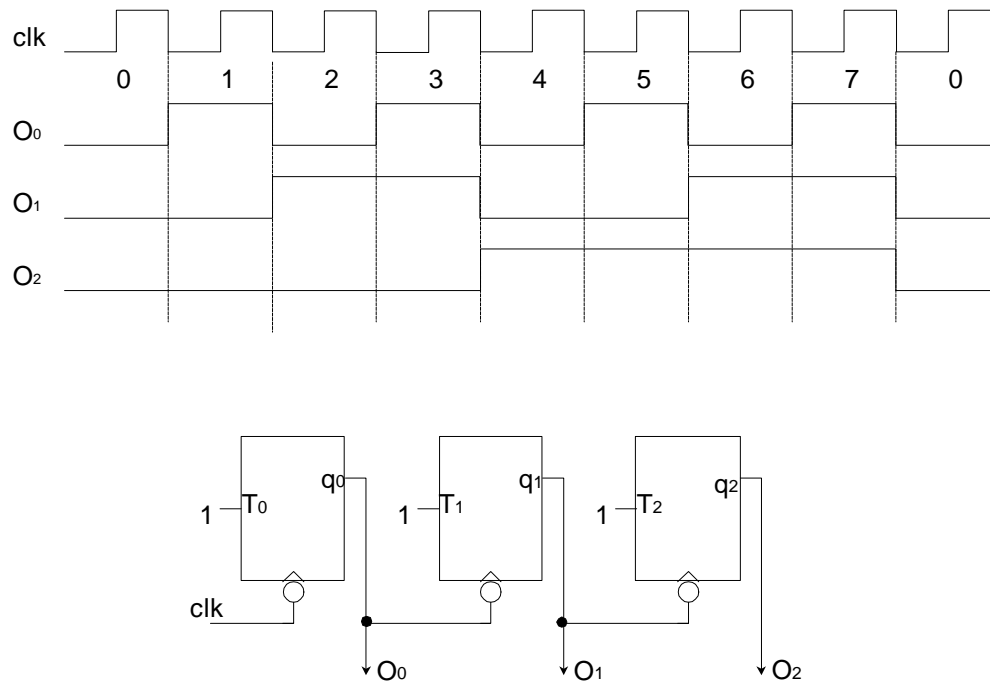
En cambio la señal de Borrow se activa cuando el contador descendente alcanza su estado de cuenta más bajo, el 0. (ver siguiente figura)



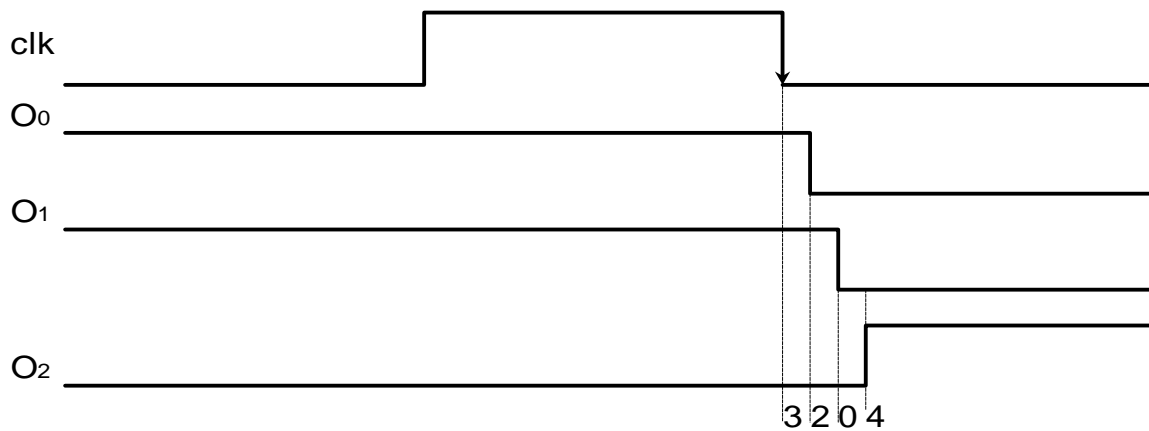
Los contadores reversibles disponen de una señal de salida TC(Terminal Count) que se pone a 1 si se alcanza el estado de cuenta más alto si el contador tiene el modo ascendente(CY) o se activa en el estado de cuenta 0 si el contador está programado como descendente (Bw)

1.2 Contadores asíncronos

Otro diseño alternativo al contador módulo 8 del principio del apartado 2.1 es el que se muestra en la siguiente figura.



Este tipo de diseño alternativo se denomina también contador de rizado o ripple-counter.



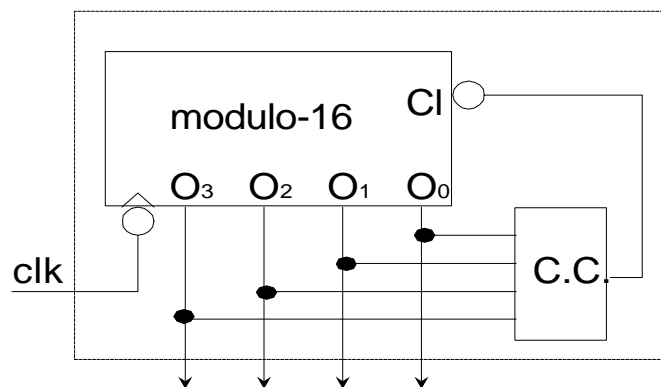
Los contadores asíncronos comerciales disponen de señales de control, como Reset, Load,.. todos, con funcionamiento asíncrono (ya descrito en el apartado anterior).

1.3 Contadores con módulo diferente a la potencia de dos

- Diseñarlo con biestables y puertas, como si se tratara de una máquina secuencial síncrona
- Usando contadores y puertas. Disponiendo de un contador con un módulo mayor del que se desea diseñar, y puertas lógicas, podemos hacer que este se comporte contando sólo aquellos estados de interés.

Ejemplo: Se desea construir un contador módulo 10, que cuente desde el 0 hasta el 9, usando un contador módulo 16 y puertas lógicas.

