

## 19 MEMORIAS DE ACCESO DIRECTO

- 19.1. Biestables, registros, pilas y memorias
- 19.2. Configuración de los bloques de memoria de acceso directo
- 19.3. Arquitectura de buses: mapa de memoria
- 19.4. Direccionamiento de un bloque RAM en sectores separados
- 19.5. Memorias de sólo lectura

En el capítulo 11 se considera la memoria de los sistemas secuenciales desde dos puntos de vista complementarios: el estado del sistema y el almacenamiento de datos. Dentro de esta segunda perspectiva, la mayoría de los sistemas digitales (salvo sistemas muy simples o directos) necesitan «memoria» para conservar en ella información (datos y resultados intermedios) que será utilizada posteriormente.

Un biestable **D** es una unidad de memoria capaz de almacenar un bit, un registro de  $n$  biestables tiene capacidad para conservar una palabra de  $n$  bits, una pila es una fila de registros a los que se accede secuencialmente (uno tras otro) y se denomina memoria de acceso directo **RAM** a un bloque de  $m$  registros numerados, con capacidad para  $m$  palabras de  $n$  bits.

Los bloques integrados **RAM** permiten acceder (leer o escribir) a cada registro por medio de su número, a través de un conjunto de líneas de direccionamiento o bus de direcciones. La transferencia de información se lleva a cabo por unas líneas bidireccionales (permiten tanto leer como escribir sobre el registro seleccionado) que conforman el bus de datos; tales líneas presentan, además, la capacidad de desconexión, pasando a un estado de alta impedancia (tri-estado).

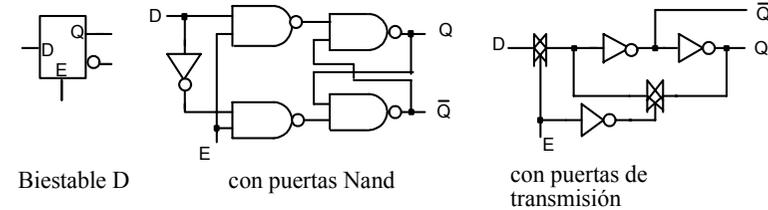
La memoria de un sistema digital puede incluir múltiples bloques **RAM**; la arquitectura de buses (datos y direcciones) permite organizar la memoria en forma sencilla, asignando a cada registro un número binario (con longitud igual al número de líneas del bus de direcciones). La correspondencia entre los registros y su numeración da lugar a un «mapa de memoria»; un bloque **RAM** ocupará un sector de dicho mapa y dos bloques diferentes deberán situarse en sectores diferentes.

El presente capítulo describe en detalle la configuración de los circuitos integrados de memoria de acceso directo y la forma de insertarlos en un mapa de memoria: cómo situar uno o varios bloques **RAM** en posiciones determinadas de un mapa de memoria e, incluso, cómo ubicar segmentos de un mismo bloque en sectores separados del mapa de memoria.

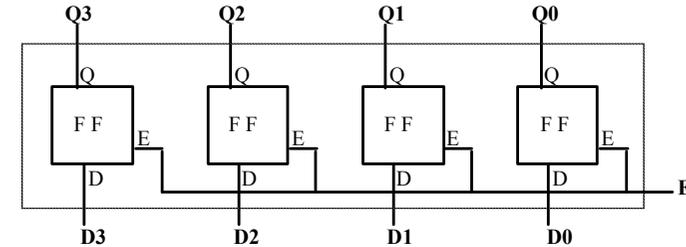
Por otra parte, los codificadores **ROM** (estudiados en el capítulo 4) pueden ser considerados como conjuntos de palabras binarias numeradas, seleccionables por su vector de entrada o «dirección»; en tal sentido, pueden ser utilizados como «memorias de sólo lectura». De forma que, además de los bloques **RAM** que solamente pueden suministrar datos después de que el sistema los haya escrito en ellos, pueden existir bloques **ROM** que corresponden a «conjuntos de registros de información fija» (programada previamente sobre ellos).

### 19.1. Biestables, registros y memorias

Una célula o unidad de memoria digital es un dispositivo capaz de almacenar y conservar un bit de información, es decir, un **0** o un **1** booleanos. La unidad de memoria básica es el biestable y, en concreto, el biestable tipo **D**: la entrada de habilitación determina el momento de aceptación de un nuevo bit, que será conservado hasta una nueva habilitación

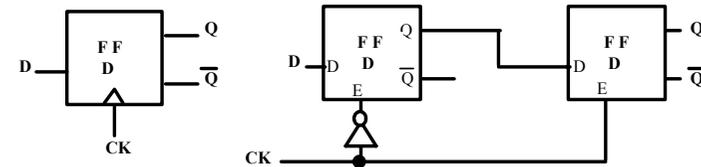


Un conjunto de  $n$  biestables **D** con habilitador común conforma un registro con capacidad de una palabra binaria de  $n$  bits: registro de retención (*latch memory*).



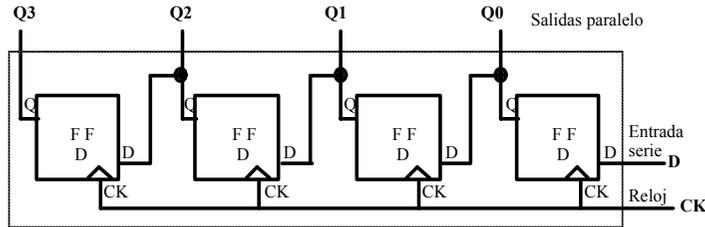
Este tipo de registros, habilitados por nivel (entrada de habilitación  $E = 1$ ), constituyen el elemento básico de almacenamiento masivo de información digital, ya que generalmente son de este tipo los registros internos de las memorias de acceso directo (bloques integrados **RAM**).

En los biestables síncronos la habilitación se realiza por flancos ( $CK = \uparrow$ ) en lugar de por niveles; cada biestable síncrono se construye por asociación de dos biestables habilitados por nivel, en configuración «amo-esclavo» (*master-slave*) de forma que la habilitación combinada del primer biestable con **0** y del segundo con **1** produce una habilitación global con flancos de subida (paso de la señal de reloj de **0** a **1**).

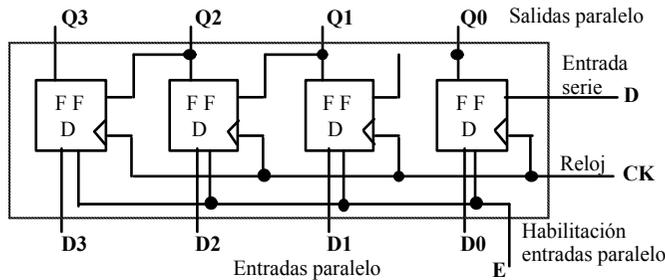


Con biestables síncronos, además de registros de retención síncronos, pueden conformarse otros dos tipos de registros de particular interés: los registros de desplazamiento y los contadores. [Los contadores, su configuración y sus aplicaciones, han sido descritos en detalle en los tres capítulos anteriores (16, 17 y 18).]

Los registros de desplazamiento (*shift register*, apartado 13.2) resultan de conectar un conjunto de biestables **D** síncronos en serie, de forma que la información que reciben avanza un biestable con cada pulso de reloj; de esta manera pueden recibir una palabra binaria, bit a bit, a través de su entrada y presentarla completa en las salidas de los biestables: conversión serie-paralelo.



Añadiendo a un registro de desplazamiento la posibilidad de carga paralelo (propia del registro de retención) se dispone, también, de la conversión paralelo-serie: una palabra recibida a través de las entradas paralelo puede ser transmitida, bit a bit, a través de la salida del último biestable.

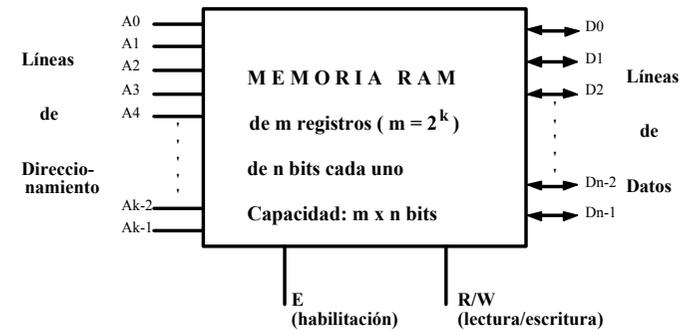


Además, los registros de desplazamiento resultan útiles para los algoritmos de multiplicación y de división, ya que el desplazamiento de un número un lugar hacia la izquierda equivale a multiplicar dicho número por 2. [Asimismo los registros de desplazamiento permiten la configuración de *pilas*, «memorias de acceso secuencial», que serán consideradas al final de este mismo apartado.]

En sistemas digitales reducidos bastan unos pocos registros junto con algunos contadores y algunos biestables de estado para configurar la memoria global del sistema. Ahora bien, en sistemas complejos suele requerirse un alto número de registros para memorizar el conjunto de datos y resultados (e instrucciones en el caso de sistemas que actúan bajo programa).

