

TEMA 8

SUBSISTEMAS SECUENCIALES

ÍNDICE

1. INTRODUCCIÓN

2. CONTADORES

2.1 Contadores síncronos

2.1.1 Reset(Clear) y Preset

2.1.2 Inhibición

2.1.3 Load

2.1.4 Contadores reversibles (UP/DOWN)

2.1.5 Líneas de Carry y Borrow

2.2 Contadores asíncronos

2.2 Contadores con módulo diferente a la potencia de dos

2.3 Contadores de anillo y conmutado en cola

3. REGISTROS

3.1 Registro de entrada serie y salida serie

3.2 Registro de entrada serie y salida paralelo

3.3 Registro de entrada paralelo y salida serie

3.4 Registro de entrada paralelo y salida paralelo

3.5 Registro universal

4. PLD's SECUENCIALES

5. RAM

5.1 RAM estática

5.2 RAM dinámica

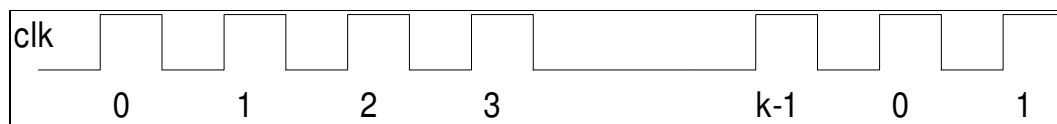
1. INTRODUCCIÓN

Existen dispositivos comerciales que implementan funciones secuenciales de propósito específico y general. Estos dispositivos son de uso muy común y un diseño a nivel de biestables y puertas lógicas de estos elementos provocaría que el dimensionado del circuito

global que los contiene sería excesivamente grande. De estos dispositivos de uso frecuente estudiaremos los contadores, registros, dispositivos lógicos programables y las RAM.

2. CONTADORES

Un contador módulo k es un circuito digital capaz de contar k sucesos distintos. Estos dispositivos no son otra cosa que circuitos secuenciales cuyos cambios de estado se producen, evidentemente, a ritmo de su señal de reloj, y en el que cada estado “memoriza” un valor de cuenta. Por tanto, un contador módulo-k, tiene k estados de cuentas distintos, desde el 0, hasta el k-1. Estos dispositivos podrán incrementarse o no en función de los valores lógicos que tomen ciertas señales de control que se estudiarán más adelante. Inicialmente se supondrá que el contador siempre está contando, por lo que cada valor de cuenta se corresponde con un ciclo de reloj. En la siguiente figura se muestra la evolución de los estados de cuenta de un contador módulo k. Se observa que a partir del ciclo k de reloj, estado de cuenta k-1, el contador vuelve a su estado inicial (estado de cuenta 0). Esto es una característica de todos los contadores.

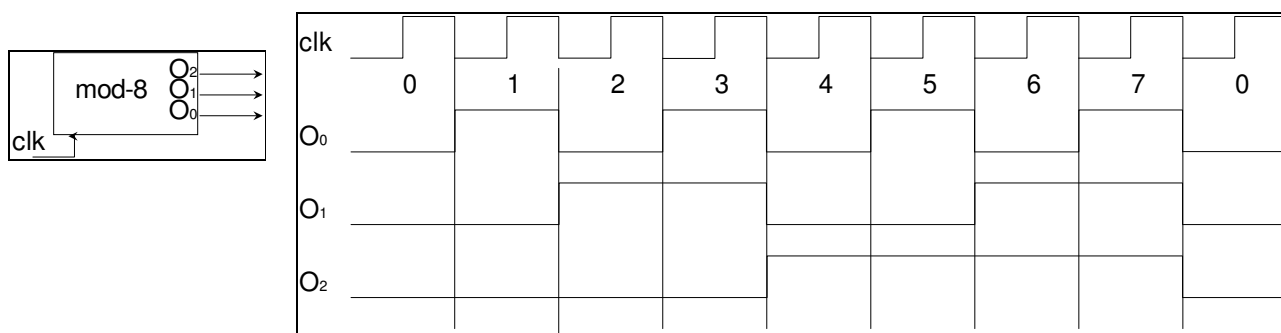


Un contador módulo-k, tiene k estados, y estos se pueden implementar con n biestables, existiendo una relación entre k y n determinada por la expresión:

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

Así un contador módulo 8 se puede construir usando 3 biestables, uno de módulo 16 con cuatro biestables, y así sucesivamente. 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).



En las siguientes secciones se presentarán dos posibles estructuras internas que permiten el funcionamiento de los contadores (estructura asíncrona y estructura síncrona) y se estudiarán

las líneas de control y salidas más comunes que, comercialmente, disponen estos dispositivos.

2.1 Contadores síncronos

Un contador tiene que estar formado internamente por biestables, cada uno de los cuales dispone de su entrada de reloj. Un contador síncrono es un circuito cuyos biestables internos son disparados por flanco y todos reciben la misma señal de reloj. Un contador síncrono, pues, cumple con las condiciones de una máquina secuencial síncrona.

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.

Se deja al estudiante que obtenga la estructura interna del contador usando el apartado 1) y aquí se describirá la 2).

Si analizamos las ondas obtenidas para el contador módulo-8 observamos que:

O_0 : cambia de valor con cada suceso de entrada (1 ciclo de clk)

O_1 : cambia de valor con cada suceso de O_0 (1 ciclo de O_0)

O_2 : cambia de valor con cada suceso de O_1 (1 ciclo de O_1)

En particular

O_0 : cambia de valor con cada flanco de bajada de clk

O_1 : cambia de valor en los flancos de bajada de clk si $O_0=1$

O_2 : cambia de valor en los flancos de bajada de clk si $O_0=O_1=1$

Si se hace corresponder la salida O_i del contador con el valor q_i del biestable interno, se tiene que el biestable q_0 debe cambiar de valor 0,1,0,1... en cada flanco de bajada de clk; el biestable q_1 debe cambiar de estado en el flanco de bajada de clk si q_0 es igual a 1, en caso contrario, q_1 mantiene su valor; y el biestable q_2 debe cambiar de estado si $q_1 = q_2 = 1$, y en caso contrario mantiene su valor.

Esta descripción nos hace pensar que el tipo de biestable que mejor encaja para el diseño de los contadores es el biestable T, ya que este, si su entrada es 0, provoca que el biestable no cambie de valor y si su entrada es 1, provoca el cambio del valor almacenado.

Por tanto, si usamos biestables tipo T, tenemos que

$$T_0 = 1$$

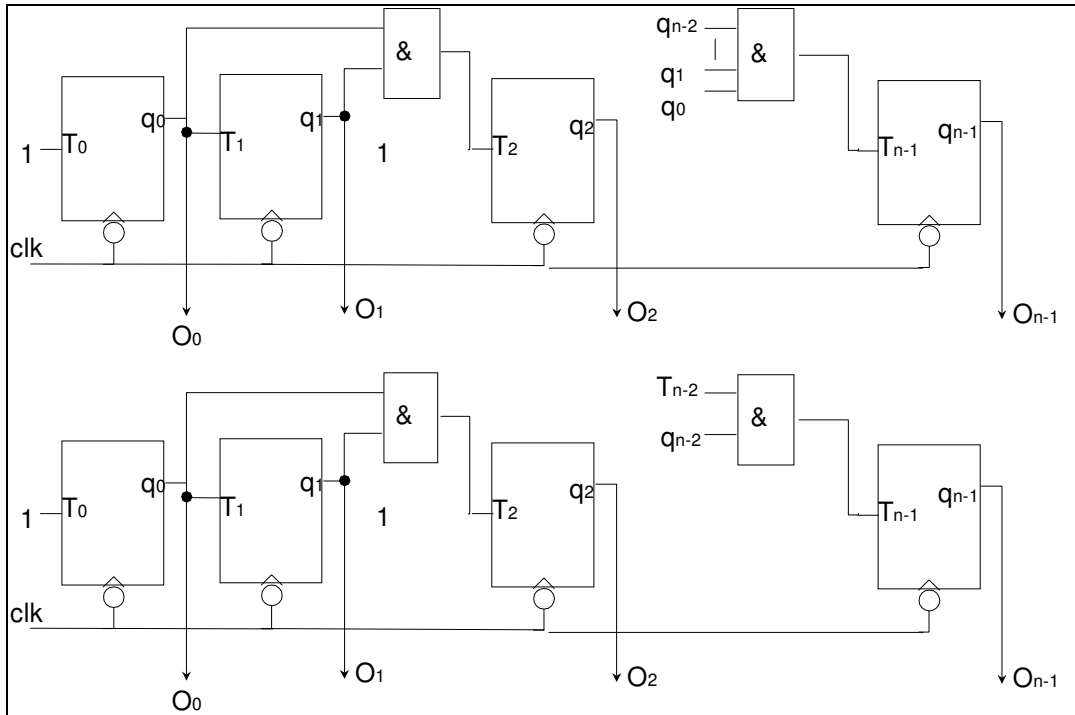
$$T_1 = q_0$$

$$T_2 = q_1 q_0$$

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

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

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



En los siguientes apartados se estudiarán las señales de control y salida más comunes que acompañan a los contadores.

2.1.1 Reset(Clear) y Preset

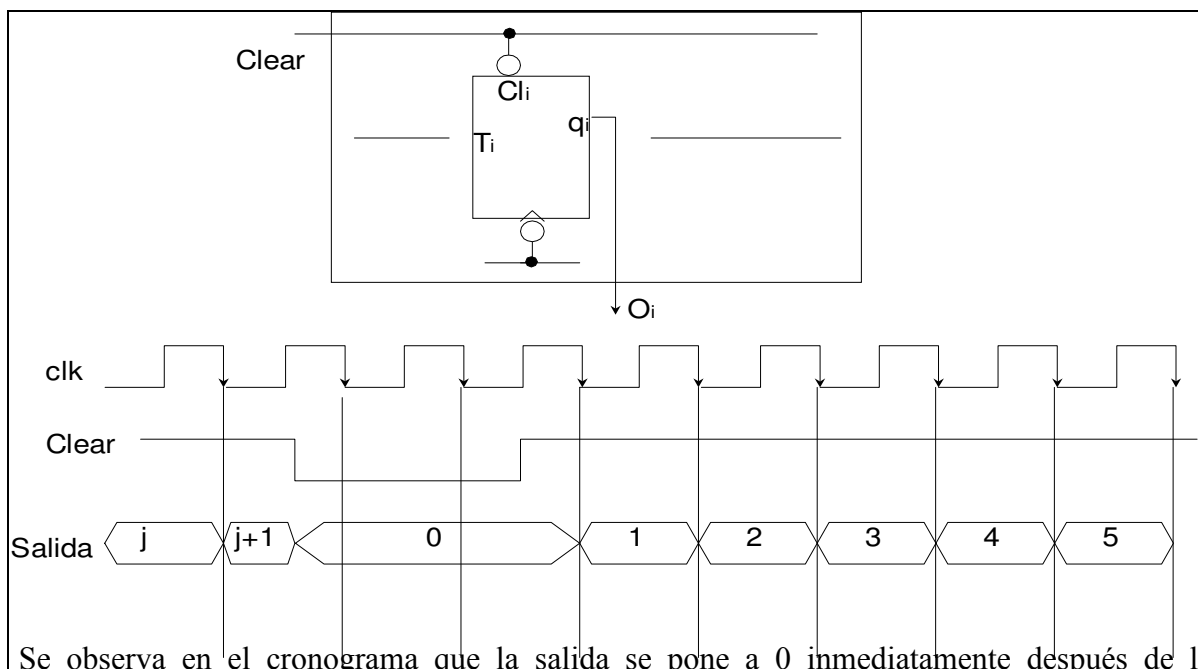
La línea de Reset o Clear permite inicializar el contador a su estado de cuenta 0 cuando es activada. La línea de Preset permite inicializar el contador a su valor de cuenta más alto (aquel en que todos los biestables están a 1). Estas líneas pueden ser activas en alto o en bajo, y existen dos modalidades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

Supongamos que la línea Clear de un contador es asíncrona. Esto implica que inmediatamente después a la activación de dicha señal, el contador pasa al estado de cuenta 0, independientemente de la señal de reloj. Por el contrario, si el clear se activase, pero tuviese un modo de funcionamiento síncrono, el contador se pondría en el estado de cuenta cero cuando llegase un flanco activo de la señal de reloj. En resumen, un clear asíncrono no depende de clk para su funcionamiento, mientras que un síncrono sí. Esto es igualmente aplicable para la línea de Preset.