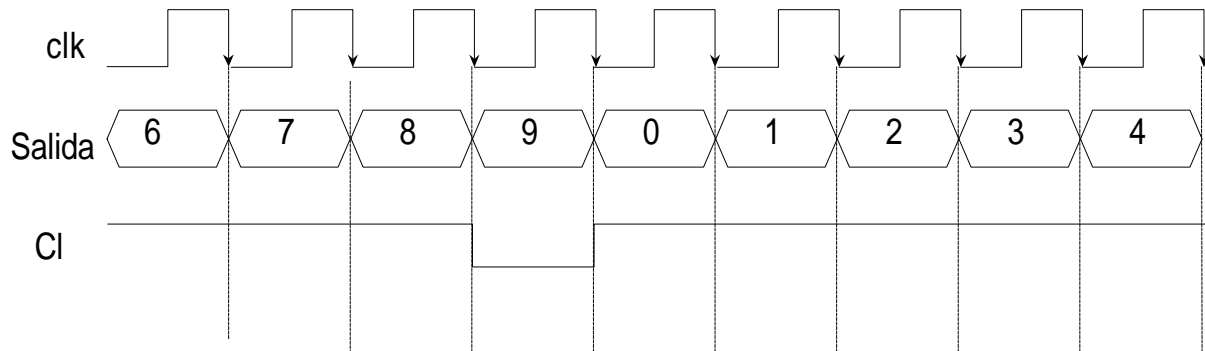
→ Con clear síncrono (activo en bajo)



		0 ₃ 0 ₂			
0 ₁ 0 ₀		00	01	11	10
	00	1	1	x	1
	01	1	1	x	0
	11	1	1	x	x
	10	1	1	x	x

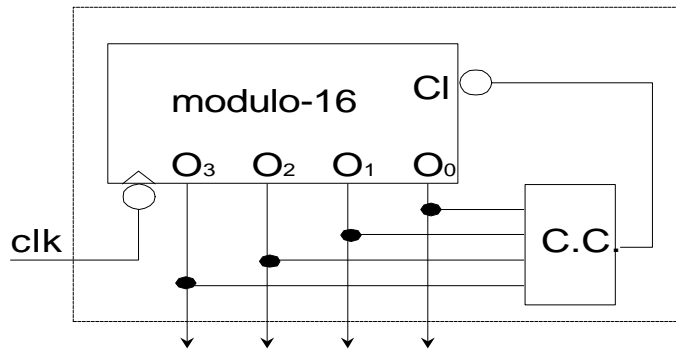
Cl

Por tanto $Cl = (O_3 O_0)'$



→ Con clear asíncrono (activo en bajo)

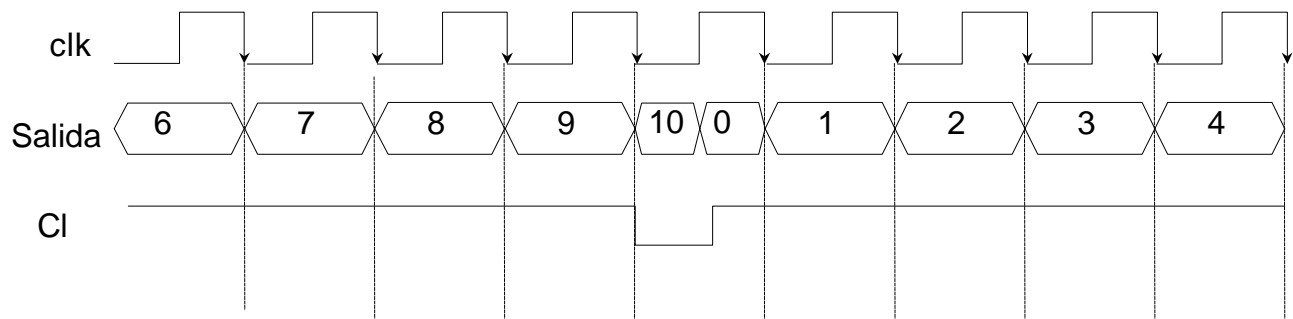
El procedimiento con el clear asíncrono es similar que con el síncrono, salvo que este caso la señal de clear debe activarse en el estado de cuenta 10.



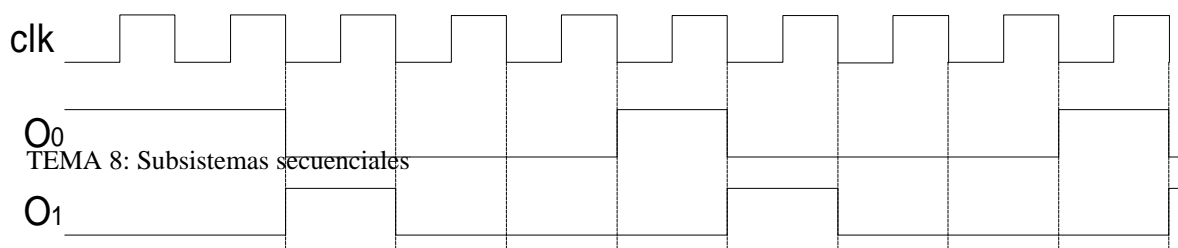
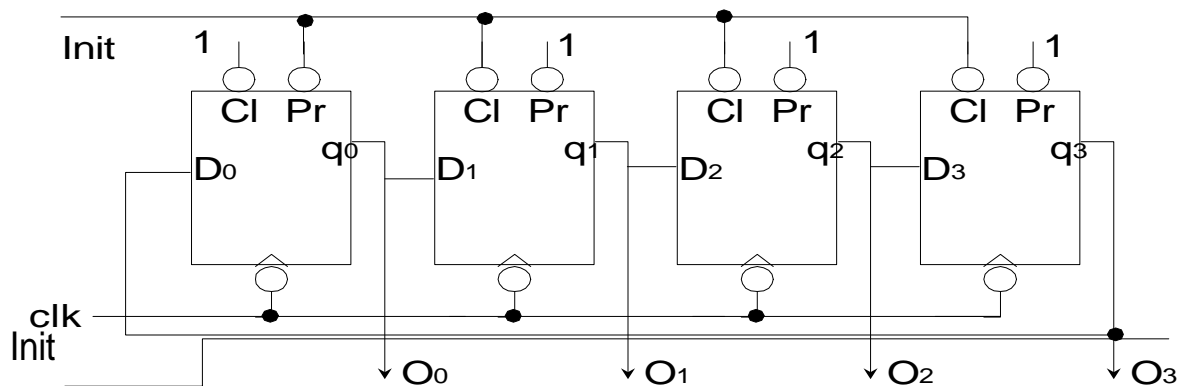
O_3O_2					
O_1O_0	00	01	11	10	
00	1	1	x	1	Cl
01	1	1	x	1	
11	1	1	x	x	
10	1	1	x	0	