Resulta muy útil disponer, en un mismo bloque digital, de un amplio número de registros, capaces, cada uno de ellos, de memorizar una palabra binaria de **n** dígitos; los terminales de entrada y salida a estos registros han de ser comunes para todos ellos y unas entradas adicionales de direccionamiento indicarán en cada momento a cuál de los registros interesa acceder.

Esta agrupación de **m** registros de **n** bits, seleccionables por **k** entradas de direccionamiento ( $m = 2^k$ ), recibe el nombre de memoria de acceso directo (*random access memory*): **RAM**.



La denominación de memoria de acceso directo («acceso aleatorio») indica que, en cualquier momento, puede leerse o escribirse directamente sobre cualesquiera de sus registros. El calificativo de aleatorio se utilizó por contraposición a las memorias de acceso secuencial, en las cuales para acceder a un dato es preciso desplazar previamente todos los anteriores; un ejemplo característico de memorias de acceso secuencial son las de cinta magnética.

En un bloque RAM el vector presente en las entradas de direccionamiento selecciona el registro sobre el que se lee o se escribe y los terminales de datos actúan como entradas y como salidas para todos los registros, de forma que el tipo de acceso (lectura del registro o escritura del mismo) ha de ser controlado por una línea adicional **R/W** (*lectura/escritura*).

Una memoria de acceso directo tendrá **k** líneas de direccionamiento **A<sub>i</sub>**, que actúan como entradas, **n** líneas de datos **D<sub>i</sub>**, bidireccionales, una entrada de selección de la operación a realizar **R/W**, que distingue entre las dos operaciones siguientes:

- **R/W = 1** operación de lectura del registro seleccionado por **A<sub>i</sub>**
- **R/W = 0** operación de escritura sobre el registro seleccionado por **A<sub>i</sub>**

y una o varias entradas de habilitación **CE**, que permiten (**CE = 1**) o inhiben (**CE = 0**) el funcionamiento global de la memoria.

La transferencia de datos de un bloque de memoria de acceso directo **RAM** presenta dos nuevas posibilidades, no contempladas hasta ahora en los bloques digitales: *la bidireccionalidad* (los terminales de datos son bidireccionales, es decir, actúan en unos momentos como entradas y en otros como salidas) y *la desconexión* (los terminales de datos pueden situarse en estado de alta impedancia).

En las funciones y bloques digitales considerados anteriormente cada terminal era claramente unidireccional, entrada o salida, y los terminales se conectaban a través de líneas unidireccionales en las que los valores booleanos se comunicaban desde una salida a una o a varias entradas. Las líneas de datos de las memorias de acceso directo son bidireccionales, ya que tales bloques utilizan los mismos terminales cuando reciben la información a memorizar (escritura) y cuando la envían (lectura).

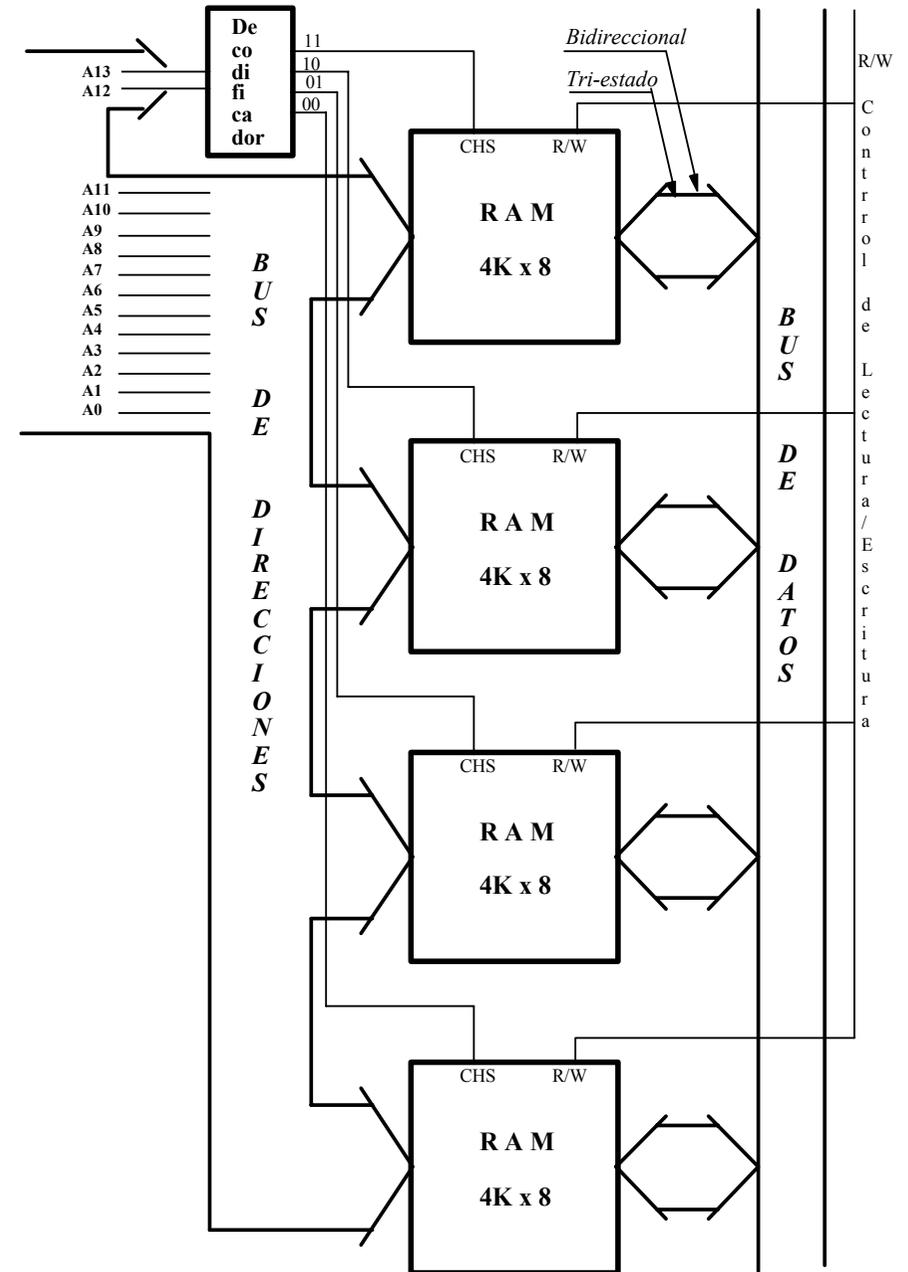
Además, se dota a los terminales de datos de una tercera posibilidad en la forma de un *tercer estado*, la *desconexión*: cuando **CE = 0** las líneas de datos de dicho bloque no actúan ni como entradas ni como salidas, sino como si el bloque no estuviese conectado a ellas. Esta capacidad de desconexión, estado de *alta impedancia*, permite la conexión de varios bloques **RAM** sobre las mismas líneas de datos, actuando en cada momento uno de ellos y estando inhibidos los demás.

La figura de la página siguiente representa la conexión de 4 bloques **RAM** de 4K registros de 8 bits, formando una memoria de 16K registros; se utilizan 12 líneas de direccionamiento para seleccionar el registro interno de entre los 4K que posee cada uno de los integrados y 2 líneas más de direccionamiento, convenientemente decodificadas en sus cuatro posibilidades, para habilitar uno de los cuatro bloques.

La *bidireccionalidad* y la desconexión o *triestado*, permiten introducir el concepto de *bus de datos* como conjunto de líneas que enlazan a diversos bloques digitales y por las que pueden transmitirse los datos en ambas direcciones. El bus de datos configura una forma sencilla de comunicación entre la parte que procesa la información (procesador) y la parte que la almacena (memoria).

Análogamente, las líneas de direccionamiento (que seleccionan un registro de entre los muchos que forman cada bloque) son compartidas por los diversos elementos de memoria y reciben el nombre de *bus de direcciones*; dicho bus es unidireccional (comunica hacia la memoria el número del registro).

*Bus de datos* y *bus de direcciones* facilitan en gran medida la organización estructural de aquellos sistemas que requieren amplia capacidad de memoria; se necesitan, además, unas pocas líneas de control que determinen el sentido de la operación a realizar (lectura/escritura) y el momento de su ejecución (sincronización).