A continuación se estudiará cómo se implementan en el contador estas funciones.

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. El contador está formado por biestables tipo T. Supongamos que estos tienen entradas de Cl y Pr. Implementar un Clear asíncrono en el contador es equivalente a unir dicha entrada de Clear a todas las entradas Cl de los biestables que forman parte del contador. De esta manera, cuando clear se active, todos los biestables se ponen a cero de forma asíncrona y por consiguiente el contador pasa al estado de cuenta 0. En la siguiente figura se ha representado la estructura de la celda i de un contador de n bits, y la posible respuesta funcional de la salida del mismo para la situación de activación de la línea Clear, que este caso es activa en baja.



Se observa en el cronograma que la salida se pone a 0 inmediatamente después de la activación de la línea de Clear (segundo ciclo de reloj) y se mantiene así hasta que Clear vuelva a valer 1 (cuarto ciclo de reloj), a partir del cual, el contador vuelve a iniciar su ritmo de cuenta.

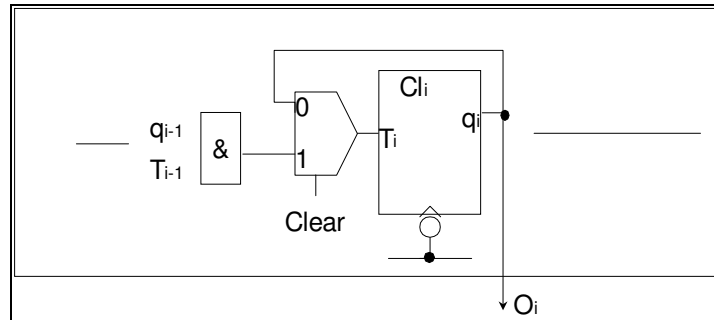
Clear o Reset síncrono

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. Hay que tener presente que el contador debe incrementarse cuando no se active el Clear, por lo que las entradas T deben tener los valores ya determinados en el inicio del apartado.

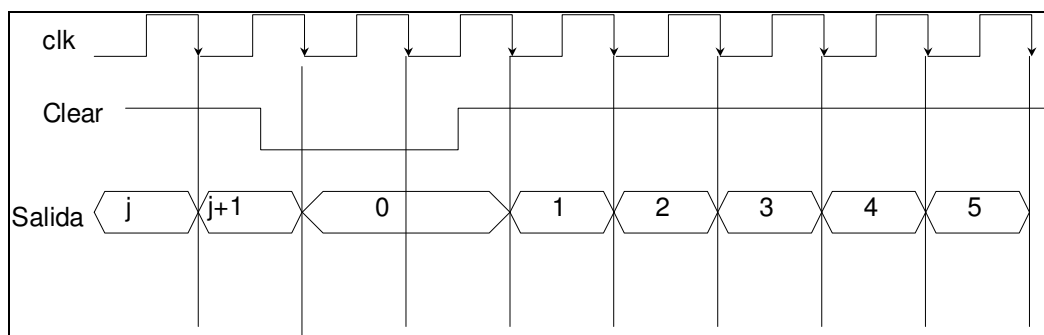
En primer lugar, hay que determinar qué se debe introducir a la entrada del biestable i para que su salida q_i pase a 0 en el siguiente ciclo de reloj. Supongamos que la salida $q_i = 0$, por tanto la entrada del biestable i debe ser $T_i = 0$ para que en el siguiente ciclo se mantenga el cero. En cambio si la salida del biestable $q_i = 1$, la entrada T_i debe ser un 1 lógico, para que en el siguiente ciclo, la salida q_i pase a valer 0. En resumen, si queremos hacer un Clear, la entrada del biestable T_i debe ser igual a q_i . Si todos los biestables del contador disponen de

esta entrada, el contador pasará al estado de cuenta 0 cuando se reciba el flanco activo de reloj.

La siguiente figura muestra la estructura de la celda básica de un contador que permite un clear síncrono. La entrada del biestable i está conectada a la salida de un multiplexor de dos canales controlado por la señal Clear (activa en baja). De esta forma, si $Clear=0$, el canal 0 del multiplexor (o sea q_i) pasa a la entrada T_i , y se produce el efecto deseado. Si $Clear=1$, el canal 1 del multiplexor (o sea q_{i-1}) pasa a la entrada T_i , y esto implica que el contador se incrementa (modo de funcionamiento normal)

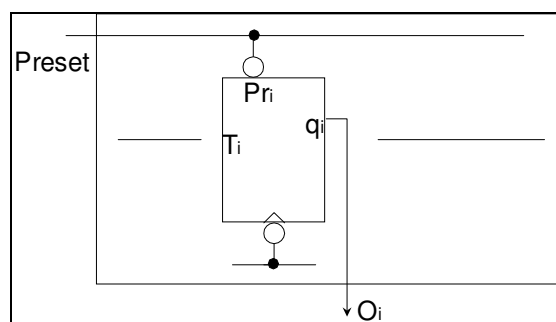


En el siguiente cronograma se ve claramente el funcionamiento del clear síncrono, el cual no se produce realmente hasta la recepción de un flanco activo en clk.



Preset asíncrono

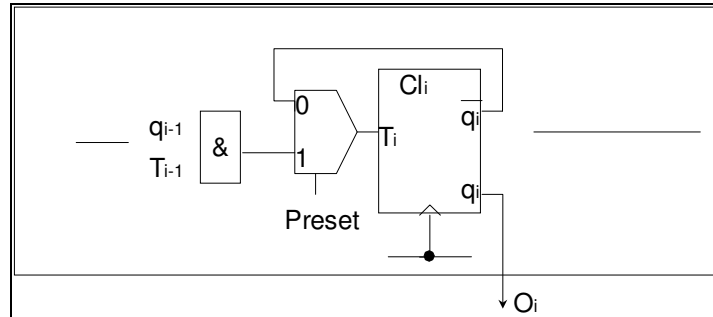
Es idéntico al clear asíncrono salvo que las entradas asíncronas de los biestables a usar es Pr.



De esta forma, cuando se activa la señal de Preset del contador, todos los biestables se ponen a 1 de forma asíncrona.

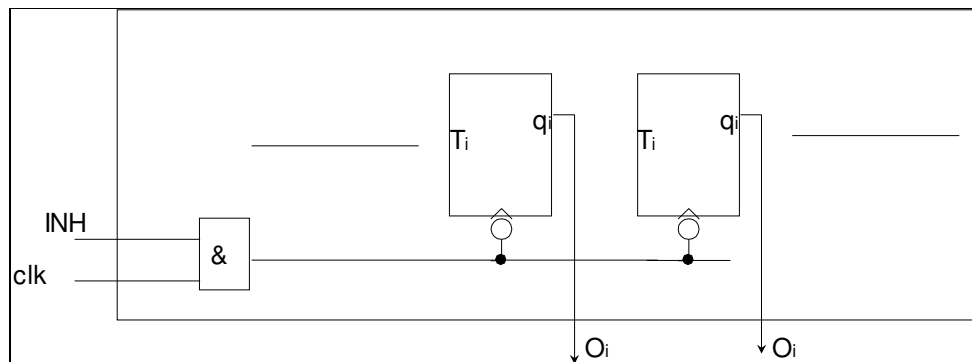
Preset síncrono

Es similar al Clear síncrono, pero en este caso pretendemos poner a 1 todos los biestables. Si $q_i = 1$, T_i debe ser cero, para que en el siguiente ciclo se mantenga el valor de q_i , en cambio si $q_i = 0$, T_i debe ser 1, para que $Q_i = 1$ en el siguiente ciclo de reloj. Por tanto $T_i = q_i'$.



2.1.2 Inhibición

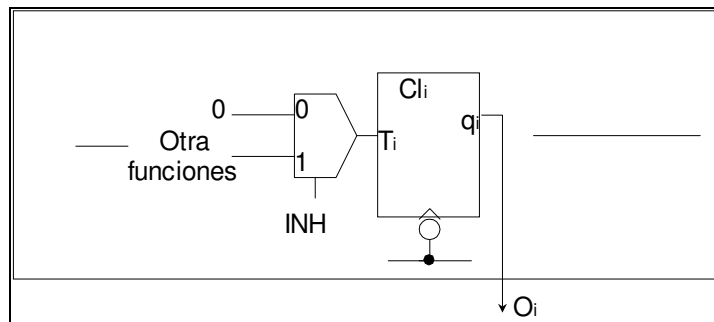
Es una línea de control (activa en alta o en baja) cuya misión es detener el proceso de cuenta del contador. Esto implica que mientras que la línea esté activa, aunque se reciban flancos activos por la señal de reloj del contador, este no modifica su estado de cuenta. Podemos encontrar varios diseños alternativos para implementar la función de inhibición del contador. Uno de ellos es el mostrado por la siguiente figura, donde la entrada de reloj del contador pasa a través de una puerta AND antes de distribuirse por las entradas de reloj de cada uno de los biestables que forma parte de dicho contador. La otra entrada de la AND está formada por la línea de control INH. Si esta vale 1, el reloj se distribuye a todos los biestables, pero sin INH vale 0, se distribuye un 0 a los biestables, por lo que estos no pueden cambiar de estado al no recibirse flancos por sus entradas de reloj.



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 el valor de cuenta del contador. Para el caso de la figura anterior, los biestables del contador son disparados por flanco de bajada y la señal de inhibición es activa en bajo. Llamemos clk' a la señal de reloj que reciben los biestables del contador, esto es, la salida de la puerta AND. Si $INH=1$, $clk'=clk$, funcionamiento normal. Si suponemos que $clk=1$ (nivel alto de reloj, y entonces $INH=0$, lo lógico es que el contador, desde ese mismo momento, no debe de cambiar de estado, pero lo

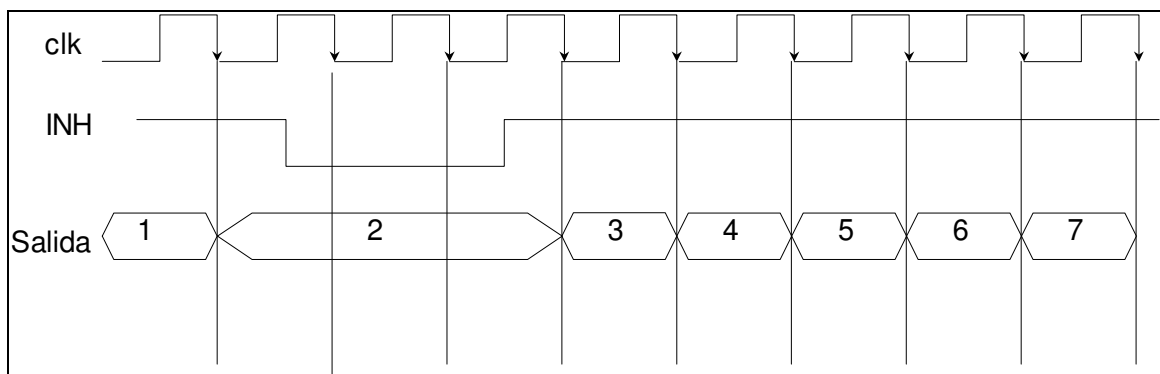
que ocurre es que clk' pasa de valer 1 a valer 0, esto es, genera un flanco de bajada que provoca que los biestables cambien de estado incrementando la cuenta del contador.

La forma más habitual de implementar la función de inhibición es la que se describirá a continuación y que suele denominarse como síncrona. Aquí suponemos que la señal de reloj del contador llega, sin obstáculos, a todos los biestables que lo forman. Por tanto, se trata, ahora, de determinar qué deben tener las entradas T_i de los biestables del contador para que este mantenga el estado de cuenta. Que el contador mantenga su valor de cuenta es equivalente a que cada uno de los biestables que lo forman mantengan su bit cuando INH está activo. Evidentemente la solución es simple, $T_i = 0$ si INH está activa. La siguiente figura muestra la estructura de la celda básica de un contador que ha implementado este tipo de inhibición cuando esta es activa en bajo.