Por tanto $Cl = (O_3 O_1)'$

La siguiente figura muestra un cronograma del funcionamiento de este contador, donde se ve que el estado de cuenta 10 aparece durante un tiempo pequeño, el necesario para la ejecución de la operación Clear.



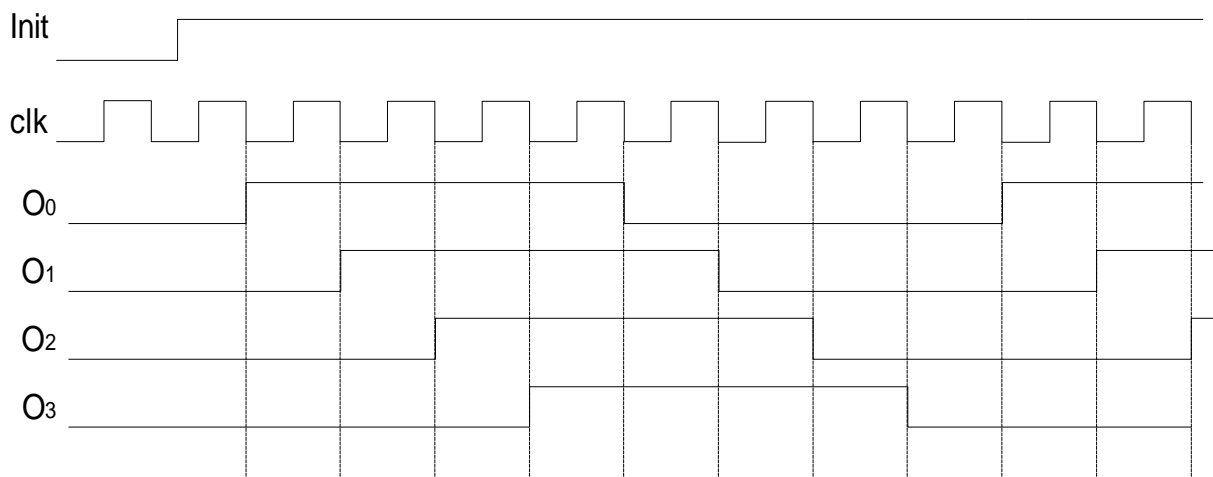
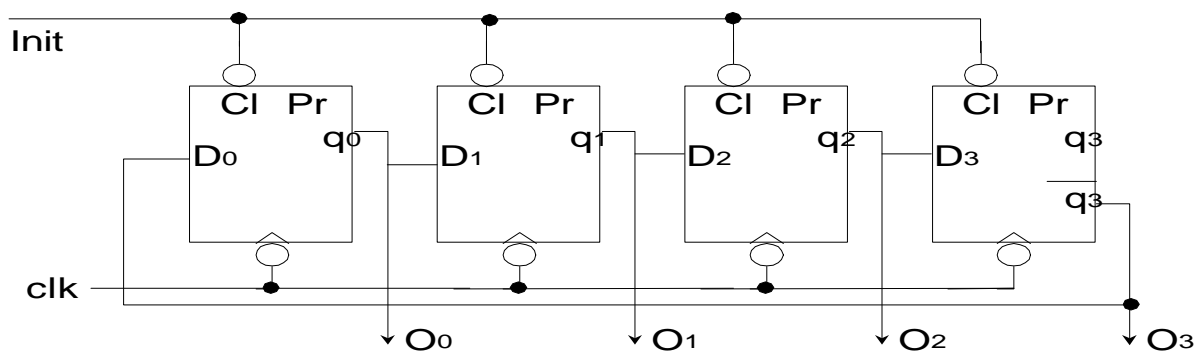
1.4 Contadores de anillo y conmutado en cola



Una modificación al contador anterior, la constituye el anillo de Johnson o conmutado en cola, el cual permite un número mayor de estados.

Son dos las diferencias con el contador en anillo:

- 1) La inicialización. Todos los biestables del contador se inicializan con 0
- 2) Las entrada de cada biestable está conectada con la salida del biestable anterior, salvo el primero, cuya entrada es la salida complementada del último biestable, el que está en la cola. (De ahí el nombre de conmutado en cola)



2. REGISTROS

Un registro de n bits es un dispositivo que tiene capacidad de almacenar n bits. Internamente están formados por biestables, tantos como bits sea capaz de almacenar el registro. Normalmente estos dispositivos son síncronos siendo los biestables D los más usados para la implementación interna.