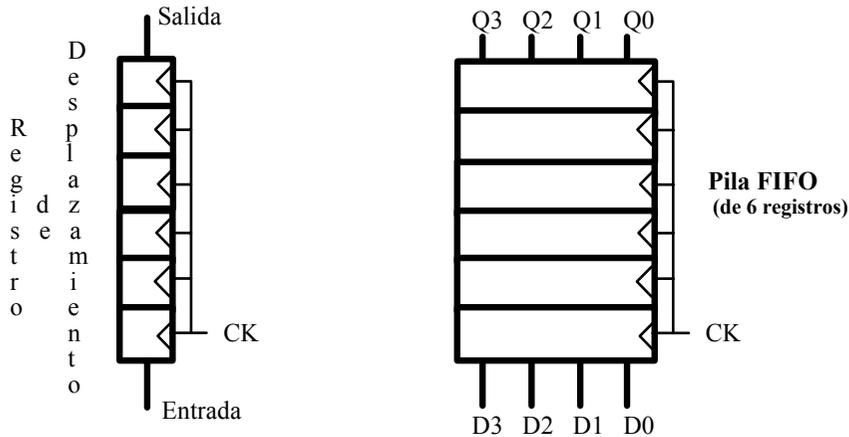


## Memorias de acceso secuencial

Consideraremos brevemente otra forma de agrupar conjuntos de registros consistente en «apilarlos» (cada uno «encima» del anterior) formando una columna de ellos; el acceso a los registros será secuencial: habrá que leer y escribir en ellos según el orden de la columna.

La agrupación de varios registros «en vertical», de manera que reciban la información por las entradas del primero de ellos y la devuelvan por las salidas del último registro, da lugar a una *pila*; el conjunto equivale a una «memoria de desplazamiento», capaz de almacenar secuencialmente varias palabras binarias y devolverlas en el mismo orden en que las ha recibido: **pila FIFO** (*first in, first out*), la primera palabra en entrar será también la primera en salir.

El desplazamiento se produce a través de los sucesivos registros; en tal sentido, una pila FIFO de  $n$  registros de  $m$  bits puede construirse con  $m$  registros de desplazamiento de  $n$  bits cada uno de ellos, orientados «verticalmente» y colocados unos al lado de otros:

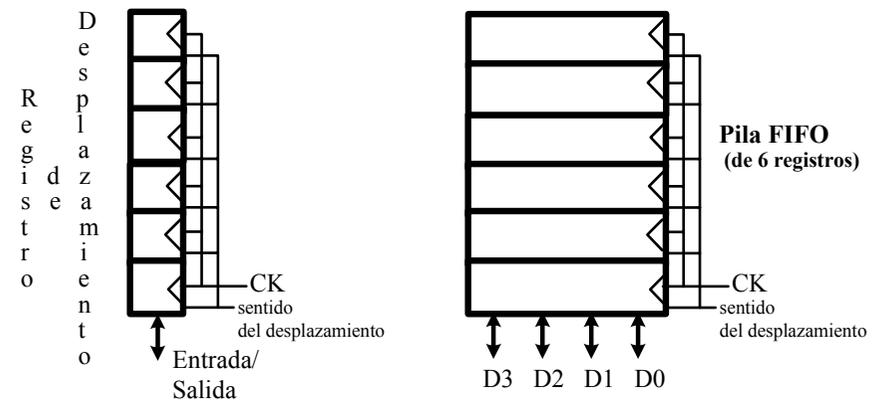


Una memoria RAM puede ser utilizada como pila FIFO si se le añaden dos contadores que realicen su direccionamiento en la siguiente forma:

- los contadores deberán ser de módulo igual a la capacidad de la memoria (para recorrer exactamente las direcciones de la misma);
- uno de ellos contendrá la dirección de escritura y deberá incrementarse (pasar a la dirección siguiente) cuando se escribe en la pila;
- el otro contador tendrá la dirección de lectura y se incrementará también cuando se produzca una operación de lectura en la misma;
- cuando ambos contadores señalen la misma dirección la pila se encontrará vacía y, en cambio, cuando el contador de escritura alcance la dirección anterior a la de lectura la pila estará llena (y no deben escribirse nuevos datos en ella).

Las pilas FIFO se utilizan para almacenar temporalmente palabras binarias (es decir, informaciones sucesivas) que serán utilizadas posteriormente en el mismo orden con que se han recibido o producido; una aplicación típica es la comunicación entre sistemas de diferente velocidad (por ejemplo entre un computador y una impresora), de forma que el emisor deposita a su propia velocidad un fichero sobre la pila y el receptor lo recoge a una velocidad distinta.

Otro tipo diferente es la **pila LIFO** (*last in, first out*) en la que la última palabra en entrar es la primera en salir; tendrá los mismos terminales para almacenar una palabra y para extraerla de la pila, de forma que las palabras se leen en orden inverso al que se han escrito en ella. Puede construirse con registros de desplazamiento bidireccionales, conformando un bus, también bidireccional, con las entradas y salidas de los primeros bistables de dichos registros.



Los procesadores de los sistemas que actúan bajo programa (computadores, microprocesadores, etc. ...) utilizan las pilas LIFO para el servicio de subrutinas, interrupciones y otros tipos de saltos que se realizan con «intención de volver»; la información relativa a la vuelta al punto desde el que se efectúa el salto ha de ser almacenada de forma que, si se producen varios saltos sucesivos, la información correspondiente al último de ellos será la primera en ser recuperada.

También es posible utilizar una memoria RAM como pila FIFO; bastará añadirle un contador de direccionamiento (de módulo igual a la capacidad de la memoria), que indique la posición de memoria en la que se debe escribir (la primera posición que se encuentra vacía). La escritura debe efectuarse sobre dicha posición de memoria y el contador debe incrementarse (señalando la siguiente), mientras que la lectura debe hacerse decrementando previamente el contador y leyendo de la posición de memoria resultante. La pila LIFO se encontrará vacía cuando el contador indique la primera posición de la memoria RAM y estará llena cuando el contador contenga la última posición de la misma.

19.2. Configuración de los bloques de memoria de acceso directo

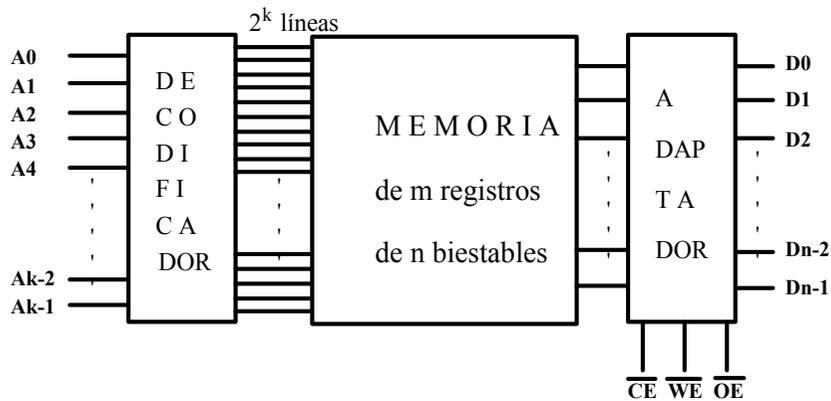
La mayoría de las memorias de acceso directo integradas presentan tres terminales de control:

- $\overline{CE}$  que habilita el circuito integrado ( $\overline{CE} = 0$ ), de forma que si  $\overline{CE} = 1$  es como si dicho circuito no se encontrara presente, es decir, todas sus líneas de datos se encuentran en alta impedancia y las diversas partes que conforman el integrado adoptan un estado de consumo mínimo (*standby*)
- $\overline{WE}$  que habilita la operación de escritura ( $\overline{WE} = 0$ ), posicionando las líneas de datos como entradas y activando el correspondiente circuito de escritura
- $\overline{OE}$  que habilita las líneas de datos como salidas ( $\overline{OE} = 0$ ), permitiendo la ejecución de una operación de lectura.

De acuerdo con ello, el circuito integrado RAM se puede encontrar en una de las siguientes cuatro situaciones:

- $\overline{CE} = 1$  desconexión y consumo mínimo
- $\overline{CE} = 0$  y  $\overline{WE} = 0$  escritura; líneas de datos como entradas
- $\overline{CE} = 0$ ,  $\overline{WE} = 1$  y  $\overline{OE} = 0$  lectura; líneas de datos como salidas
- $\overline{CE} = 0$ ,  $\overline{WE} = 1$  y  $\overline{OE} = 1$  conectado; las líneas de datos en alta impedancia.

La estructura interna de una memoria de acceso directo es, en términos conceptuales, la representada en la siguiente figura:



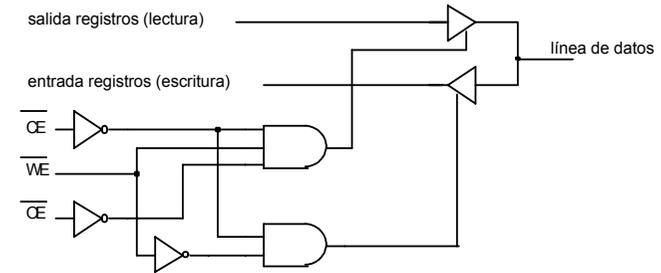
El módulo central de la memoria contiene  $m = 2^k$  registros, constituido cada uno de ellos por  $n$  biestables. El decodificador es un bloque combinacional típico que selecciona numéricamente una de entre  $m$  líneas, cada una de las cuales va a corresponder a uno de los registros.