Como se aprecia en la figura, si $INH=1$, la entrada T_i se corresponderá con otras funciones a realizar por el biestable, clear, cuenta,...

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



2.1.3 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. Evidentemente el contador debe de disponer de algunas líneas de entrada adicionales por las que se introduce el dato a cargar, en caso de que se proceda con esta operación. Existen tantas líneas de datos como bits tenga el contador. Cada línea de datos contiene un bit del dato que será cargado en un biestable del contador. La línea de Load puede ser activa en alta o en baja y tiene dos modos de funcionamiento, asíncrono y síncrono.

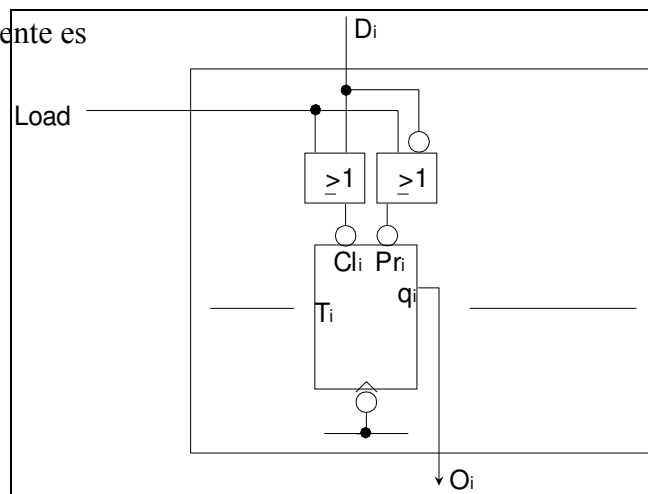
Load asíncrono

Esto se consigue mediante la activación de las señales asíncronas de Pr y Cl de cada uno de los biestables del contador. Reduzcamos el problema a una sólo etapa y supongamos que Load es activo en baja. Si Load = 0 y el bit a escribir en la etapa i es 0, se deben activar las entradas $Cl_i = 0$ y $Pr_i = 1$. Si Load = 0 y el bit a escribir es 1, entonces $Cl_i = 1$, y $Pr_i = 0$. Si Load = 1, $Cl_i = Pr_i = 1$. Si representamos estos valores en un K-mapa y simplificamos, obtenemos las siguientes expresiones para Cl_i y Pr_i

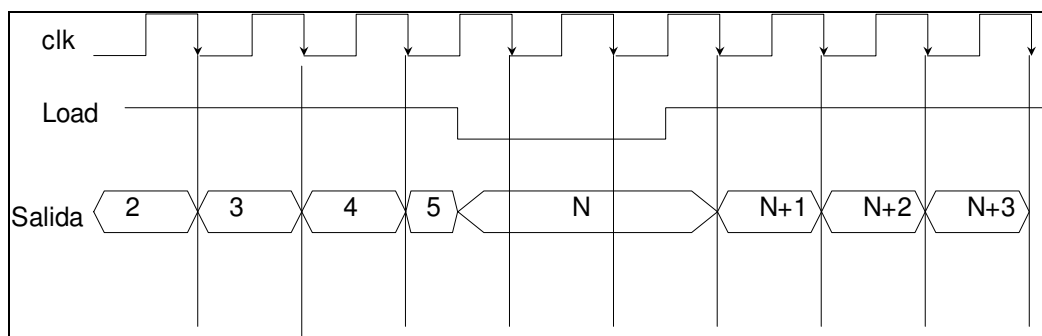
$$Cl_i = Load + D_i$$

$$Pr_i = Load + D_i'$$

El circuito equivalente es



La siguiente figura ilustra el funcionamiento de un contador que ha implementado la operación de Load asíncrono. Como se observa, en el momento que se activa Load, se produce la carga del dato, simbolizado como el número N.

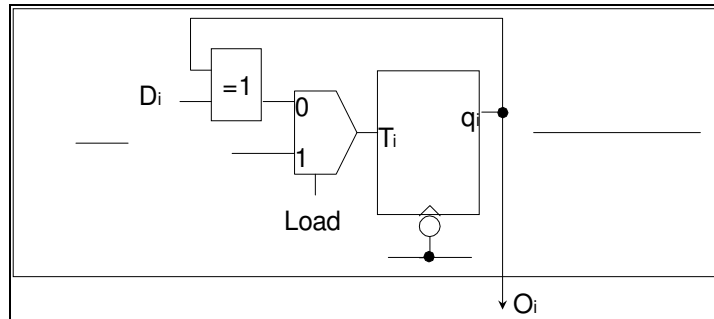


Load síncrono

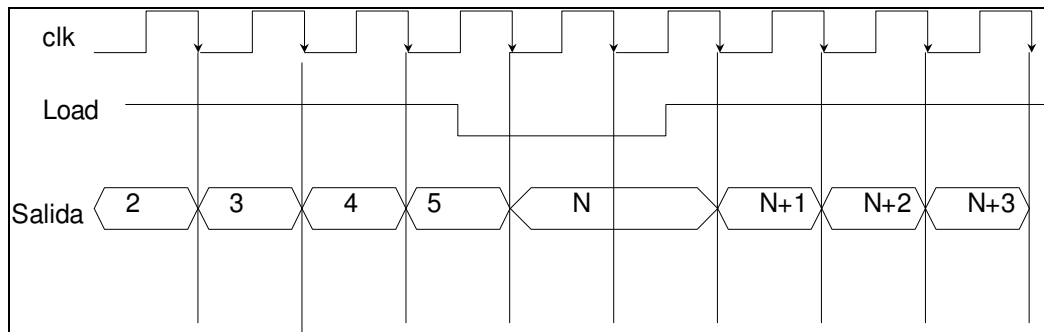
En el load síncrono se trata de determinar cuál debe ser la entrada del biestable T_i para que este se cargue con el valor del dato D_i de entrada. Si suponemos que Load es activo en baja, tenemos que si Load=0, si $D_i = 0$ y $q_i = 0$, entonces T_i debe ser 0, si $D_i = 0$ y $q_i = 1$, T_i debe ser 1 para forzar en el siguiente ciclo de reloj que la salida se ponga a 0. De forma equivalente si $D_i = 1$, y $q_i = 1$, entonces T_i debe ser 0 y si $D_i = 1$ y $q_i = 0$ T_i debe ser 1. En resumen, para que q_i sea igual al dato D_i que queremos cargar debemos comparar q_i con D_i , si los dos son iguales, T_i debe ser cero para no realizar ningún cambio, pero si D_i y q_i son distintos, T_i debe ser 1 para que en el siguiente ciclo, q_i tome el valor del bit a cargar.

$$T_i = q_i \oplus D_i$$

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



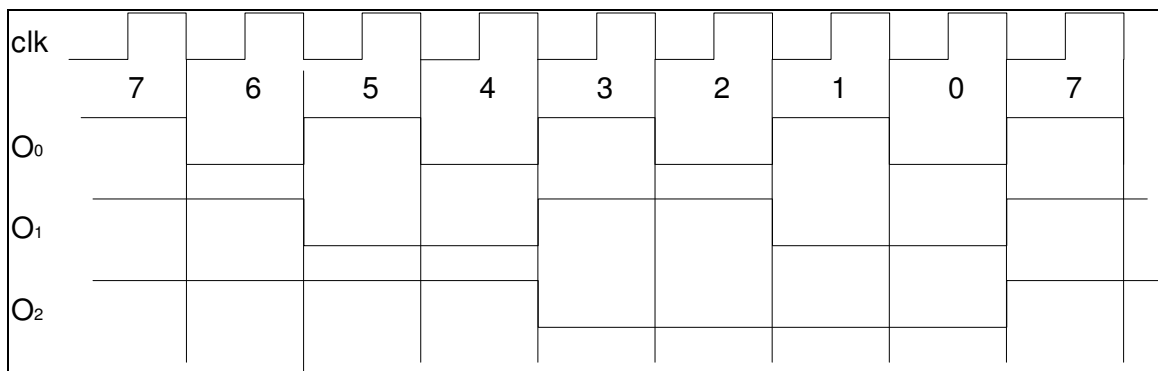
En la siguiente figura se muestra un cronograma donde se observa el funcionamiento de este tipo de operación.



2.1.4 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. En primer lugar se determinará la estructura de un contador descendente, y posteriormente se unirá con la del ascendente, introduciendo la señal de control UP/DOWN, que controla el sentido de la cuenta

La siguiente figura muestra la salida de un contador descendente de módulo 8.



Procedemos de forma similar al ascendente.

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 $T_2=q_1' q_0'$.

Si en lugar de un contador módulo 8, disponemos de un contador de n bits, la expresión de la entrada de cualquier biestables de dicho contador sería

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

Se puede comprobar que la expresión es muy similar a la del ascendente salvo que aquí, los términos van complementados.

De forma equivalente, la expresión anterior se puede escribir como

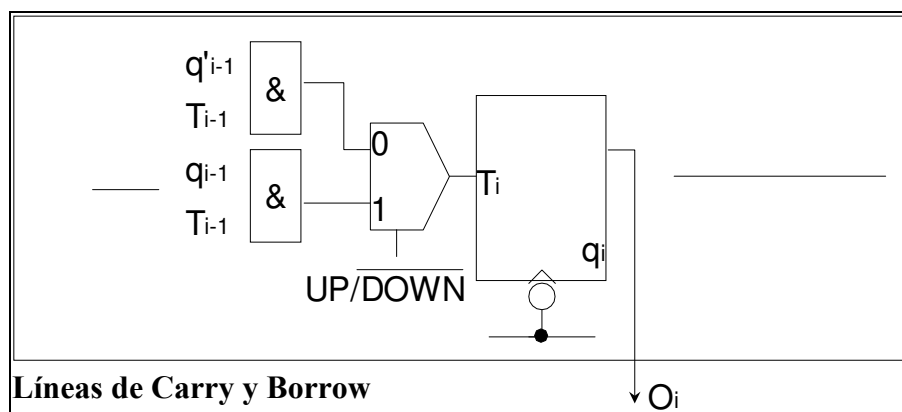
$$T_i = q_{i-1}' T_{i-1}$$

Ya estamos en condiciones de diseñar el contador reversible. Estos disponen de una señal de control UP/DOWN, que indica el sentido de la cuenta. Si UP/DOWN' está a 1, cuenta ascendente, si UP/DOWN está a 0, cuenta descendente.

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

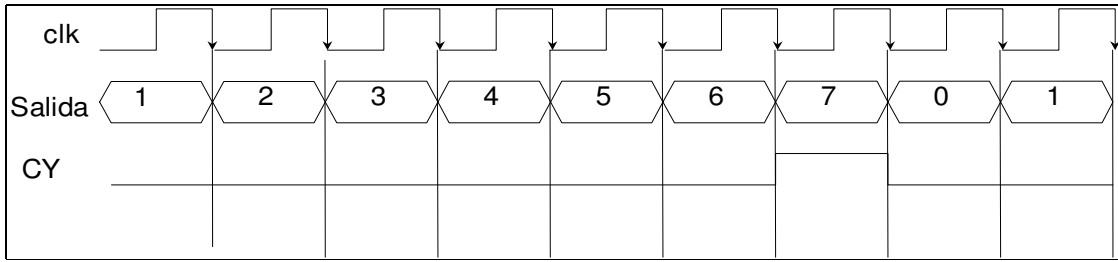
$$T_i = q_{i-1} T_{i-1} \text{UP/DOWN} + q_{i-1}' T_{i-1} \text{UP/DOWN}'$$

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

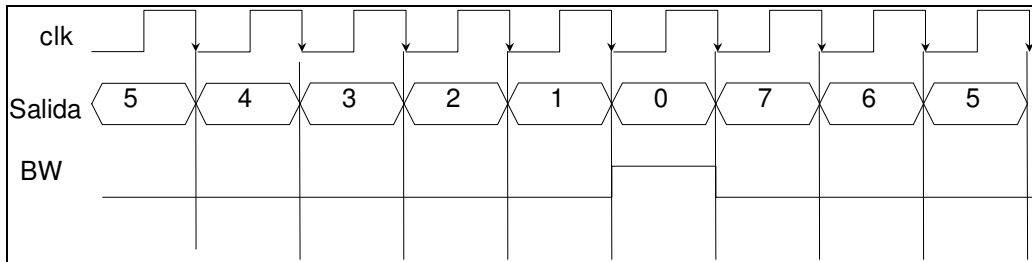


2.1.5 Líneas de Carry y Borrow

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). La salida CY se pone a 1 cuando el contador ascendente alcanza su estado de cuenta más alto. Por ejemplo, si el contador es de módulo 8 la salida CY se pone a 1 cuando el contador llega al estado 7 (ver la siguiente figura).