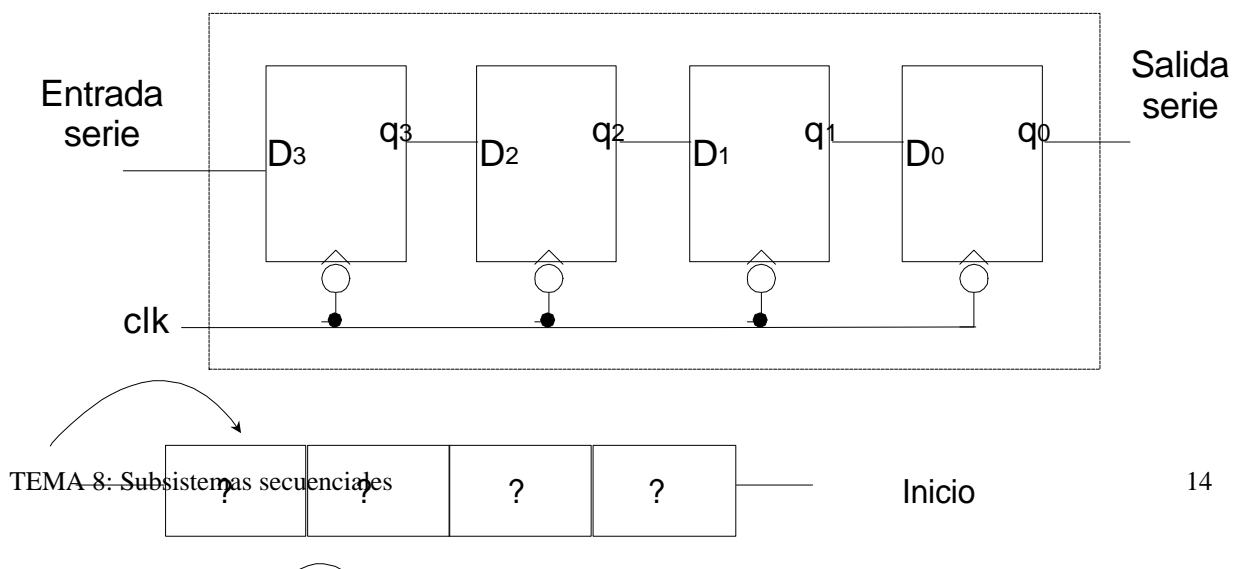
En cuanto a las operaciones básicas que se realizan sobre los registros destacamos fundamentalmente dos: escritura (W) y lectura (R), aunque habitualmente se pueden encontrar registros que incorporan operaciones como Clear.

Los registros se pueden clasificar en función de cómo se lean o escriban los bits, así podemos encontrar:

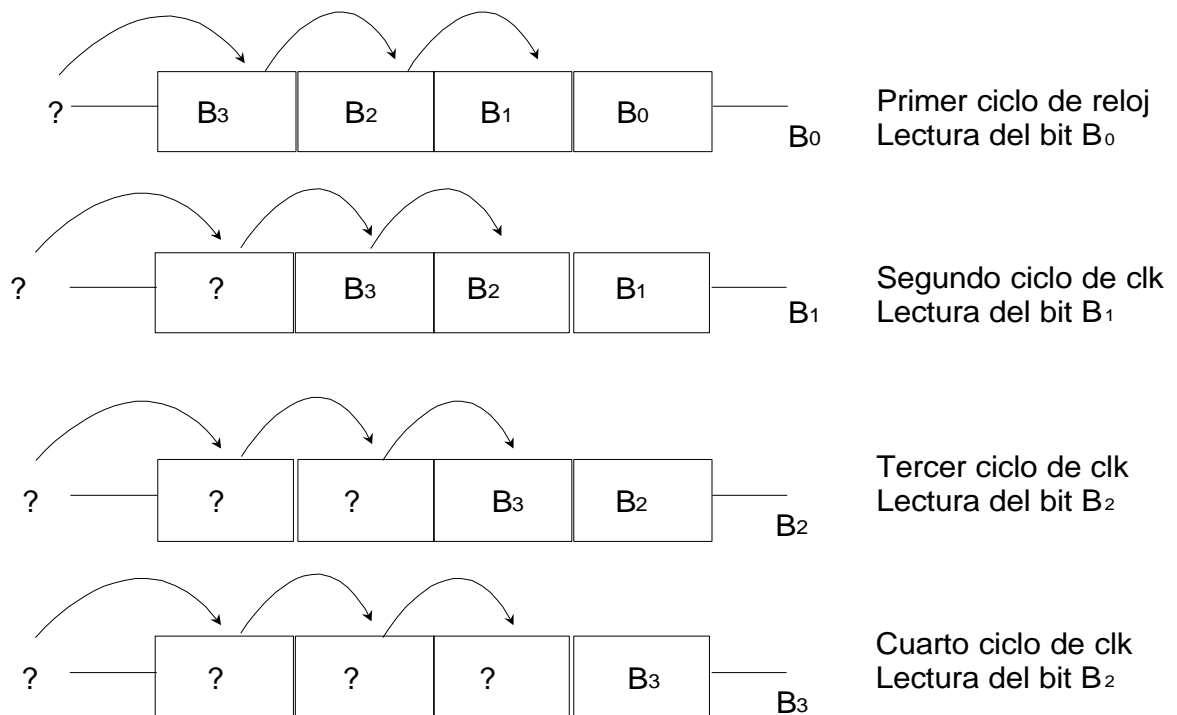
- registros con entrada serie (entrada hace referencia a escritura) y salida serie (salida hace referencia a lectura)
- registros con entrada serie y salida paralelo
- registros con entrada paralelo y salida serie
- registros con entrada paralelo y salida paralelo.

Todos los registros que tengan algún modo de funcionamiento serie para lectura o para escritura se denominan **registros de desplazamiento** (Shift registers). Podemos encontrar dos tipos de registros de desplazamiento en función del sentido de movimiento de los bits: izquierda o derecha.

3.1. Registro de entrada serie y salida serie

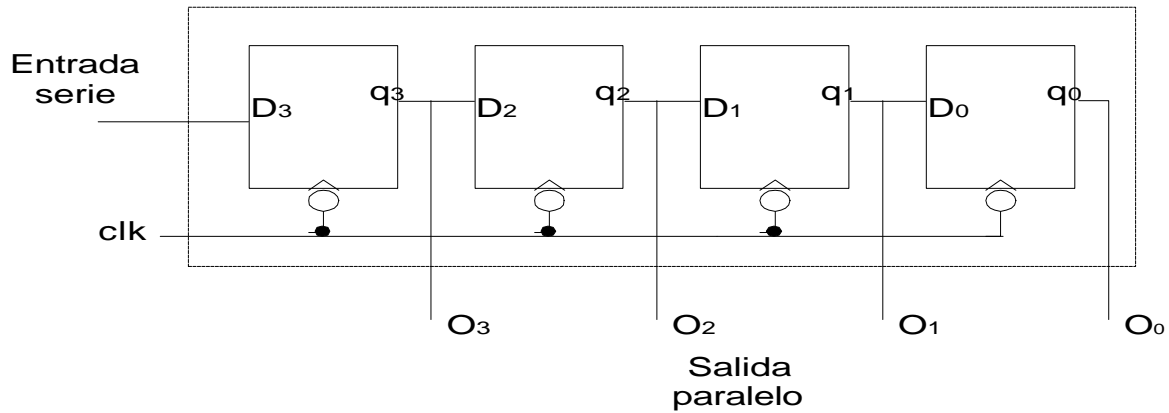


La siguiente figura ilustra el proceso de lectura en cada ciclo de reloj.