Un circuito adaptador determina la dirección del flujo de información del registro direccionado:

- lectura del contenido del mismo, operando las líneas de datos como salidas
- escritura de un nuevo dato sobre sus biestables, actuando las líneas de datos como entradas.

Obviamente, la lectura de un registro de un bloque RAM es una operación no destructiva: el registro no se borra ni se modifica, sino que sigue conservando la palabra binaria que se ha leído; solamente la escritura de una nueva palabra en tal registro modifica su contenido.

El circuito adaptador realiza el control de las operaciones de la memoria (lectura/escritura/desconexión), a través de adaptadores triestado cuya entrada de habilitación determina su situación de conexión o desconexión (estado de alta impedancia); la bidireccionalidad se consigue mediante la utilización de dos adaptadores triestado, que permiten actuar como salidas de los registros (lectura) o como entradas de los mismos (escritura).



Debido al gran número de registros presentes, resulta necesario minimizar el área de integración y la forma de selección de los mismos, para lo cual se adoptan configuraciones muy simples.

Basta con un solo decodificador para seleccionar el registro sobre el que se ejecuta la correspondiente operación (lectura/escritura); las  $k$  líneas de direccionamiento serán decodificadas en sus  $2^k$  posibilidades, cada una de las cuales sirve para habilitar uno de los registros de la memoria. Ahora bien, el tamaño del decodificador y el número de líneas de selección de registros puede reducirse mucho dividiendo la decodificación en dos partes (fila y columna), de forma que cada registro corresponda a dos líneas de selección (su fila y su columna).

De manera que un bloque RAM de 1 Mega, en lugar de decodificar de una vez sus 20 líneas de direccionamiento sobre sus 1.048.576 registros, divide dichas líneas en dos grupos de 10, con dos decodificadores mucho más pequeños, cuyas líneas de salida se reducen a  $2 \times 1.024$  (un número quinientas veces inferior al anterior). El decodificador de 20 líneas de entrada hubiera requerido  $10^6$  puertas con, al menos,  $20 \times 10^6$  transistores, mientras que dos decodificadores de 10 líneas de entrada requieren del orden de  $2 \times 10^3$  puertas con unos  $20 \times 10^3$  transistores).

Por ello, para reducir el número de líneas necesarias para la selección de registro y el tamaño de los decodificadores, la distribución de los biestables de los registros adopta una estructura de tipo «bidimensional», conformando una matriz en la que cada registro queda identificado por la fila y la columna que ocupa ; las líneas de direccionamiento se agrupan en dos subconjuntos, uno de los cuales indica la fila y el otro la columna a la que pertenece el registro seleccionado. [Véase la correspondiente figura en la página siguiente.]

Los biestables de cada registro se localizan en «hojas» sucesivas; cada una de ellas contiene un biestable de cada uno de los registros y se encuentra recorrida horizontalmente por las líneas de selección de fila y verticalmente por líneas que enlazan cada una de las columnas; cada hoja se corresponde con una de las líneas del bus de datos y cuenta con un amplificador de lectura/escritura.

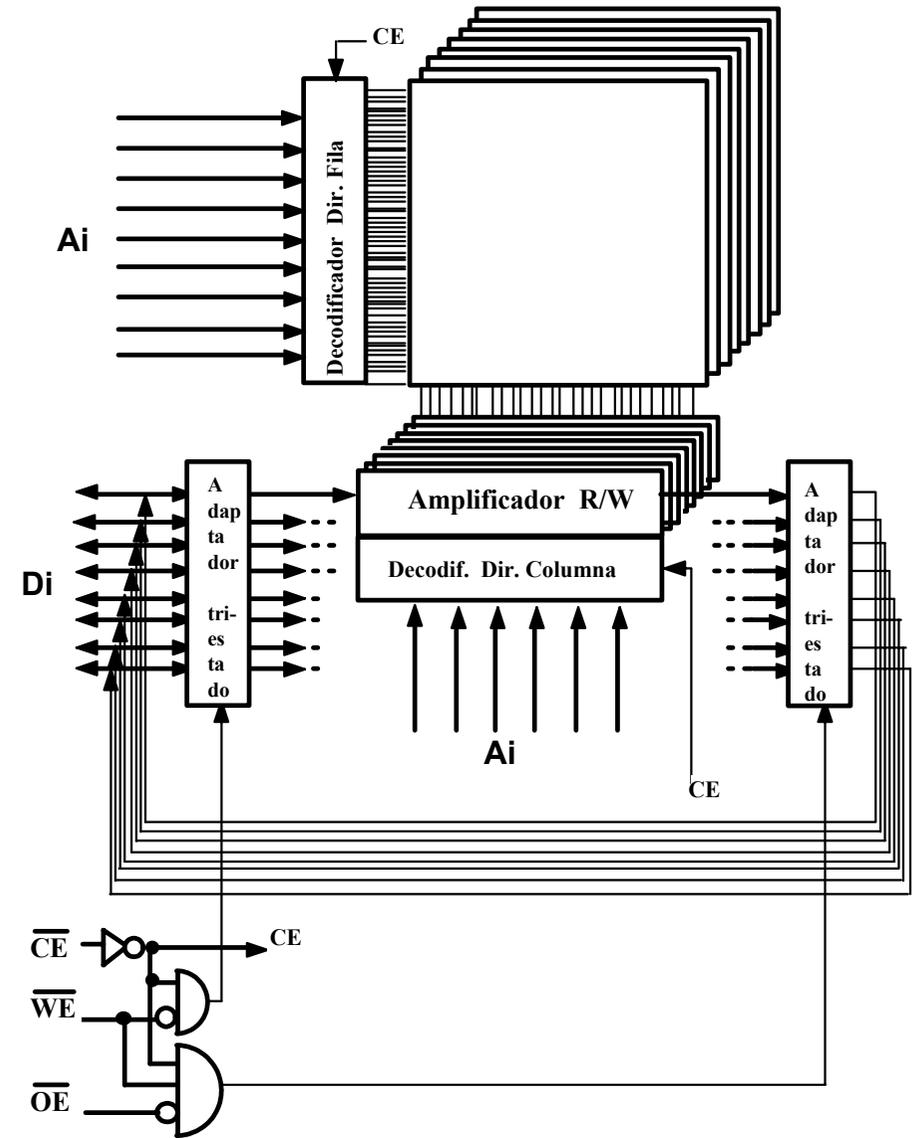
Cada columna cuenta con dos líneas una para  $Q$  y otra para  $\bar{Q}$ , conectadas ambas a los correspondientes terminales de los biestables; al seleccionar una de las filas, mediante la activación de la correspondiente línea de selección de fila, los biestables que se encuentran en ella quedan unidos a las «líneas de columna» que les corresponden. [Véanse en relación con este párrafo y siguientes las figuras de la página 189.]

Las «líneas de columna» establecen la comunicación de los biestables con el amplificador de lectura/escritura; un amplificador para cada una de las hojas, es decir, para cada uno de los bits de los registros. Las líneas de selección de columna determinan que, en cada momento, solamente una de las columnas se encuentre unida a dicho amplificador de lectura/escritura.

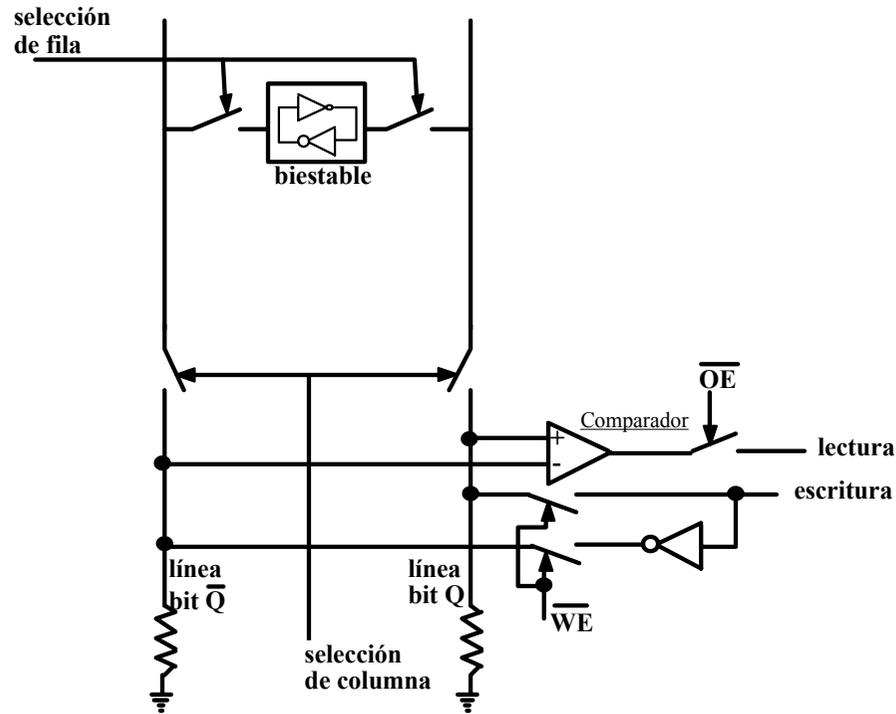
Las operaciones de lectura y de escritura son realizadas por el citado amplificador de lectura/escritura actuando sobre las «líneas de columna», esto es, comparando las tensiones existentes en ambas (lectura) o imponiéndoles el correspondiente valor booleano (escritura):

- una operación de escritura se ejecuta estableciendo en la línea  $Q$  de la columna seleccionada el valor booleano a almacenar en el correspondiente biestable y en la línea  $\bar{Q}$  el valor booleano inverso
- una operación de lectura se realiza por comparación entre las tensiones de ambas «líneas de columna» para discriminar cuál de ellas se encuentra a **1** (a mayor tensión).