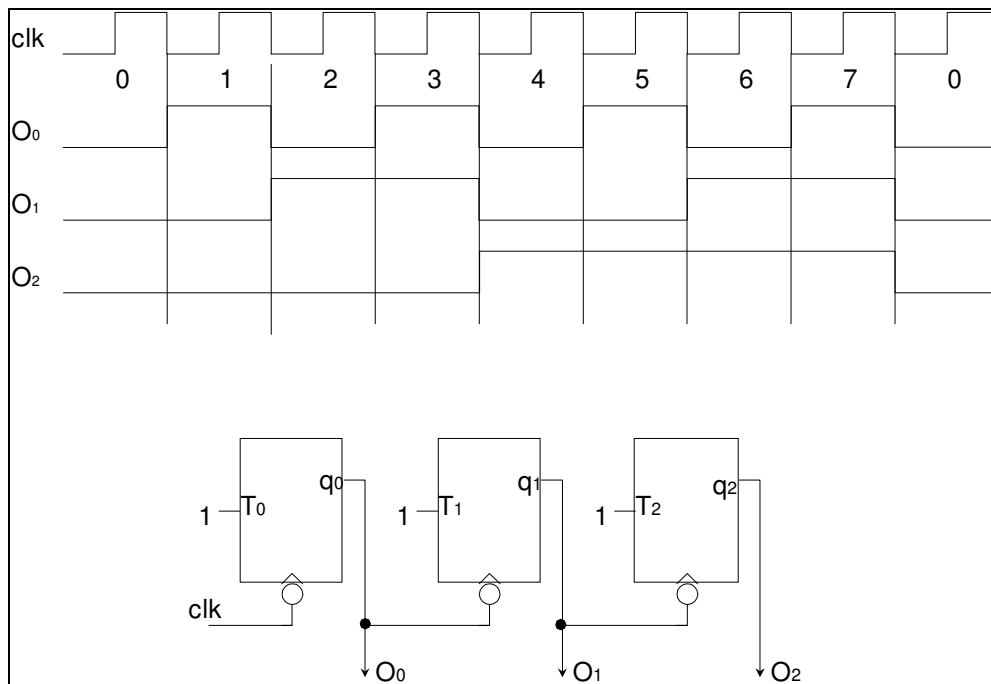
En cambio la señal de Borrow se activa cuando el contador descendente alcanza su estado de cuenta más bajo, el 0. (ver siguiente figura)



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)

2.2 Contadores asíncronos

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



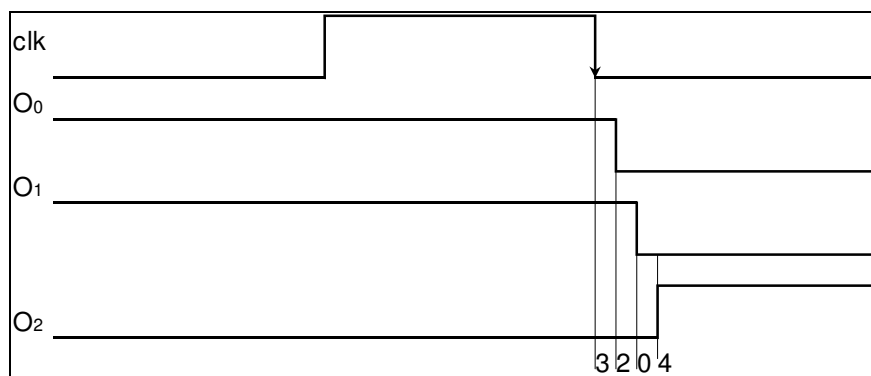
Según se desprende de las formas de onda de salida del contador:

O_0 cambia en los flancos de bajada de clk , por lo que el biestable 0 tiene su entrada T igual a 1 y su entrada de reloj igual a la señal de reloj clk

O_1 es a O_0 como O_0 es a clk . Esto es, si O_0 fuese considerado como una señal de reloj, entonces la entrada del biestable T puede estar a 1. Con esto conseguimos que este biestable siempre cambie en los flancos de bajada de O_0 .

De igual forma, O_2 cambia en los flancos de bajada de O_1 , por lo que la entrada de reloj del biestable 2 es O_1 y su entrada T es un 1.

Este tipo de diseño alternativo se denomina también contador de rizado o ripple-counter, por la especie de rizo que hace la conexión de la salida de un biestable a la entrada de reloj del siguiente biestable. La estructura de este contador es más simple que la del sincrónico, por no necesitar de puertas lógicas adicionales, sin embargo presenta algunos inconvenientes, como la velocidad de operación (que es menor en este caso) y la aparición de estados de cuentas fantasmas. Analicemos este último aspecto. Supongamos la situación realista de que los biestables del contador de rizado anterior, tienen un tiempo de propagación no nulo, igual a t_p . Como se muestra en la siguiente figura, el contador va a pasar del estado de cuenta 3 al 4. Al usarse la salida de un biestable como reloj del siguiente, los retrasos van acumulándose de una etapa a otra, por lo que la salida de cuenta verdadera, tarda un tiempo, en este caso, igual a $3 t_p$. En el intervalo de tiempo comprendido entre 0 y $3 t_p$ se han producido cuentas transitorias erróneas.



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

2.3 Contadores con módulo diferente a la potencia de dos

Es habitual que en la práctica se necesiten contadores con un módulo diferente a la potencia de dos (en el mercado es muy común encontrarse con contadores módulo 10 por la importancia que tiene esta base de numeración). Si se desea construir un contador con un módulo distinto a lo que se ofrece comercialmente, tenemos dos posibilidades:

- Diseñarlo con biestables y puertas, como si se tratara de una máquina secuencial síncrona

b) 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.

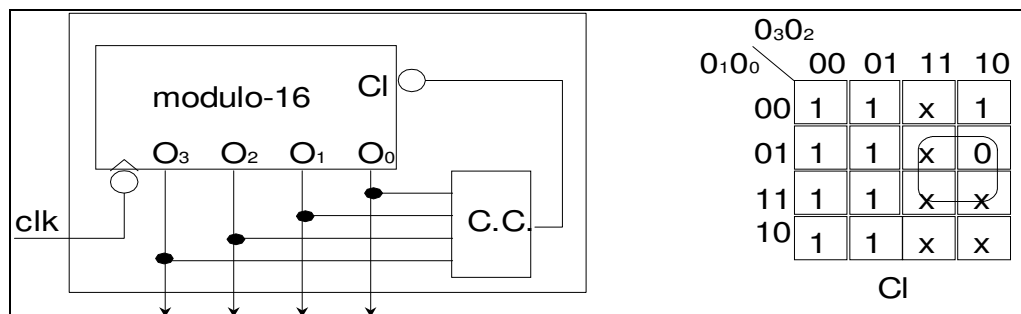
Aquí se desarrollará el método indicado por el apartado b), remitimos al alumno al tema 7 para el diseño por el apartado a).

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

Es lógico que el contador módulo 16 usado para implementar el módulo 10 debe interrumpir su cuenta cuando llega al estado de cuenta 9. Esto es, el contador módulo 16 pasaría, después del 9, al estado de cuenta 10, pero debemos obligarle a que, en lugar del 10, pase al estado de cuenta 0. Esto es posible si el contador dispone de línea de Clear. De alguna manera, esta línea debe de activarse cada vez que sea necesario para forzar el paso del estado de cuenta 9 al 0. Existen dos alternativas de diseño, en función de que la línea de Clear tenga un modo de funcionamiento asíncrono o síncrono. Analizaremos las dos situaciones.

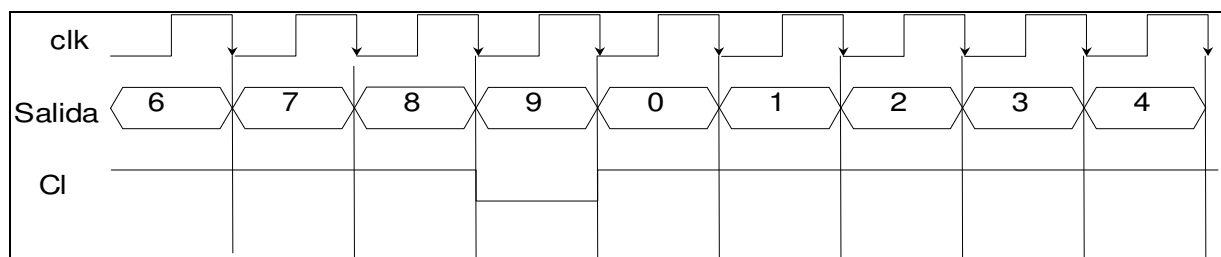
→ Con clear síncrono (activo en bajo)

Aquí se plantea diseñar un circuito combinacional que en función del estado de cuenta del contador módulo 16 active la señal de clear. Como esta es síncrona, debe activarse en el estado de cuenta 9, para que el próximo estado, en el siguiente flanco activo de clk, sea 0.



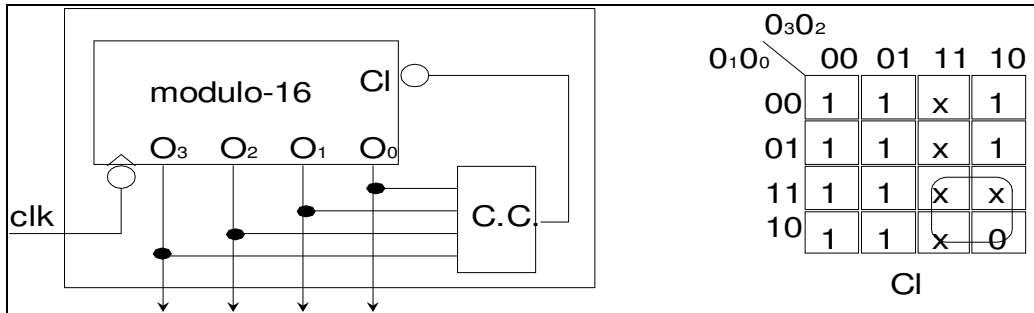
Por tanto $Cl = (O_3 O_0)'$

En la siguiente figura se muestra un cronograma donde se aprecia el funcionamiento del conjunto contador-puerta. Se observa que en el estado de cuenta 9, la señal de Cl se pone a 0, lo que obliga a que el contador se ponga a 0 en el siguiente ciclo de reloj.



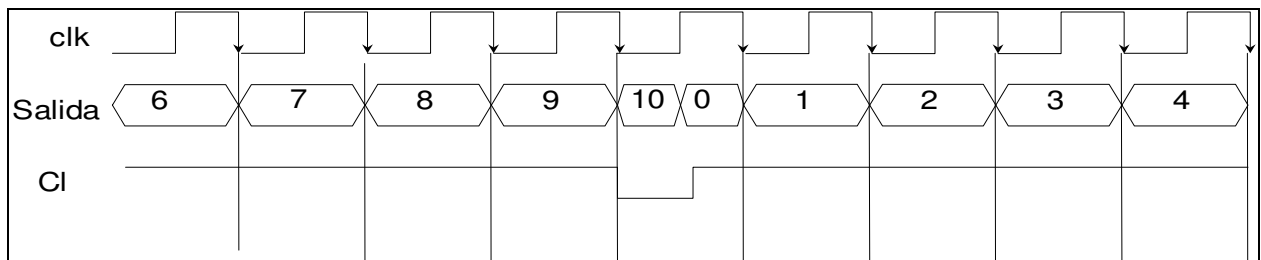
→ Con clear asíncrono (activo en bajo)

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. Una vez que el circuito combinacional detecta la llegada de este estado, activa la señal de clear, lo que provoca que inmediatamente el contador se ponga a cero (sin esperar el flanco activo de clk).