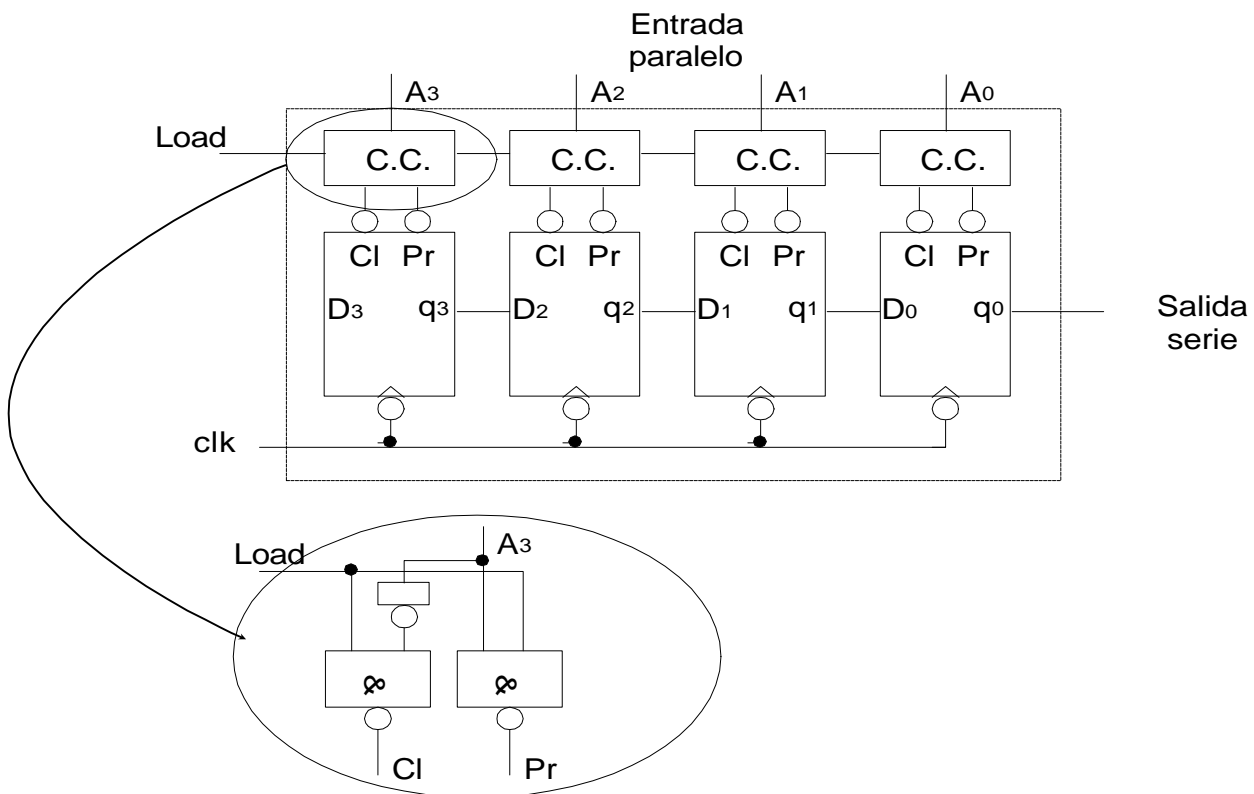
3.2 Registro con entrada serie y salida paralelo

La siguiente figura muestra la estructura básica de un registro de entrada serie y salida paralelo de 4 bits. Al igual que en el apartado anterior, este es un registro de desplazamiento a derecha



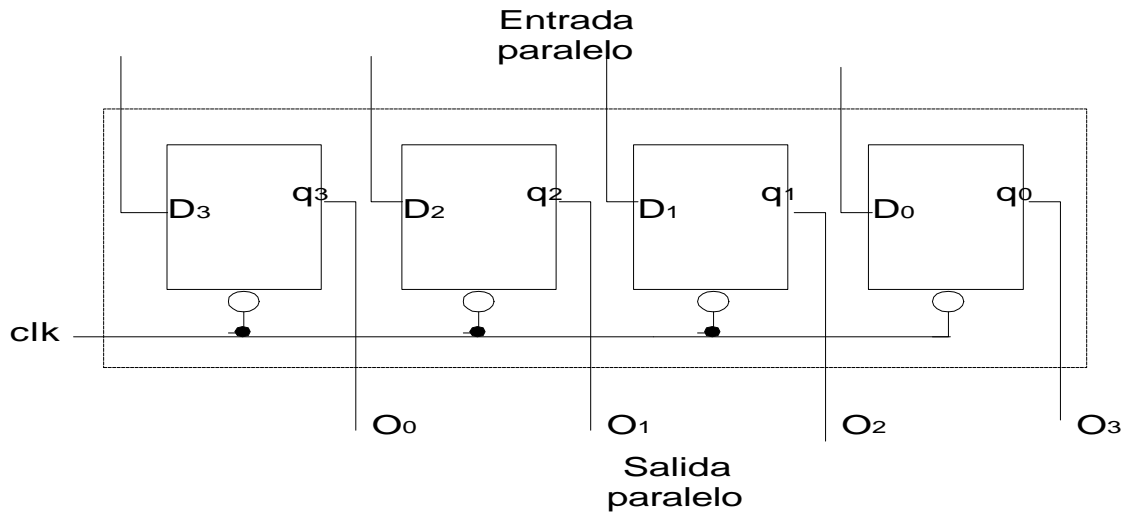
3.3 Registro con entrada paralelo y salida serie.