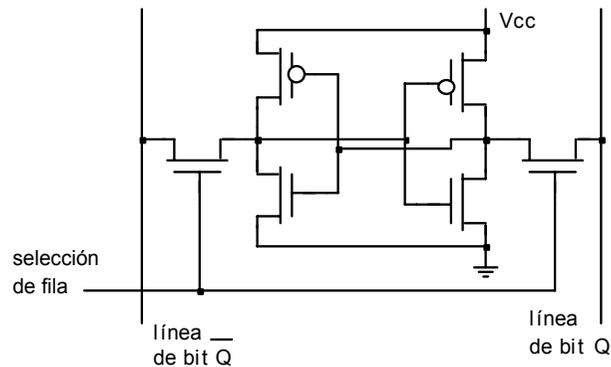
La configuración electrónica de cada biestable ha de ser también muy simple para reducir su área de integración: basta un par de inversores en lazo cerrado, la salida de cada uno de ellos conectada a la entrada del otro.



Estructura de una memoria de acceso directo (diagrama de bloques)



Conexión y forma de selección de los biestables en una memoria de acceso directo



Configuración y conexiones de uno de los biestables en tecnología CMOS

### 19.3. Arquitectura de buses: mapa de memoria

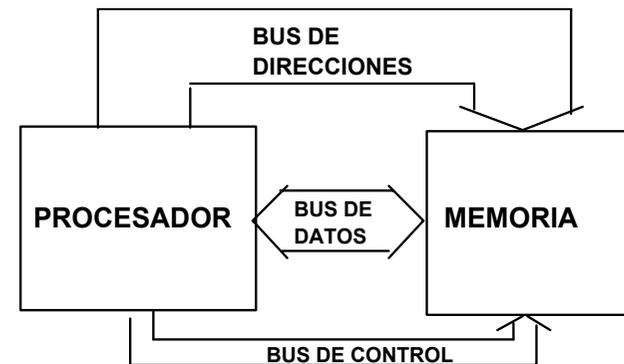
Un bus es un conjunto de líneas de comunicación entre varios subsistemas o bloques digitales, en concreto, entre varios circuitos integrados, entre varias placas de circuitos en los sistemas amplios o, incluso, entre sistemas digitales diferenciados.

La utilización conjunta del bus de datos y del bus de direcciones ha permitido organizar la memoria en forma muy simple: todos los registros se numeran correlativamente accediendo a ellos (lectura o escritura) por las mismas líneas de datos mientras que las líneas de direccionamiento seleccionan el registro sobre el que se opera. Las líneas de datos son bidireccionales (lectura/escritura), mientras que las de direccionamiento son unidireccionales (se dirigen siempre hacia la memoria).

A cada registro se le hace corresponder circuitalmente un número binario diferenciado; a cada número o dirección le corresponde un sólo registro o ninguno (posición de memoria vacía): dicho número de selección (dirección) de registro es recibido por la memoria, a través del bus de direcciones  $A_i$ . La correspondencia entre cada registro o conjunto de registros y la dirección o direcciones que ocupan configura el mapa de memoria del sistema.

Los buses determinan una división estructural del sistema digital en dos partes: el procesador que efectúa el procesamiento de la información (unidad operativa y de control) y la memoria que almacena la información. Procesador y memoria se comunican a través de tres buses:

- el bus de datos, por el que viaja la información en forma de palabras digitales
- el bus de direcciones, que selecciona el registro sobre el que se opera
- el bus de control, que determina la dirección de transferencia de la información y sincroniza dicha transferencia.



El bus de datos es bidireccional; puede recibir la información del procesador y comunicarla hacia la memoria o, al revés, actuar la memoria como salida de la información y el procesador como entrada. El número de líneas de datos determina la longitud de palabra y de los registros de la memoria.





Por ello, es preciso ocupar los siguientes sectores de memoria:

A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	
0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	<b>1 0 0 0 H</b>
0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	<b>1 F F F H</b>
<b>0</b>	<b>0</b>	<b>0</b>	<b>1</b>	-	-	-	-	-	-	-	-	-	-	-	-	<b>4K</b>
0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	<b>2 0 0 0 H</b>
0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	<b>3 F F F H</b>
<b>0</b>	<b>0</b>	<b>1</b>	-	-	-	-	-	-	-	-	-	-	-	-	-	<b>8K</b>
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	<b>4 0 0 0 H</b>
0	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	<b>4 F F F H</b>
<b>0</b>	<b>1</b>	<b>0</b>	<b>0</b>	-	-	-	-	-	-	-	-	-	-	-	-	<b>4K.</b>

En el primero de los sectores anteriores caben 4K (dos de los bloques), el segundo sector es de 8K (cuatro bloques de 2K) y, por último, quedan 4K (dos bloques) que irán en el tercero de los sectores.

La situación de los 8 bloques sucesivos será la siguiente:

<b>0 0 0 1</b>	<b>0 - - -</b>	<b>- - - -</b>	<b>- - - -</b>	<b>1000 - 17FF</b>
<b>0 0 0 1</b>	<b>1 - - -</b>	<b>- - - -</b>	<b>- - - -</b>	<b>1800 - 1FFF</b>
<b>0 0 1 0</b>	<b>0 - - -</b>	<b>- - - -</b>	<b>- - - -</b>	<b>2000 - 27FF</b>
<b>0 0 1 0</b>	<b>1 - - -</b>	<b>- - - -</b>	<b>- - - -</b>	<b>2800 - 2FFF</b>
<b>0 0 1 1</b>	<b>0 - - -</b>	<b>- - - -</b>	<b>- - - -</b>	<b>3000 - 37FF</b>
<b>0 0 1 1</b>	<b>1 - - -</b>	<b>- - - -</b>	<b>- - - -</b>	<b>3800 - 3FFF</b>
<b>0 1 0 0</b>	<b>0 - - -</b>	<b>- - - -</b>	<b>- - - -</b>	<b>4000 - 47FF</b>
<b>0 1 0 0</b>	<b>1 - - -</b>	<b>- - - -</b>	<b>- - - -</b>	<b>4800 - 4FFF.</b>

Como los 8 bloques van seguidos, las líneas **A13 A12 A11** sirven para diferenciarlos (si bien el primer bloque en el mapa de memoria corresponde al valor **010** de dichas líneas, el segundo al valor **011**,... y el último al valor **001**) y puede utilizarse un decodificador análogo al del ejercicio anterior; pero, en este caso, el decodificador debe habilitarse en las siguientes situaciones:

$$A_{15} = 0, A_{14} = 0, A_{13} = 0, A_{12} = 1$$

$$A_{15} = 0, A_{14} = 0, A_{13} = 1$$

$$A_{15} = 0, A_{14} = 1, A_{13} = 0, A_{12} = 0$$

$$E(\text{decodif.}) = \overline{A_{15}} \cdot (\overline{A_{14}} \cdot \overline{A_{13}} \cdot A_{12} + \overline{A_{14}} \cdot A_{13} + A_{14} \cdot \overline{A_{13}} \cdot \overline{A_{12}}) \cdot \text{DIRV}$$

$$a_{10} - a_0 = A_{10} - A_0; \quad WE = \overline{R/W} \cdot \text{DATV}; \quad OE = R/W.$$

**19.3.4. Posicionamiento de 4 bloques RAM en sectores no contiguos:** sean 4 circuitos integrados RAM de 2K registros que deben colocarse en el mapa de memoria

a) a partir de las posiciones **0000H, 2000H, 4000H, y 6000H**, respectivamente.