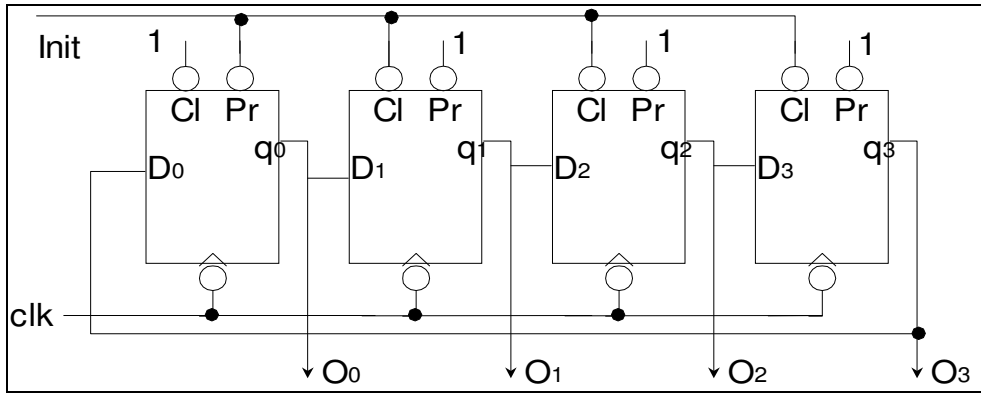
Por tanto $Cl = (O_3 O_1)'$

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.

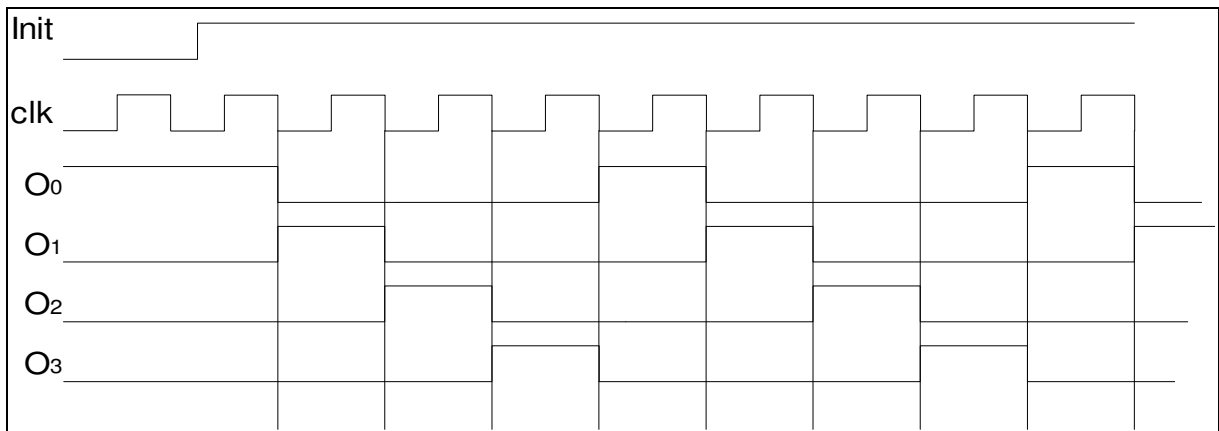


2.4 Contadores de anillo y conmutado en cola

La siguiente figura muestra la estructura de un contador en anillo de módulo-4. Este está constituido por 4 biestables, conectados entre sí de modo que la salida de uno es la entrada del siguiente y así sucesivamente hasta llegar al último biestable, cuya salida se reintroduce por la entrada del primero. Esto da idea del nombre de contador en anillo. En este contador, el estado de cuenta j , viene determinado por un 1 en la salida O_j .

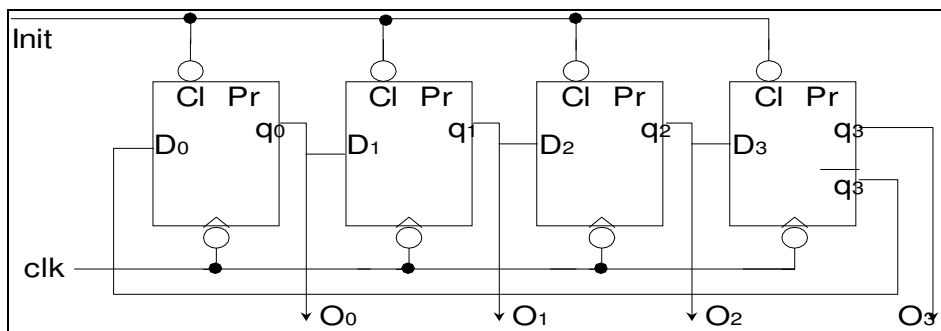


La línea Init sirve para inicializar el circuito, de forma que cuando está vale 0, los biestables 0,1,2,3 toman, de forma asíncrona, los valores 1,0,0,0 respectivamente. La puesta a 1 de Init sirve como indicador para la puesta en marcha del contador. La siguiente figura muestra el cronograma temporal del funcionamiento de este contador.



En el ciclo de reloj posterior a la inicialización, el biestable 1 captura el 1 que tiene el biestable 0, mientras que este, junto con los biestables 2 y 3, se ponen a cero. En el siguiente ciclo, el biestable 2 captura el 1, los restantes están a cero, y así sucesivamente. En definitiva, existe un único 1 que en cada ciclo de reloj va pasando de un biestable a otro del contador.

Una modificación al contador anterior, la constituye el anillo de Johnson o conmutado en cola, el cual permite un número mayor de estados. En concreto, si este nuevo contador tiene n biestables, el número de estados de cuenta es de $2 \cdot n$. Sin embargo, este contador, a diferencia del de anillo, no muestra de forma evidente el estado de cuenta. La siguiente figura muestra la estructura de un contador de anillo conmutado en cola o contador de Johnson.

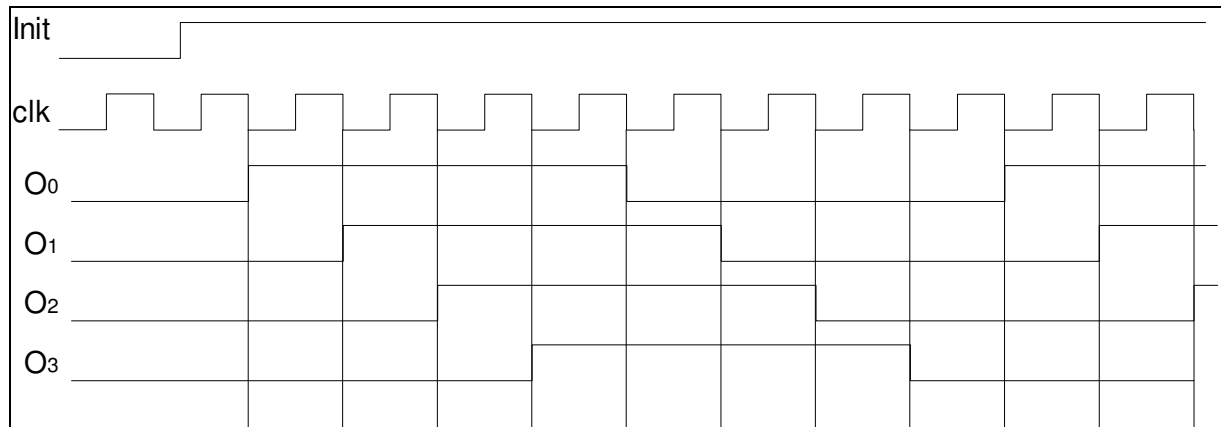


Son dos las diferencias con el contador en anillo:

- 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)

Las salidas de este contador se muestra en la siguiente figura. Si inicialmente todos los biestables están a cero, en el siguiente ciclo de reloj, el biestable 0, se carga con un 1, ya que

su entrada es el complemento del contenido del biestable situado en cola. En el siguiente ciclo de reloj este 1 pasa al biestable 1, mientras que el biestable 0 sigue cargándose con un 1 (mientras que $q_3=0$). Este proceso se repite hasta que por fin, todos los biestables están a 1. En el siguiente ciclo de reloj, el biestable 0 se carga con un 0 (al ser $q_3=1$). A continuación este cero pasa al biestable 1, en el siguiente ciclo de reloj, mientras que el biestable 0 sigue cargándose con un 0. Esto se va repitiendo hasta que finalmente todos los biestables se encuentran a 0, el punto de partida. A partir de aquí se repite toda la secuencia. Se ve claramente, que todos los estados posibles son 8, justamente el doble del número de biestables.



3. 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.

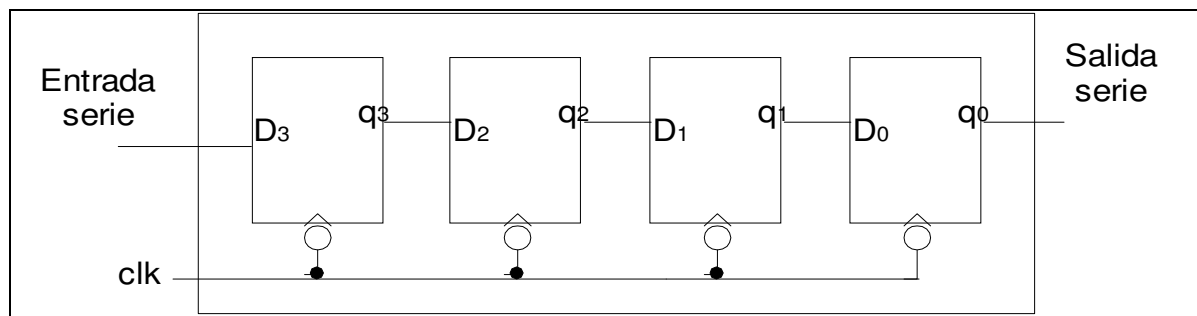
Si disponemos de un registro de n bits, diremos que si este tiene entrada serie, entonces el registro sólo tiene una línea de entrada de datos por la que, uno tras otro, se introducen los n bits que serán almacenados por el registro. Si un registro de n bits se dice que tiene entrada

paralelo, entonces dicho registro dispone de n líneas de entrada, una por cada bit, por la que se introducen simultáneamente los n bits al registro. Estos conceptos de entrada serie y entrada paralelo son igualmente aplicables para la salida serie y salida paralelo. Un registro de n bits tiene salida serie, implica que tiene una única línea de salida por la que, uno tras otro, van saliendo los n bits almacenados en el registro. Si un registro de n bits tiene salida paralelo, implica que dispone de n salidas, una por cada bit, de forma que simultáneamente se leen todos los bits del registro.