La siguiente figura ilustra la estructura básica de un registro de estas características.



3.4. Registro con entrada paralela y salida paralelo.

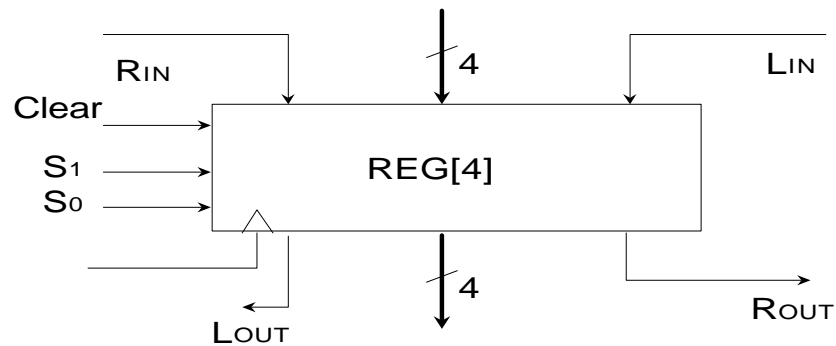
La siguiente figura ilustra la estructura de un registro con entrada y salida en paralelo de 4 bits.



3.5 Registro Universal

Un registro universal es aquel que tiene todas las formas de lectura y escritura posibles (tanto en serie como en paralelo)

Ejemplo.: Diseñar un registro universal de 4 bits que tenga las siguientes operaciones: desplazamiento a la derecha, desplazamiento a la izquierda, carga en paralelo y clear.



S ₁ S ₀	Operación
00	SHR (Desplazamiento a la derecha)
01	Load
10	NOP o INH
11	SHL (Desplazamiento a la izquierda)

Si $S_1 S_0=00$, desplazamiento a la derecha, las entradas de los cuatro biestables del registro deben ser (suponiendo D_3 el registro situado a la izquierda y D_0 el situado a la derecha)

$$D_3 = R_{in}$$

$$D_2 = q_3$$

$$D_1 = q_2$$

$$D_0 = q_1$$

Y la salida R_{out} está formada por q_0

Si $S_1 S_0=01$. Operación de Load, las entradas deben ser los valores que tengan las entradas en paralelo del registro. Designemoslas como B_0, B_1, B_2 y B_3

$$D_3 = B_3$$

$$D_2 = B_2$$

$$D_1 = B_1$$

$$D_0 = B_0$$

Si $S_1 S_0=10$. NOP, las entradas de los biestables deben ser

$$D_3 = q_3$$

$$D_2 = q_2$$

$$D_1 = q_1$$

$$D_0 = q_0$$

Para que no pierdan la información

Si $S_1 S_0=11$. Desplazamiento a la izquierda, las entradas de los biestables deben ser

$$D_3 = q_2$$

$$D_2 = q_1$$

$$D_1 = q_0$$

$$D_0 = L_{in}$$

Y la salida L_{out} está formada por q_3

En cualquier caso, las salidas O_3-0 son los valores de los biestables q_3-0

En general podemos decir que

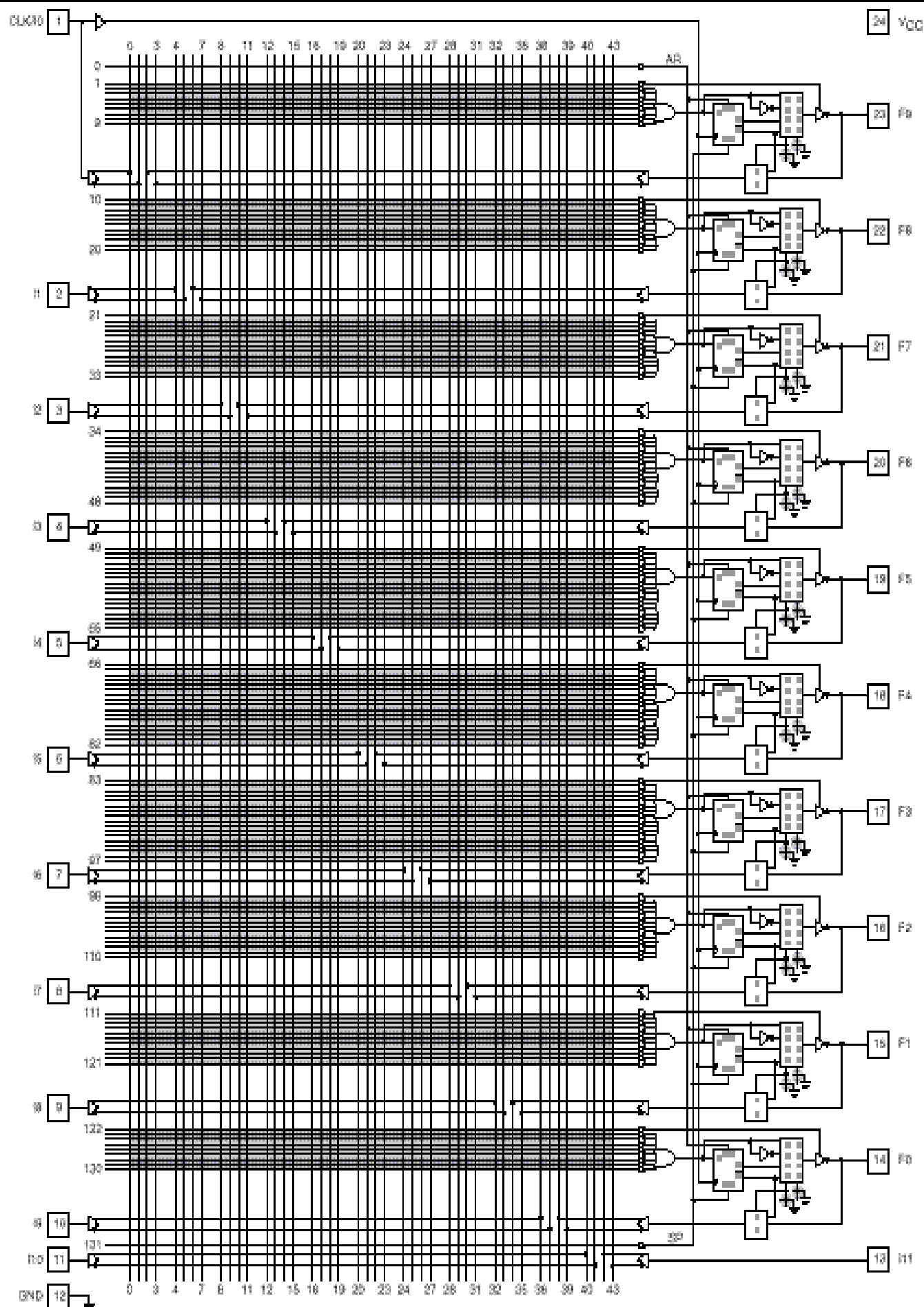
$$D_3 = S_1' S_0' R_{in} + S_1' S_0 B_3 + S_1 S_0' q_3 + S_1 S_0 q_2$$