La situación de los 4 bloques sucesivos será la siguiente:

A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	
<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	-	-	-	-	-	-	-	-	-	-	-	<b>0000 - 07FF</b>
<b>0</b>	<b>0</b>	<b>1</b>	<b>0</b>	<b>0</b>	-	-	-	-	-	-	-	-	-	-	-	<b>2000 - 27FF</b>
<b>0</b>	<b>1</b>	<b>0</b>	<b>0</b>	<b>0</b>	-	-	-	-	-	-	-	-	-	-	-	<b>4000 - 47FF</b>
<b>0</b>	<b>1</b>	<b>1</b>	<b>0</b>	<b>0</b>	-	-	-	-	-	-	-	-	-	-	-	<b>6000 - 67FF</b>

Se observa que los 4 bloques pueden diferenciarse por los valores de las líneas **A14 A13** de forma que pueden seleccionarse mediante un decodificador de 2 líneas de entrada (**A14 A13**) y 4 de salida (para las habilitaciones **CE** de cada uno de los circuitos integrados); las líneas **A14** y **A13** adoptan los cuatro vectores posibles (**00, 01, 10** y **11**) y las otras tres líneas **A15 A12** y **A11** tienen el mismo valor en los cuatro bloques, de forma que el decodificador debe habilitarse cuando:  $A_{15} = 0, A_{12} = 0, A_{11} = 0$ .

$$E(\text{decodificador}) = \overline{A_{15}} \cdot \overline{A_{12}} \cdot \overline{A_{11}} \cdot \text{DIRV}$$

$$a_{10} - a_0 = A_{10} - A_0; \quad WE = \overline{R/W} \cdot \text{DATV}; \quad OE = R/W.$$

b) a partir de las posiciones **1000H, 5000H, 9800H, y B000H**.

La situación de los 4 bloques sucesivos será la siguiente:

<b>0 0 0 1</b>	<b>0 - - -</b>	<b>- - - -</b>	<b>- - - -</b>	<b>1000 - 17FF</b>
<b>0 1 0 1</b>	<b>0 - - -</b>	<b>- - - -</b>	<b>- - - -</b>	<b>5000 - 57FF</b>
<b>1 0 0 1</b>	<b>1 - - -</b>	<b>- - - -</b>	<b>- - - -</b>	<b>9800 - 9FFF</b>
<b>1 0 1 1</b>	<b>0 - - -</b>	<b>- - - -</b>	<b>- - - -</b>	<b>B000 - B7FF</b>

Como no existe ninguna regularidad en las posiciones que ocupan, los 4 bloques deberán habilitarse independientemente, a través de la función correspondiente a su situación en el mapa de memoria:

$$CE 1 = \overline{A_{15}} \cdot \overline{A_{14}} \cdot \overline{A_{13}} \cdot A_{12} \cdot \overline{A_{11}} \cdot \text{DIRV}$$

$$CE 2 = \overline{A_{15}} \cdot A_{14} \cdot \overline{A_{13}} \cdot A_{12} \cdot \overline{A_{11}} \cdot \text{DIRV}$$

$$CE 3 = A_{15} \cdot \overline{A_{14}} \cdot \overline{A_{13}} \cdot A_{12} \cdot A_{11} \cdot \text{DIRV}$$

$$CE 4 = A_{15} \cdot \overline{A_{14}} \cdot A_{13} \cdot A_{12} \cdot \overline{A_{11}} \cdot \text{DIRV}$$

Estas funciones pueden ser programadas sobre un bloque **PAL** (precisamente el direccionamiento para configurar mapas de memoria fue una de las primeras aplicaciones de dichos bloques programables).

**19.4. Direccionamiento de un bloque RAM en sectores separados**

Aunque los ejemplos siguientes constituyen casos muy particulares de ubicación de un circuito integrado RAM en un mapa de memoria (que no se presentan habitualmente), son de gran interés de cara a comprender en profundidad el posicionamiento de registros en un mapa de memoria.

**19.4.1.** Sea un circuito integrado RAM de 8K (13 líneas de direccionamiento), cuyos registros se desean colocar en el mapa de memoria, en sectores no contiguos:

a) 4K registros en el sector inicial del mapa de memoria y 4K en el final.

Un segmento de 4K registros requiere 12 líneas de direccionamiento **a11 – a0**, de forma que el situado en el sector inicial tendrá las restantes líneas **A15 – A12** a **0**, mientras que su valor para el sector final será **1**; el circuito integrado de 8K tendrá 13 líneas de direccionamiento **a12 – a0**, 12 de las cuales **a11 – a0** sirven para conformar segmentos de 4K y la línea **a12** distinguirá entre los dos segmentos (inicial y final del mapa de memoria):

<b>0 0 0 0</b>	- - - -	- - - -	- - - -	<b>0000 – 0FFF</b>
<b>1 1 1 1</b>	- - - -	- - - -	- - - -	<b>F000 – FFFF</b>

$$CE = (\overline{A_{15}} \cdot \overline{A_{14}} \cdot \overline{A_{13}} \cdot \overline{A_{12}} + A_{15} \cdot A_{14} \cdot A_{13} \cdot A_{12}) \cdot DIRV$$

$$a_{12} = A_{12} \text{ o, también, } a_{12} = A_{15}$$

$$a_{11} - a_0 = A_{11} - A_0; \quad WE = \overline{R/W} \cdot DATV; \quad OE = R/W$$

b) 2K registros en los sectores del mapa de memoria cuya dirección inicial es, respectivamente, **0000H, 4000H, 8000H y C000H**.

Un segmento de 2K registros requiere 11 líneas de direccionamiento **a10 – a0**; las posiciones de memoria ocupadas por los cuatro segmentos de 2K serán:

<b>0 0 0 0</b>	<b>0</b> - - -	- - - -	- - - -	<b>0000 – 07FF</b>
<b>0 1 0 0</b>	<b>0</b> - - -	- - - -	- - - -	<b>4000 – 47FF</b>
<b>1 0 0 0</b>	<b>0</b> - - -	- - - -	- - - -	<b>8000 – 87FF</b>
<b>1 1 0 0</b>	<b>0</b> - - -	- - - -	- - - -	<b>C000 – C7FF</b>

Para situar el bloque en estas posiciones, debe habilitarse cuando

$$A_{13} = 0, A_{12} = 0, A_{11} = 0: \quad CE = \overline{A_{13}} \cdot \overline{A_{12}} \cdot \overline{A_{11}} \cdot DIRV$$

$$a_{10} - a_0 = A_{10} - A_0; \quad WE = \overline{R/W} \cdot DATV; \quad OE = R/W$$

Para diferenciar los 4 segmentos del bloque RAM que van a encontrarse en 4 sectores separados del mapa de memoria, disponemos de las líneas **A15 A14** que diferencian dichos 4 sectores (**A15 A14** recorren todos vectores posibles); las entradas **a12 a11** que distinguen los 4 segmentos del bloque han de conectarse a dichas líneas **A15 A14**:  $a_{12} = A_{15}; \quad a_{11} = A_{14}$

Téngase en cuenta que, si se conectasen las líneas **a12** y **a11** a las del mismo número del bus de direcciones, solamente se utilizaría el primer segmento de 2K del bloque **RAM**, pues los valores de las entradas **a12a11 = A12A11 = 00** serían iguales (**a12a11 = 00**) en los cuatro sectores del mapa de memoria (lo cual significa que los otros segmentos **a12a11 = 01, 10, 11** no serían utilizados, sino que dicho segmento **a12a11 = 00** se encontraría repetido cuatro veces en sectores distintos del mapa de memoria).

c) 6K registros en el sector inicial del mapa de memoria y 2K en el final.

Distribuyendo el circuito integrado en segmentos de 2K (11 líneas **a10 – a0**):

<b>0 0 0 0</b>	<b>0</b> - - -	- - - -	- - - -	<b>0000 – 07FF</b>
<b>0 0 0 0</b>	<b>1</b> - - -	- - - -	- - - -	<b>0800 – 0FFF</b>
<b>0 0 0 1</b>	<b>0</b> - - -	- - - -	- - - -	<b>1000 – 17FF</b>
<b>1 1 1 1</b>	<b>1</b> - - -	- - - -	- - - -	<b>F800 – FFFF</b>

Los tres primeros segmentos han de situarse en el sector inicial de la memoria ( $A_{15} = 0, A_{14} = 0, A_{13} = 0$ , y  $A_{12}A_{11} = 00, 01, 10$ ) y el cuarto segmento en el sector final ( $A_{15} = 1, A_{14} = 1, A_{13} = 1, A_{12} = 1, A_{11} = 1$ ):

$$CE = (\overline{A_{15}} \cdot \overline{A_{14}} \cdot \overline{A_{13}} \cdot (\overline{A_{12}} + \overline{A_{11}}) + A_{15} \cdot A_{14} \cdot A_{13} \cdot A_{12} \cdot A_{11}) \cdot DIRV$$

Los cuatro segmentos pueden diferenciarse por los valores de las líneas **A12 A11** que recorren todos vectores posibles:  $a_{12} = A_{12}; \quad a_{11} = A_{11}$

$$a_{10} - a_0 = A_{10} - A_0; \quad WE = \overline{R/W} \cdot DATV; \quad OE = R/W$$

d) 2K registros a partir de la posición **4000H** y 6K a partir de **A000H**

En segmentos de 2 K:

<b>0 1 0 0</b>	<b>0</b> - - -	- - - -	- - - -	<b>4000 – 47FF</b>
<b>1 0 1 0</b>	<b>0</b> - - -	- - - -	- - - -	<b>A000 – A7FF</b>
<b>1 0 1 0</b>	<b>1</b> - - -	- - - -	- - - -	<b>A800 – AFFF</b>
<b>1 0 1 1</b>	<b>0</b> - - -	- - - -	- - - -	<b>B000 – B7FF</b>

el primer segmento ha de situarse en  $A_{15} = 0, A_{14} = 1, A_{13} = 0, A_{12} = 0$  y  $A_{11} = 0$  y los otros tres segmentos en  $A_{15} = 1, A_{14} = 0, A_{13} = 1$  y  $A_{12}A_{11} = 00, 01, 10$ :

$$CE = (\overline{A_{15}} \cdot A_{14} \cdot \overline{A_{13}} \cdot \overline{A_{12}} \cdot \overline{A_{11}} + A_{15} \cdot \overline{A_{14}} \cdot A_{13} \cdot (\overline{A_{12}} + \overline{A_{11}})) \cdot DIRV$$

En este caso, para diferenciar los cuatro segmentos y determinar la conexión de las entradas de direccionamiento que los numeran **a12 a11** podemos construir una pequeña tabla en la forma siguiente:

A15	A14	A13	A12	A11	a12	a11
0	1	0	0	0	0	0
1	0	1	0	0	0	1
1	0	1	0	1	1	0
1	0	1	1	0	1	1

A partir de esta «tabla de verdad» se obtienen:  
 $a_{12} = A_{12} + \overline{A_{11}}$   
 $a_{11} = A_{13} \cdot \overline{A_{11}}$

**19.4.2.** Sea un circuito integrado RAM de 4K (12 líneas de direccionamiento), cuyos registros se desea situar en las posiciones iniciales del mapa de memoria, pero sabiendo que el primer sector de 1K del mapa se encuentra ocupado previamente.

El sector de 1K ocupado corresponde a:

**0 0 0 0 0 0 - - - - - 0000 - 03FF**

A continuación de dicho sector podemos situar los 4K registros en la siguiente forma:

**0 0 0 0 0 1 - - - - - 0400 - 07FF**  
**0 0 0 0 1 - - - - - 0800 - 0FFF**  
**0 0 0 1 0 0 - - - - - 1000 - 17FF**

El primero de los sectores anteriores es de 1K (10 líneas), el segundo permite colocar 2K (11 líneas) y el tercero corresponde a 1K restante; la habilitación del circuito integrado ha de producirse en las tres situaciones:

- $A_{15} = 0, A_{14} = 0, A_{13} = 0, A_{12} = 0, A_{11} = 0, A_{10} = 1$  1 sector de 1K
- $A_{15} = 0, A_{14} = 0, A_{13} = 0, A_{12} = 0, A_{11} = 1$  2 sectores de 1K
- $A_{15} = 0, A_{14} = 0, A_{13} = 0, A_{12} = 1, A_{11} = 0, A_{10} = 0$  1 sector de 1K

$$CE = \overline{A_{15}} \cdot \overline{A_{14}} \cdot \overline{A_{13}} \cdot (\overline{A_{12}} \cdot \overline{A_{11}} \cdot A_{10} + \overline{A_{12}} \cdot A_{11} + A_{12} \cdot \overline{A_{11}} \cdot A_{10}) \cdot \overline{DIRV}$$

Como los 4 sectores de memoria de 1K van seguidos, las líneas **A11 A10** sirven para diferenciarlos (si bien el primer segmento del bloque integrado en el mapa de memoria corresponde al valor **01** de dichas líneas, el segundo al valor **10**, el tercero a **11** y el último al valor **00**):

$$a_{11} - a_0 = A_{11} - A_0; \quad WE = \overline{R/W} \cdot \overline{DATV}; \quad OE = R/W$$

**19.4.3.** Sea un circuito integrado RAM de 8K (13 líneas de direccionamiento), cuyos registros se desean situar en la parte inicial de un mapa de memoria, en el cual se encuentra ocupado el sector **0800H a 6FFFH**.

**0 8 0 0 H = 0000 1000 0000 0000**  
**6 F F F H = 0110 1111 1111 1111**

Previamente a la posición **0800H** se pueden poner registros en:

**0 0 0 0 0 - - - - - 0000 - 07FF**  
 sector de 2K; quedarán por situar otros 6K registros detrás de la posición 6FFFH:  
**0 1 1 1 - - - - - 7000 - 7FFF**  
**1 0 0 0 0 - - - - - 8000 - 87FF**

sectores de 4K y 2K, respectivamente.

La habilitación del circuito integrado ha de producirse en:

- $A_{15} = 0, A_{14} = 0, A_{13} = 0, A_{12} = 0, A_{11} = 0$  1 segmento de 2K
- $A_{15} = 0, A_{14} = 1, A_{13} = 1, A_{12} = 1$  2 segmentos de 2K
- $A_{15} = 1, A_{14} = 0, A_{13} = 0, A_{12} = 0, A_{11} = 0$  1 segmento de 2K

(de los tres vectores anteriores, primero y tercero pueden simplificarse entre sí)

$$CE = (\overline{A_{14}} \cdot \overline{A_{13}} \cdot \overline{A_{12}} \cdot \overline{A_{11}} + \overline{A_{15}} \cdot A_{14} \cdot A_{13} \cdot A_{12}) \cdot \overline{DIRV}$$

El bloque RAM queda organizado en cuatro segmentos de 2K registros, de los cuales los tres últimos se sitúan juntos; para diferenciar los segmentos y determinar la conexión de sus entradas de selección **a12 a11** podemos utilizar la siguiente tabla:

A15	A14	A13	A12	A11	a12	a11
0	0	0	0	0	0	0
0	1	1	1	0	0	1
0	1	1	1	1	1	0
1	0	0	0	0	1	1

A partir de esta tabla se obtienen:  
 $a_{12} = A_{11} + A_{15}$   
 $a_{11} = A_{12} \cdot \overline{A_{11}} + A_{15}$

$$a_{10} - a_0 = A_{10} - A_0; \quad WE = \overline{R/W} \cdot \overline{DATV}; \quad OE = R/W$$

**19.4.4. Determinación de la posición en un mapa de memoria:** ¿cuál será el mapa de memoria ocupado por un bloque RAM de 8K, cuya habilitación es la siguiente:

$$CE = \overline{A_{15}} \cdot (\overline{A_{14}} \cdot A_{13} \cdot (\overline{A_{12}} + \overline{A_{11}})) + A_{14} \cdot \overline{A_{13}} \cdot \overline{A_{12}} \cdot \overline{A_{11}} \cdot \overline{DIRV}$$

El bloque se habilitará en cada una de las siguientes situaciones:

- $A_{15} = 0, A_{14} = 0, A_{13} = 1, A_{12} = 0$
- $A_{15} = 0, A_{14} = 0, A_{13} = 1, A_{11} = 0$
- $A_{15} = 0, A_{14} = 1, A_{13} = 0, A_{12} = 0, A_{11} = 0$

que podemos expresarlas ordenadamente en la siguiente tabla

A <sub>15</sub>	A <sub>14</sub>	A <sub>13</sub>	A <sub>12</sub>	A <sub>11</sub>	
0	0	1	0	0	sector <b>2000-27FF</b>
0	0	1	0	1	sector <b>2800-2FFF</b>
0	0	1	1	0	sector <b>3000-37FF</b>
0	1	0	0	0	sector <b>4000-47FF</b>

Los sectores anteriores son de 2K (11 líneas **A<sub>10</sub> - A<sub>0</sub>** para direccionar sus registros) y los tres primeros son contiguos (6K), de forma que el mapa de memoria ocupado está dividido en dos trozos: **2000-37FF** y **4000-47FF** (entre ambos queda un sector de 2K 3800-3FFF).

¿Cómo deben conectarse las entradas de direccionamiento **a<sub>12</sub> - a<sub>0</sub>** de este bloque?

A <sub>15</sub>	A <sub>14</sub>	A <sub>13</sub>	A <sub>12</sub>	A <sub>11</sub>	a <sub>12</sub>	a <sub>11</sub>	
0	0	1	0	0	0	0	
0	0	1	0	1	0	1	
0	0	1	1	0	1	0	$a_{12} = A_{14} + A_{12}$
0	1	0	0	0	1	1	$a_{11} = A_{14} + A_{11}$

### 19.5. Memorias de sólo lectura

Los codificadores **ROM**, aunque no contienen registros, pueden ser considerados como «memorias de sólo lectura» (a lo cual alude su propia denominación ROM: *read only memory*, debida a que ésta fue la primera aplicación de los grandes codificadores con estructura ROM).