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

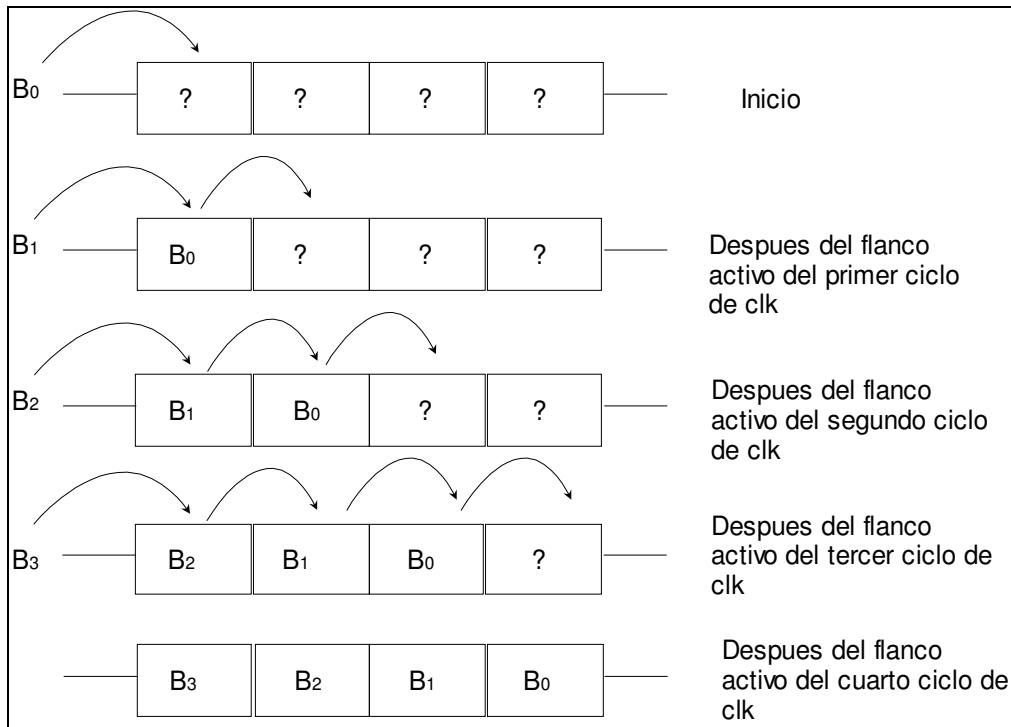
La estructura básica de un registro de 4 bits de entrada serie y salida serie se muestra en la siguiente figura.



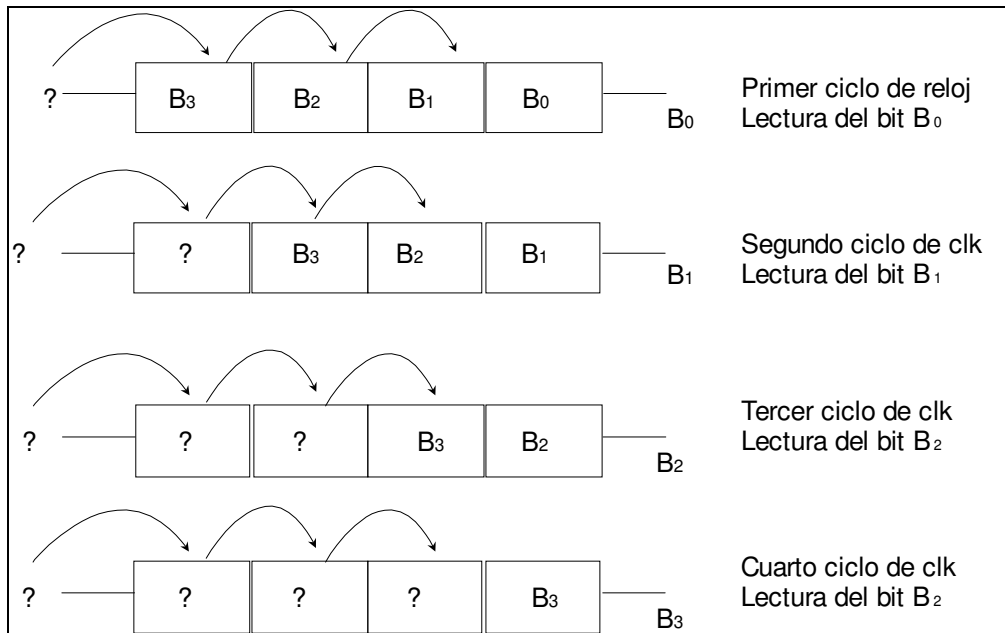
Para todos los registros de desplazamiento es necesario que los biestables D que lo constituyen sean disparados por flanco. Sólo se salva de esta regla el registro de entrada paralelo y salida paralelo, el cual suele estar formado por biestables disparados por nivel.

Se observa en el registro de la figura anterior, que los biestables están conectados de forma que la salida de cada uno se corresponde con la entrada del biestable situado inmediatamente a su derecha. Asimismo, la salida del biestable situado a la derecha del conjunto se corresponde con la salida del registro serie, mientras que la entrada para el biestable situado a la izquierda del conjunto se corresponde con la entrada de datos del registro.

Para que el registro opere de forma adecuada, los bits de entrada deben estar sincronizados con la señal de reloj, esto es, un bit de entrada por cada ciclo de reloj. El primer bit, B₀, de entrada se escribirá en el biestable D₃ en el primer ciclo de reloj, el segundo bit de entrada, B₁, vuelve a escribirse en D₃, encima de B₀, el cual pasa, al biestable D₂. El tercer bit de entrada, B₂, se escribe nuevamente en D₃, encima de B₁, el cual pasa al biestable D₂, sobrescribiendo el bit B₀, que, a su vez, pasa a D₁. Así se completa la secuencia de escritura para los cuatro bits.



Para la lectura serie el procedimiento es similar. Sólo hay que tener en cuenta que para leer el primer bit almacenado, el B_0 , no es necesario esperar ningún ciclo de reloj, ya que la salida q del último biestable, (y por tanto el bit B_0), se encuentra conectada con la salida del registro. La siguiente figura ilustra el proceso de lectura en cada ciclo de reloj.

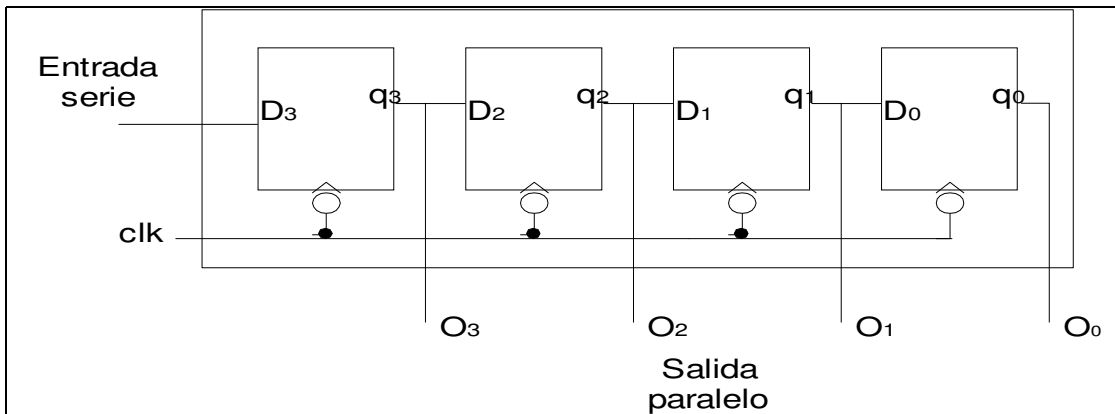


Por todo lo visto, podemos decir que el registro de entrada serie y salida serie presentado en este apartado es un registro de desplazamiento a derecha. El registro de desplazamiento a izquierda se hubiera construido de forma similar sin más que conectar la salida de un

biestable con la entrada del biestable situado a su izquierda. La entrada del registro sería en este caso D_0 , y la salida, q_3 .

3.2 Registro con entrada serie y salida paralelo

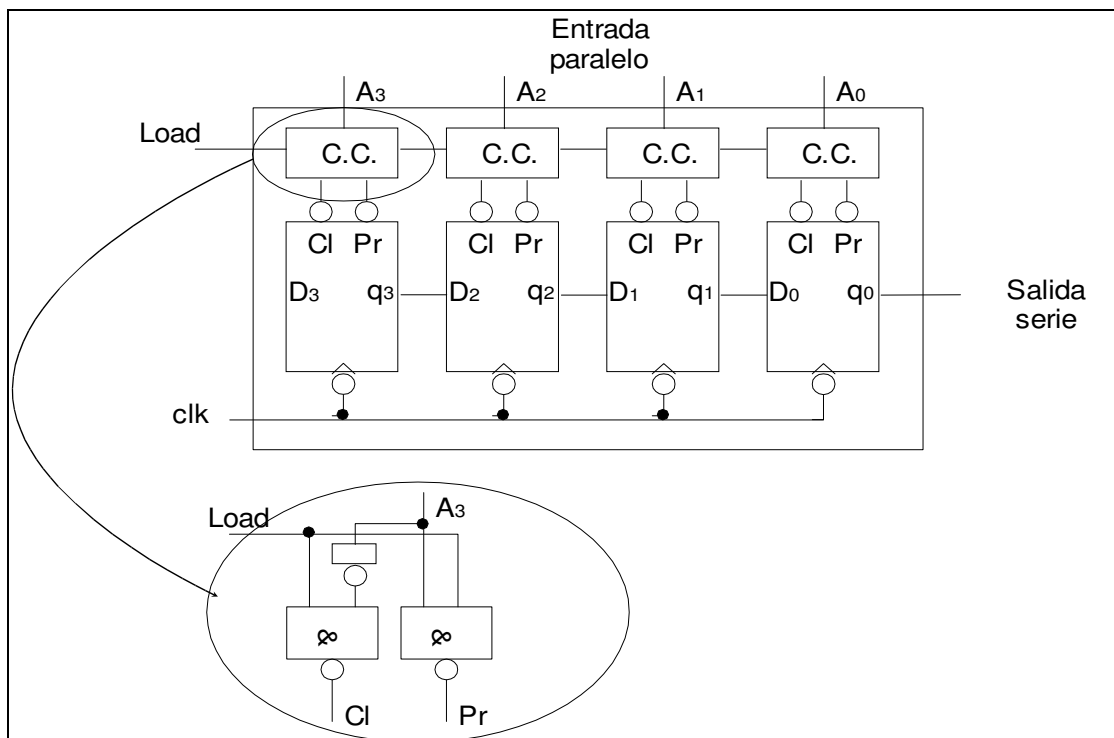
La siguiente figura muestra la estructura básica de un registro de entrada serie y salida paralelo de 4 bits. Al igual que en el apartado anterior, este es un registro de desplazamiento a derecha



El procedimiento de escritura es el mismo que el descrito en el apartado anterior. En cuanto a la lectura, se hace evidente que el registro no depende de la señal de reloj para mostrar su contenido, este siempre aparece en las líneas de salida.

3.3 Registro con entrada paralelo y salida serie.

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

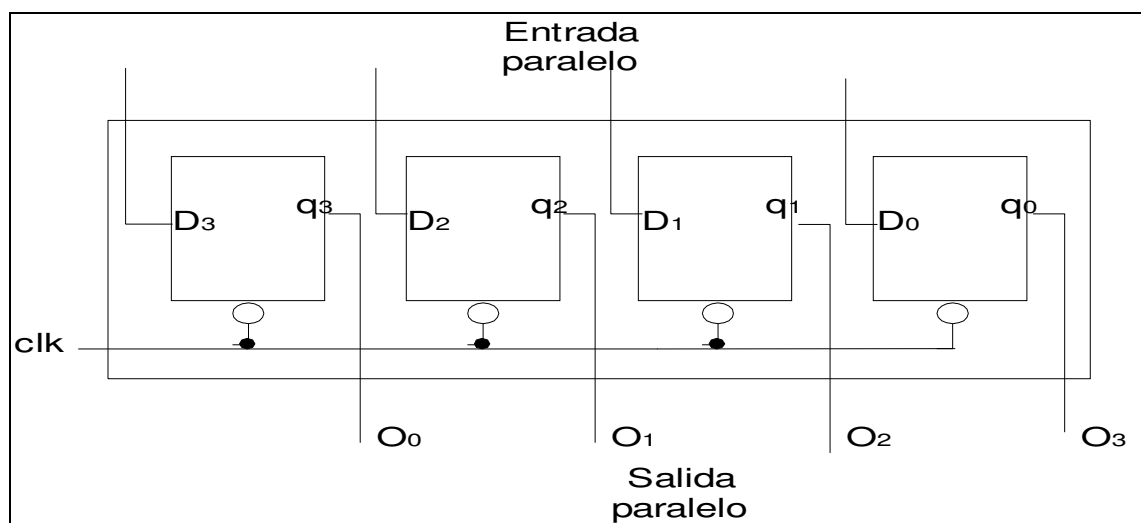


El procedimiento de lectura es igual al descrito en el apartado 3.1, para lo que se hace necesario que los biestables esten conectados entre sí, salida de uno con la entrada del siguiente. (Se comprueba que el registro de desplazamiento es nuevamente a la derecha). Aquí se modifica el proceso de escritura, que se realiza cuando la línea Load está activa.