$$D_2 = S_1' S_0' q_3 + S_1' S_0 B_2 + S_1 S_0' q_2 + S_1 S_0 q_1$$

$$D_1 = S_1' S_0' q_2 + S_1' S_0 B_1 + S_1 S_0' q_1 + S_1 S_0 q_0$$

$$D_0 = S_1' S_0' q_1 + S_1' S_0 B_0 + S_1 S_0' q_0 + S_1 S_0 L_{in}$$

PLD's SECUENCIALES



NOTE:

----- Programmable connection.

La estructura de la macrocelda aparece en la siguiente figura

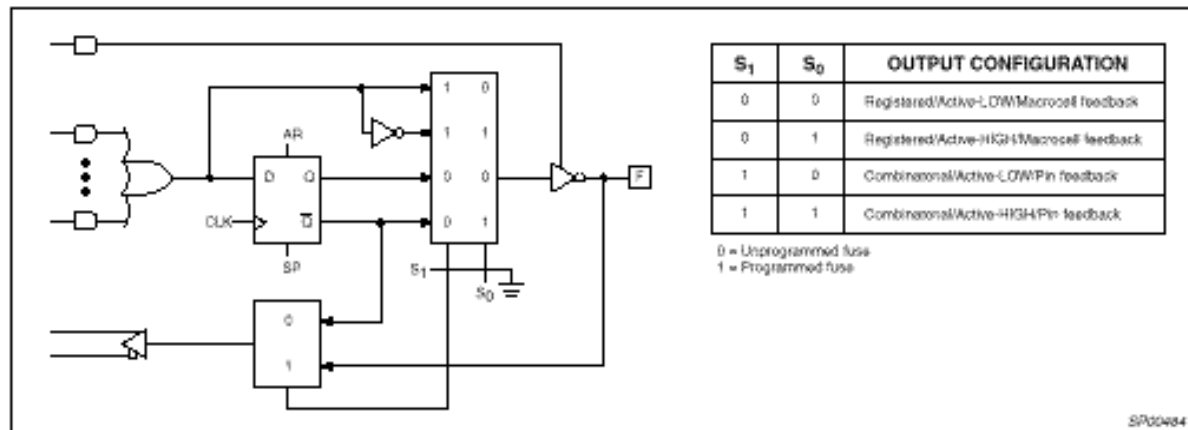


Figure 2. Output Macro Cell Logic Diagram

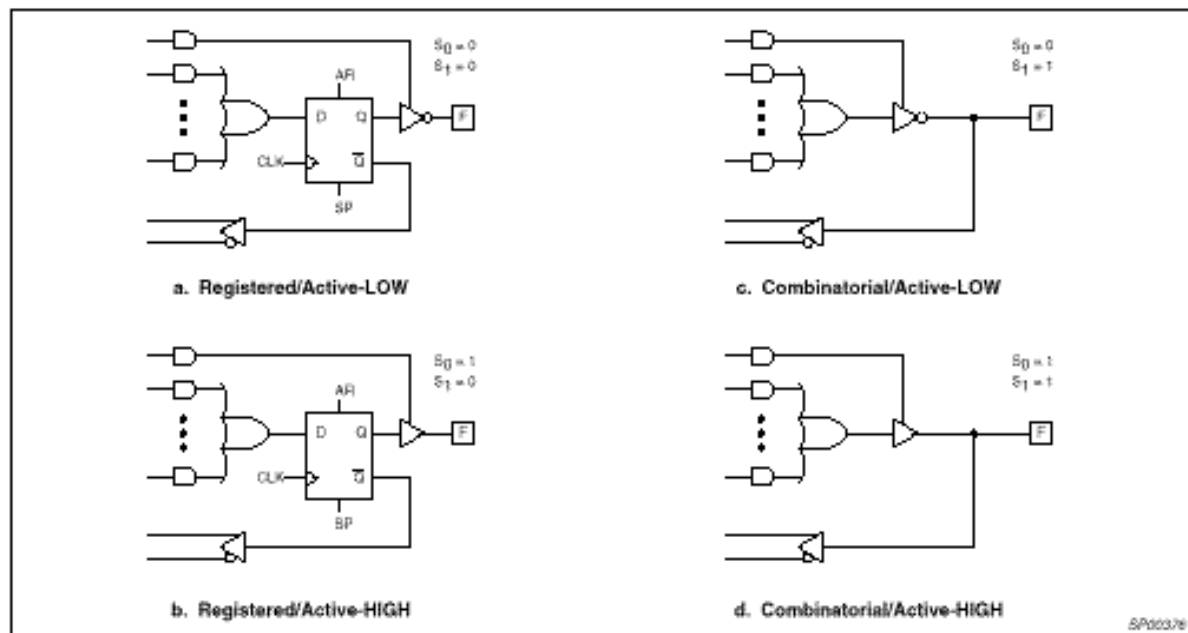
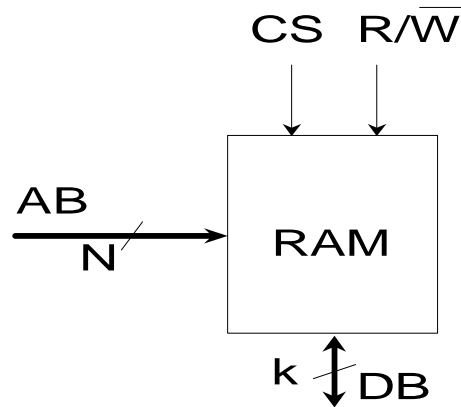