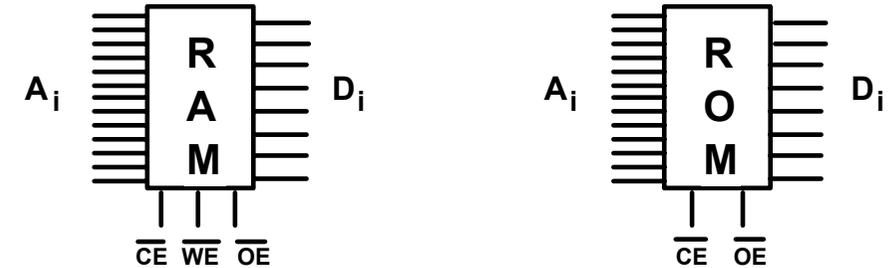
Un codificador proporciona un vector de salida (dato) para cada vector de entrada (dirección) que recibe; es como si a una «dirección» recibida a través de sus entradas respondiese con un «dato» en sus líneas de salida; el resultado es la obtención en las salidas de una palabra binaria, seleccionada por el número binario que hay en sus entradas.

Ciertamente la relación entrada – salida (dirección – dato) es meramente combinatorial y no hay «memoria» en el sentido propio de los sistemas secuenciales; pero, funcionalmente, un codificador presenta un conjunto de palabras binarias numeradas.

En tal sentido, el comportamiento de un codificador equivale al de una memoria de acceso directo cuyos registros estuviesen ya escritos con información fija y solamente se pudieran leer: *memoria de sólo lectura*. Las entradas del codificador corresponden a las líneas de direcciones y las salidas a las líneas de datos en una operación de lectura.

Por otra parte, para poder conectar el codificador ROM a un bus de datos será necesario dotar a sus salidas de capacidad tri-estado, con la posibilidad de desconexión (alta impedancia) controlada por una línea de habilitación (CE), de forma que pueda compartir las líneas de datos con otros bloques RAM o ROM.

Por analogía con los circuitos integrados **RAM**, los codificadores **ROM** suelen tener dos entradas de control: **CE** para habilitar el bloque y **OE** para habilitar las salidas del mismo (lectura), ambas activas con valor booleano **0**. [**WE** no tiene sentido por cuanto que la escritura en este tipo de bloques no es posible.]



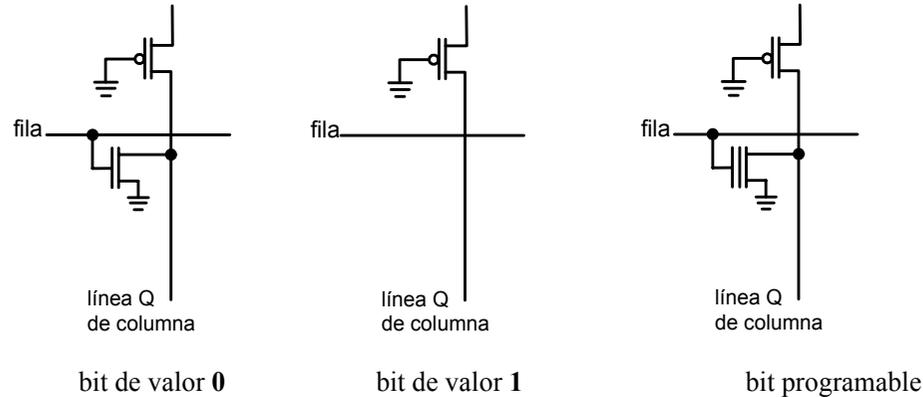
Las memorias de sólo lectura son útiles para datos fijos, tablas de valores, tablas de conversión, tablas funcionales, etc., y, en particular, para los programas específicos de los microprocesadores dedicados a aplicaciones de control.

La disponibilidad de codificadores **ROM programables** permite contar con «memorias de sólo lectura» cuya «escritura» previa puede realizarse mediante programación. De forma que las palabras binarias contenidas en una «memoria» ROM pueden venir fijadas «de fábrica», resultantes de la inclusión o no de transistores en los correspondientes nudos de la matriz "O", o bien pueden ser «escritas» por el diseñador (codificadores programables **PROM**), a través de un programador, previamente a su utilización circuital.

La estructura de una memoria ROM es análoga a la de una RAM, prescindiendo de los circuitos correspondientes a la escritura y utilizando una sola línea Q (en lugar de las dos líneas Q y Q̄); la celda básica, para cada bit, consistirá en la presencia/ausencia de un transistor en el nudo correspondiente, según que el valor de dicho bit sea **0/1**.

La figura de la página siguiente muestra tal estructura; en ella se observa que el transistor PMOS «pone» la línea Q a **1** y dicho valor permanece cuando no hay transistor NMOS en la fila seleccionada; en cambio, cuando el transistor NMOS está presente en dicha fila, conducirá si está seleccionada y llevará la línea Q a **0**.

En el caso de codificadores **PROM** se incluye un transistor **EPROM** programable para cada bit, de forma que equivale a valor **1** (el transistor no conduce nunca) cuando se carga negativamente su puerta aislada y, en caso de que dicha puerta esté descargada, implica valor **0** (el transistor conduce al ser seleccionado y lleva la línea Q a 0 V).



Configuración y conexiones de uno de los «biestables» ROM

Existen tres tipos de transistores MOS programables: EPROM, E<sup>2</sup>PROM y FLASH (ver apartado 9.5., primer volumen). En cuanto a la forma de programarlos (y, también, en cuanto a la velocidad de lectura) no hay diferencias funcionales entre ellos; se distinguen, en cambio, en que el «borrado» de los primeros puede hacerse con luz ultravioleta y los otros dos permiten hacerlo mediante tensión eléctrica (de signo contrario a la de su programación). En el caso E<sup>2</sup>PROM el borrado es individual (cada transistor se puede programar –bit a 1– o borrar –bit a 0– individualmente), mientras que en el caso FLASH el borrado es global (se borra todo el bloque, mediante una tensión eléctrica positiva aplicada al terminal de fuente que es común a todos los transistores).

Cualquiera de estos tres tipos puede ser utilizado como ROM, memoria de sólo lectura, programando previamente su contenido a través de un programador.

#### Memorias de sólo lectura reprogramables

Los transistores MOS tipo E<sup>2</sup>PROM son reprogramables eléctricamente, es decir, puede «escribirse» en ellos un 1 (transistor programado) o un 0 (transistor borrado) mediante la aplicación a la puerta de una tensión positiva o negativa relativamente alta. Esta capacidad de programación individual de un 0 o un 1 sobre cada bit ha permitido la construcción de memorias ROM que se pueden re-escribir en el propio circuito funcional, mediante una operación de «escritura» a través de los buses.

Dicha operación requiere tiempos más amplios que los normales de acceso a una memoria RAM ya que es necesario efectuar la programación de la correspondiente palabra binaria sobre los transistores E<sup>2</sup>PROM, cargando con electrones su puerta aislada (caso de un bit a 0) o descargándola (caso de programar un 1).

Además tal operación requiere, por lo general, tensiones más elevadas que las habituales de alimentación digital y necesita los correspondientes circuitos de control de la programación; tanto las tensiones como los circuitos adicionales se incorporan dentro del circuito integrado en su diseño y fabricación, de forma que, desde el exterior, actúa como una memoria de acceso directo con capacidad de lectura y de escritura, solo que la escritura es lenta y requiere varios ciclos de reloj.

Incluso, se fabrican memorias RAM duplicadas con otra ROM dentro del propio circuito integrado para evitar la pérdida de información cuando dejan de estar alimentadas a la correspondiente tensión eléctrica: cuando se detecta una «caída» de tensión, un circuito de control determina el traspaso de todos los datos del bloque RAM al bloque ROM disponible en el interior del mismo circuito integrado y, viceversa, al recuperarse la tensión de alimentación el control reescribe el bloque RAM con los datos guardados en la parte ROM del integrado.

Con la misma finalidad, existen series especiales de circuitos integrados RAM no volátiles, dotados de una batería interna (recargable con la propia alimentación del integrado) que conserva los valores almacenados en sus registros, en ausencia de tensión de alimentación.

#### Memorias FLASH de acceso secuencial

Un bloque ROM programable tipo FLASH constituye una memoria de lectura rápida, con capacidad de escritura lenta (ya que debe hacerse por programación) si el bloque ha sido previamente «borrado»; en las memorias FLASH el borrado se efectúa en forma global (se borra todo el bloque a la vez) ya que los transistores, una vez programado, no pueden borrarse individualmente.

Las reducidas dimensiones de los transistores FLASH y de su agrupación en configuración ROM han permitido la integración de memorias FLASH de muy alta capacidad. Tales memorias se utilizan para el almacenamiento masivo de datos, con una funcionalidad análoga a la de los disquetes o discos compactos CDs y con importantes ventajas sobre ellos al no necesitar un sistema mecánico para su lectura.

Para estas aplicaciones de memorias de conservación y transporte de datos (o de memoria de acumulación de datos en un sistema de adquisición de los mismos) se utilizan memorias FLASH serie, con acceso secuencial que se gestiona a través de contadores incluidos en la propia memoria. Estas memorias permiten escribir los datos en ellas y recuperarlos posteriormente a través de una simple entrada serie.