En la escritura en paralelo, todos los bits de entrada A_i , se cargarán simultáneamente en los biestables q_i . En la figura se ha escogido un procedimiento de carga asíncrona, usando las entradas de Clear y Preset de los biestables. (Se deja al estudiante la estructura para un procedimiento de carga o escritura síncrona). Se ha diseñado un circuito combinacional, formado por dos puertas NAND y un inversor, que controla las líneas de Cl y Pr de cada biestable en función del bit a escribir, A_i , y de la señal de control Load. Cuando esta última está a 0, las entradas Cl y Pr están a 1, no ocurre nada. Si $Load = 1$, entonces si $A_i = 0$, se activa Cl y si $A_i = 1$, se activa Pr, esto es, escritura asíncrona de un 0 o un 1 respectivamente. Haciendo extensible este proceso de escritura de un biestable a todos los biestables que constituyen el registro, tenemos el proceso de escritura del mismo.

3.4. Registro con entrada paralela y salida paralela.

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



Como se observa, los biestables son del tipo disparado por nivel bajo en lugar de disparado por flanco. Esto suele ser habitual en este tipo de registros, denominados también como LATCH, aunque no se descarta la posibilidad de que se puedan diseñar con biestables disparados por flanco.

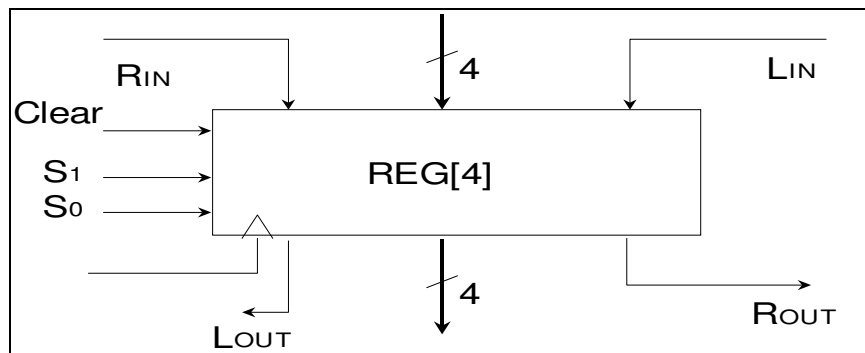
Si la señal clk vale 0, los biestables capturan simultáneamente todos los bits de entrada. Si clk vale 1, los biestables mantienen la información capturada que muestran siempre por sus líneas de salida.

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.

De las cuatro operaciones anteriores, las de desplazamiento a derecha y a izquierda, forzosamente tienen que ser síncronas. La carga y el clear pueden ser síncronas o asíncronas. Supongamos que clear es asíncrono y la carga es síncrona. Tenemos un total de 4 operaciones a realizar más la de NOP (no operate) o inhibición que debe estar presente en cualquier diseño de registro ya que esta es la equivalente a mantener la información. Está claro que para implementar la operación asíncrona se necesita que los cuatro biestables que forman parte del registro tengan entradas asíncronas, en este caso de Cl , la cual se conecta directamente con la entrada de CLEAR del registro. Las otras cuatro operaciones síncronas, se van a codificar en dos líneas de control S_1, S_0 , tal como se muestra en la siguiente tabla.



$S_1 S_0$	Operación
00	SHR (Desplazamiento a la derecha)
01	Load
10	NOP o INH
11	SHL (Desplazamiento a la izquierda)

Para la escritura o carga en paralelo, el registro ha de disponer de cuatro líneas de entrada. Para la lectura en paralelo, el registro dispone de cuatro líneas de salida. Para la lectura/escritura serie a la derecha, el registro dispone de dos líneas, R_{in}, R_{out} y para la lectura/escritura a la izquierda, el registro también dispone de las líneas L_{in}, L_{out} . Además se necesita la entrada de reloj.

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 B0,B1,B2 y B3

$$D_3=B3$$

$$D_2=B2$$

$$D_1=B1$$

$$D_0=B0$$

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

$$D_3= q_2$$

$$D_2= q_1$$

$$D_1= q_0$$

$$D_0=Lin$$

Y la salida L_{out} está formada por q_3

En cualquier caso, las salidas $O3-0$ son los valores de los biestables $q3-0$

En general podemos decir que

$$D_3= S_1'S_0'R_{in} + S_1'S_0 B3 + S_1 S_0'q_3 + S_1 S_0 q_2$$

$$D_2= S_1'S_0'q_3 + S_1'S_0 B2 + S_1 S_0'q_2 + S_1 S_0 q_1$$

$$D_1= S_1'S_0'q_2 + S_1'S_0 B1 + S_1 S_0'q_1 + S_1 S_0 q_0$$

$$D_0= S_1'S_0'q_1 + S_1'S_0 B0 + S_1 S_0'q_0 + S_1 S_0 L_{in}$$

4. PLD's SECUENCIALES

Se estudió en el capítulo 5 la estructura de diversos dispositivos programables, del tipo PLA y PAL. Los PLD's secuenciales se basan en dichas estructuras, con la novedad de que incorporan biestables en los que sus salidas q son reintroducidas al array programable y cuyas entradas pueden ser programadas. Usando PLD's podemos implementar funciones de conmutación combinatoriales y secuenciales con un número de estados que depende de la cantidad de biestables disponibles en el propio PLD.

En la página 27bis aparece la estructura básica de un PAL secuencial, el cual incorpora cuatro biestables D del tipo disparado por flanco. Dicho PLD dispone de 8 entradas de datos I_{0-7} , una entrada de reloj que se distribuye a todos los biestables del PLD, una entrada de habilitación de salida, o Output Enable, que habilita las salidas de los valores almacenados en los biestables, cuatro salidas B_{6-7} y B_{1-0} que pueden configurarse como entradas y cuatro salidas correspondientes a las q de los biestables.

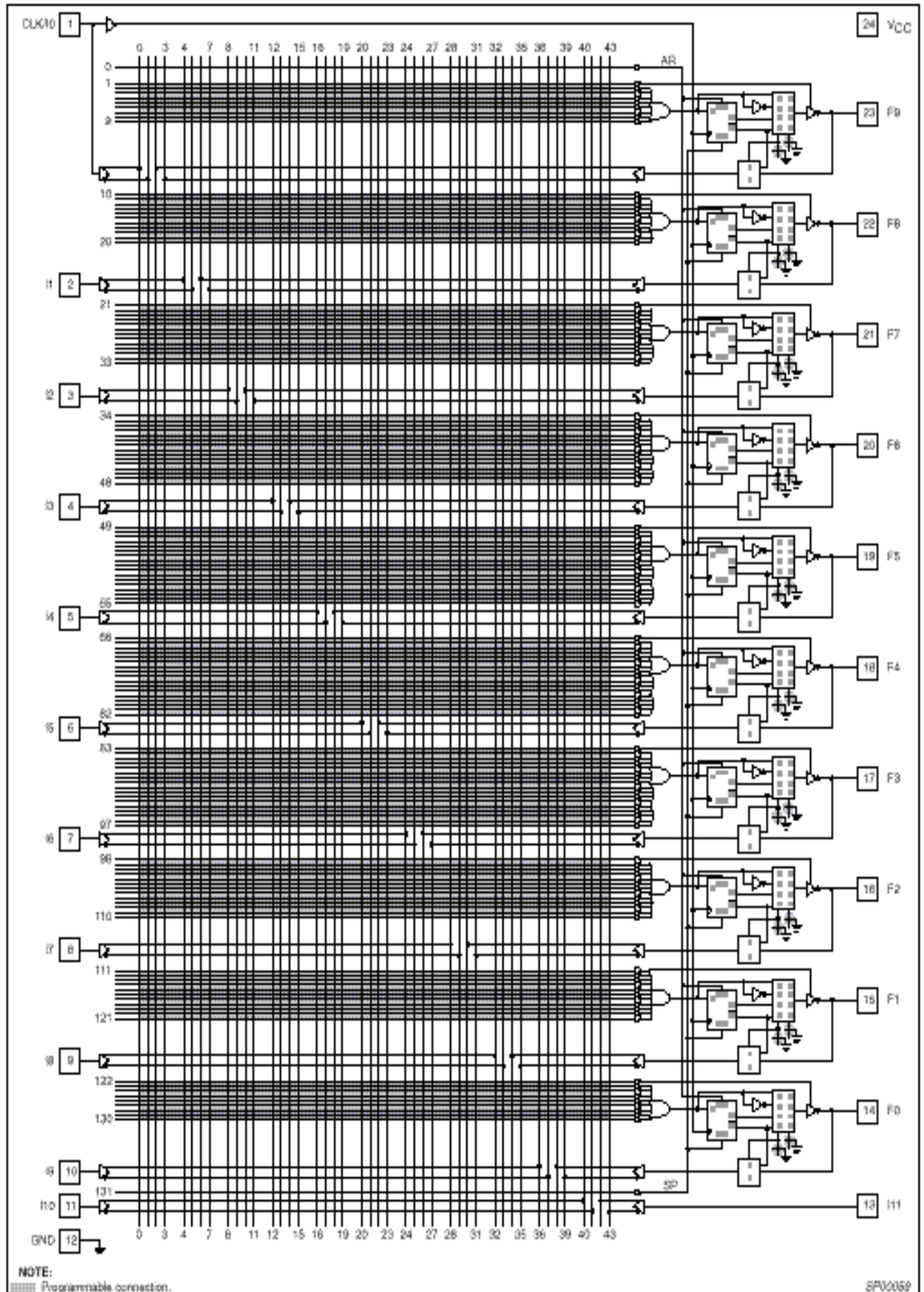
La entrada OE se utiliza como habilitador de un grupo de cuatro inversores triestado, de forma que si $OE=0$, dichos inversores están activos y si $OE=1$, los inversores se ponen en estado de alta impedancia.

Las salidas de los biestables se reintroducen en doble raíl al array programable, de esta forma la PAL puede generar términos producto que contengan tanto variables de entrada como valores q de biestables.

Una estructura más eficiente, pero más compleja, de PAL secuencial aparece en la siguiente figura (página 28), donde vemos un conjunto de estructuras repartidas por igual en todas las posibles salidas. En efecto, la PAL anterior, usa estructuras distintas en función de qué salida. Sería mucho más eficiente usar el mismo tipo de estructura (llamada macrocelda) para todas las salidas y que estas tuvieran la capacidad de programarse como se quisiera, es decir, algunas para generar términos combinacionales, otras para usar biestables, etc...