Figure 3. Output Macro Cell Configurations

3. RAM

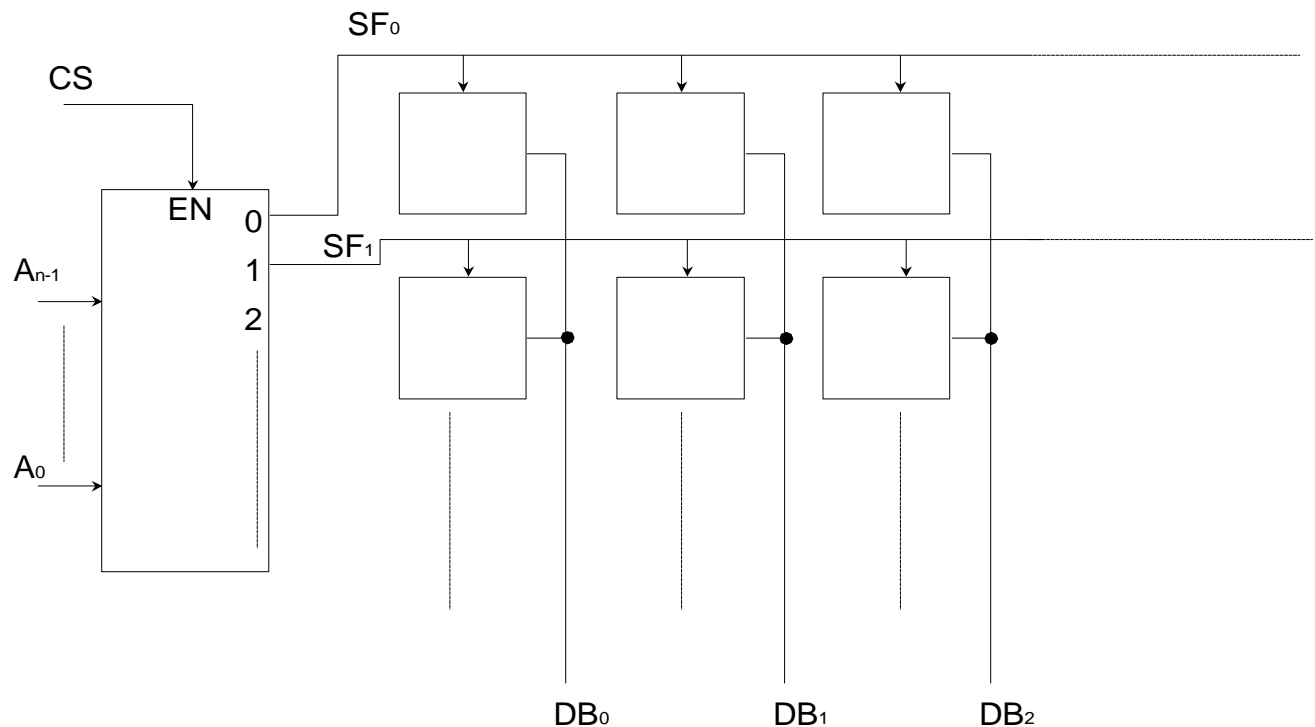
Son dispositivos de memoria de lectura y escritura, organizados de forma idéntica a las ROM, esto es, como un conjunto de palabras, cada una de las cuales tiene un número de bits determinado.



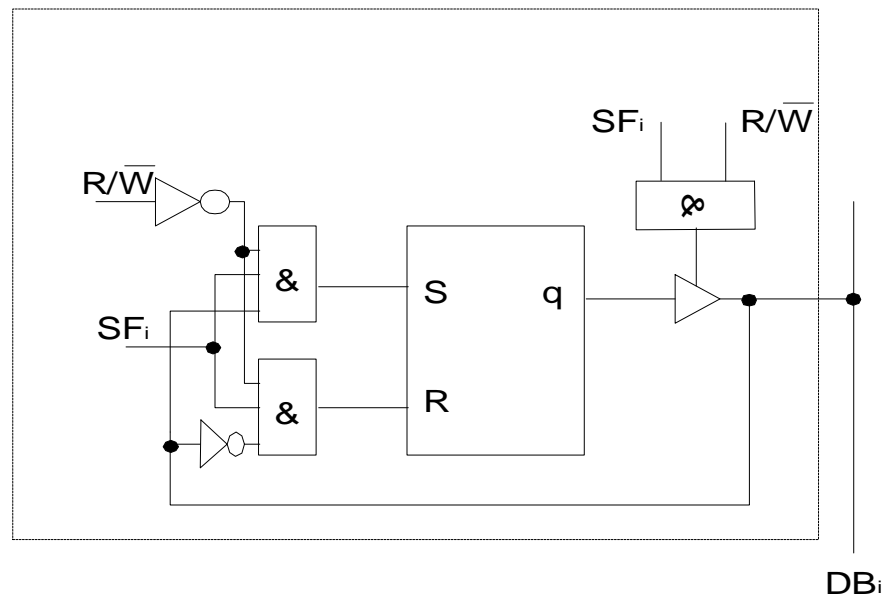
Existen dos tipos de RAM: RAM estática y RAM dinámica

RAM estática (SRAM)

Cada bit es almacenado en un biestable (normalmente del tipo SR).



La estructura de cada módulo aparece en la siguiente figura. Esta constituido por un biestable SR que en función del selector de fila SF_i y la señal de R/W , se escribirá con el bit contenido en la línea de datos, o mostrará su bit dicha línea en operación de lectura.



RAM dinámica(DRAM)