La PAL de la figura tiene 10 entradas fijas I_{1-10} , una entrada I_0 compartida con la señal de reloj CLK, de forma que si la función a implementar en esta PAL no requiere de ningún biestable, se pueda aprovechar el pin de entrada I_0 como una variable más. Todas las macroceldas usan biestables tipo D, disparados por flanco y con entradas asíncronas de Cl (AR en la figura) y Preset (Pr), las cuales se determinan por sendos términos productos el array programable (líneas 0 y 131 de la figura)

LOGIC DIAGRAM



La estructura de la macrocelda aparece en la siguiente figura
 Autor: Alberto Molina

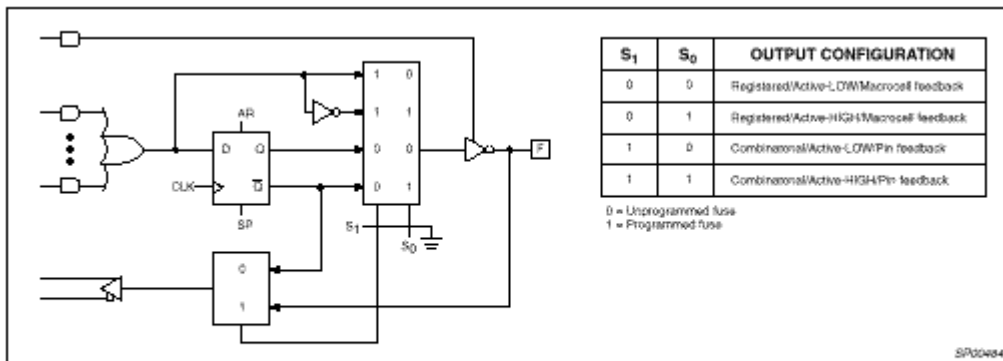


Figure 2. Output Macro Cell Logic Diagram

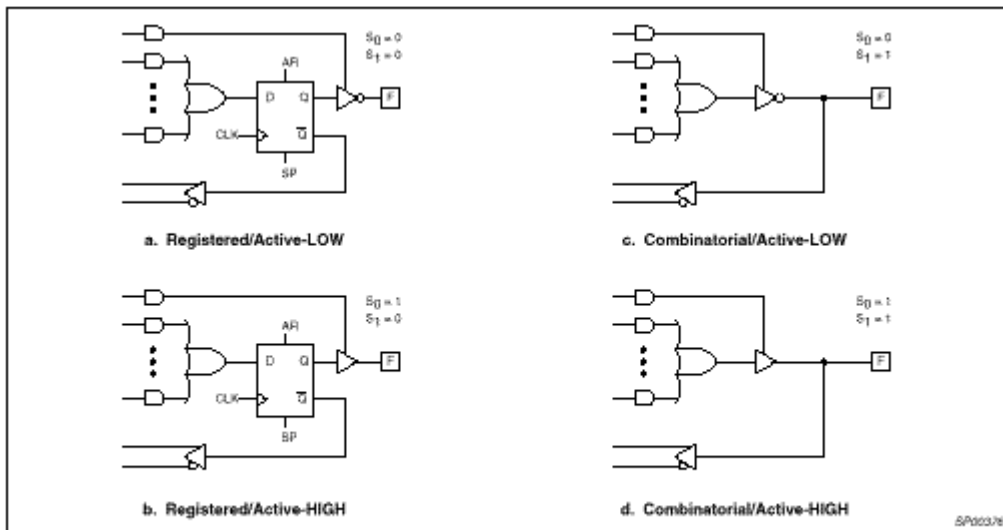


Figure 3. Output Macro Cell Configurations

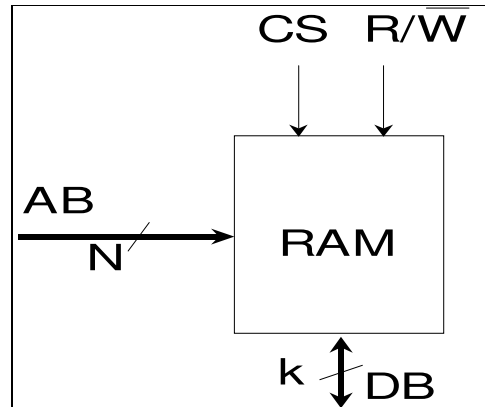
Esta consta de un biestable D, un multiplexor de cuatro canales cuyas entradas de selección son programables, y un multiplexor de dos entradas cuya señal de selección coincide con la de mayor peso de selección del multiplexor de cuatro canales. En función de cómo se programen los dos fusibles contenidos en la macrocelda, tenemos una salida combinacional con buffer triestado o inversor triestado, o bien tenemos una salida secuencial con buffer triestado o inversor triestado con realimentación de la salida q del biestables al interior del array programable.

5. 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.

La RAM tiene un bus de direcciones que selecciona la palabra interna; un bus de datos, en este caso bidireccional, donde se muestran los bits de la palabra seleccionada en operación de lectura, o donde se colocan los bits que serán escritos en la posición seleccionada por el bus de direcciones; y un bus de control que permite seleccionar el tipo de operación a realizar, lectura o escritura y habilitación del chip.

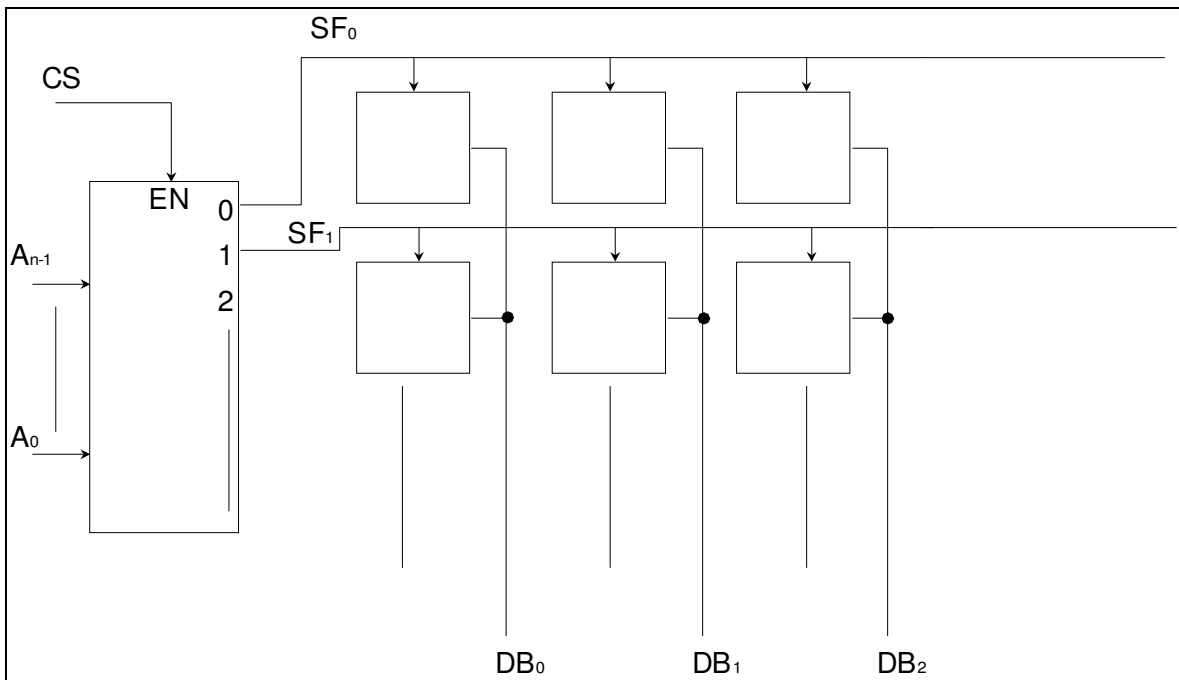
Una RAM con N líneas de dirección y K líneas en el bus de datos permite almacenar $2^N * K$ bits, esto es 2^N palabras distintas de K bits.



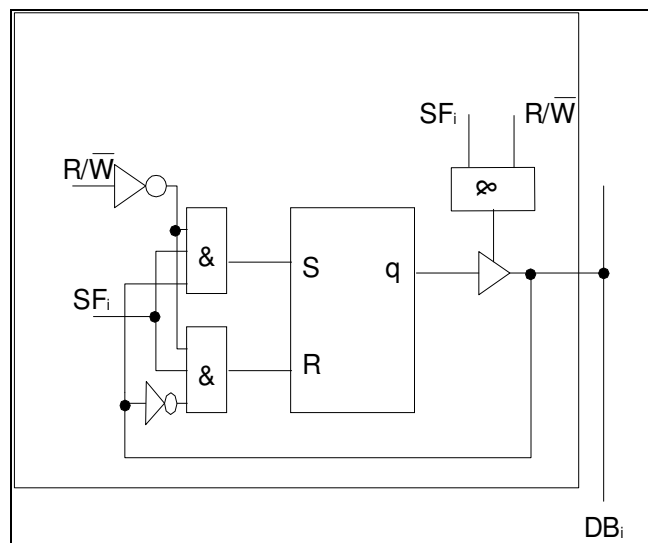
Existen dos tipos de RAM: RAM estática y RAM dinámica

RAM estática (SRAM)

Cada bit es almacenado en un biestable (normalmente del tipo SR). Una posible estructura lógica de la RAM estática es la mostrada en la siguiente figura. Los biestables están contenidos en unos módulos que están organizados por filas y columnas, de forma que existen 2^N filas con K columnas para una RAM de $2^N * K$ bits. La RAM dispone de un decodificador de direcciones que permite seleccionar una fila de módulos. Se ha denominado Sf_i al selector de la fila i . Si el selector de fila Sf_i está activo, todos los módulos de la fila seleccionada pueden acceder a las líneas del bus de datos. Si el selector de fila está inactivo, las salidas de los módulos conectados a dicho selector están en alta impedancia.



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.



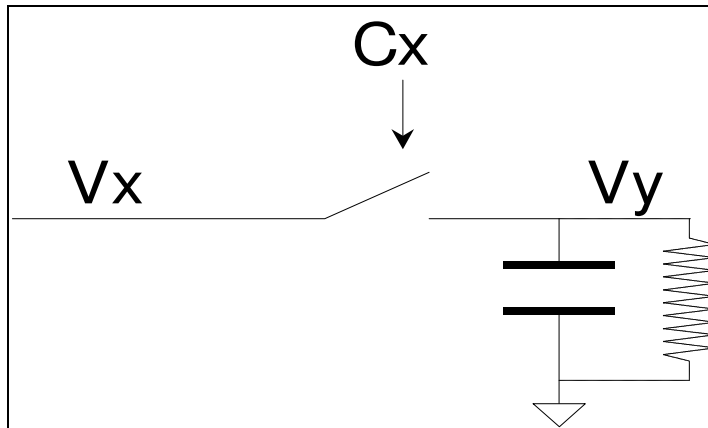
Si $Sf_i = 0$, el módulo está en alta impedancia y las entradas $SR=00$, por lo que el biestable no cambia de valor.

Si $Sf_i = 1$ y $R/W = 1$, las entradas del biestable son $SR=00$, no modifica su contenido, y el buffer triestado está habilitado, por lo que el valor de q pasa a la línea del bus de datos.

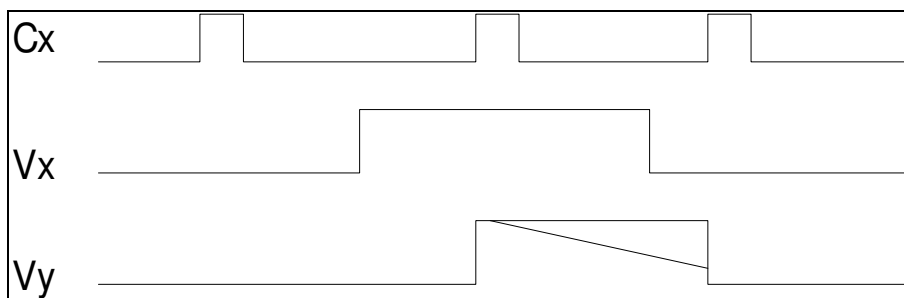
Si $Sf_i = 0$ y $R/W = 0$, la salida q está aislada del bus de de datos que ahora se comporta como entrada, y las entradas del biestable son $SR=10$ o $SR=01$ en función de sí el dato a escribir es un 1 o un 0 respectivamente.

RAM dinámica(DRAM)

En este tipo de dispositivos la información se almacena en condensadores y no en biestables. Su organización interna es similar a la de la RAM estática, desde un punto de vista lógico, pero su celda básica se muestra de forma esquemática en la siguiente figura.



Esta formada por un condensador y una llave electrónica que se abre y cierra en función de una señal de control C_x para programar el bit que será almacenado en el condensador. En la siguiente figura se ha representado la evolución temporal para esta celda, en donde V_x hace referencia a la tensión de entrada con la que se programa la celda y V_y la tensión almacenada en la misma.



Si $C_x=0$, la llave está abierta y el condensador mantiene su información, que en principio es 0volts. Si $C_x=1$, la llave se cierra y el condensador se carga con el valor de tensión de entrada. Por ejemplo, el segundo pulso de C_x , provoca que $V_y=V_x=5$ volts, los cuales deben mantenerse hasta la llegada del tercer pulso de C_x , en cuyo caso, V_y vuelve a ponerse a 0 porque la entrada, V_x , tiene ese valor. No obstante, estas celdas plantean un problema, y este viene representado por la existencia de una resistencia parásita que provoca que el condensador vaya descargándose lentamente (línea punteada). Esto hace que las DRAM no puedan almacenar indefinidamente su información si esta no es refrescada periódicamente.