

# ***DISPOSITIVOS***

# ***DE***

# ***ALMACENAMIENTO (I)***

Autores: Fco. Javier Quiles Latorre  
Manuel A. Ortiz López  
Miguel A. Montijano Vizcaíno

**ÁREA DE CONOCIMIENTO DE**  
**ARQUITECTURA Y TECNOLOGÍA DE COMPUTADORES**  
**UNIVERSIDAD DE CORDOBA**

# ÍNDICE

1. Introducción.
2. Fundamentos de las memorias.
  - 2.1. Terminología utilizada.
3. Memorias semiconductoras.
  - 3.1. Introducción.
  - 3.2. Memorias de Acceso Aleatorio (RAM).
    - 3.2.1. Fundamentos.
    - 3.2.2. Estructura de una RAM semiconductora.
    - 3.2.3. Tipos de RAM.
    - 3.2.4. RAM Estática. SRAM.
    - 3.2.5. RAM Dinámica. DRAM.
      - 3.2.5.1. Matriz de celdas y funcionamiento de la celda de memoria.
      - 3.2.5.2. Estructura interna de una DRAM.
      - 3.2.5.3. Conexión de una DRAM.
      - 3.2.5.4. Cronograma de los ciclos de lectura/escritura.
    - 3.2.6. RAM no volátil.
      - 3.2.6.1. NVRAM con pila de litio interna.
    - 3.2.7. Asociación de memorias de acceso aleatorio.
      - 3.2.7.1. Asociación de SRAM.
  - 3.3. Memorias de Sólo Lectura. ROM.
    - 3.3.1. Estructura de una ROM.
    - 3.3.2. Clasificación de las ROM.
    - 3.3.3. ROM Programables. PROM.
    - 3.3.4. ROM Programables y Borrables. EPROM.
      - 3.3.4.1. UVPROM
      - 3.3.4.2. PROM Borrables Eléctricamente. EEPROM.
      - 3.3.4.3. Flash EPROM.
    - 3.3.5. Aplicaciones de las memorias ROM.
  - 3.4. Memorias de Acceso Secuencial.
    - 3.4.1. Memoria secuencial con registro de desplazamiento circular.
    - 3.4.2. Memoria LIFO.
    - 3.4.3. Memoria FIFO.

## BIBLIOGRAFIA

- [1]    Introducción al Diseño Lógico Digital.  
     John P. Hayes  
     Addison-Wesley Iberoamericana  
     Año 1996
  
- [2]    Circuitos Digitales y Microprocesadores  
     Herbert Taub  
     Mc-Graw Hill  
     Año 1983
  
- [3]    Sistemas Digitales. Principios y aplicaciones  
     Ronald J. Tocci  
     Prentice Hall  
     Año 1993
  
- [4]    Fundamentos de los Computadores  
     P. De Miguel Anasagasti  
     Paraninfo  
     Año 1992
  
- [5]    Diseño Digital  
     M. Mano  
     Prentice Hall  
     Año 1987

## 1. INTRODUCCION.

Una ventaja importante de los sistemas digitales sobre los analógicos es la capacidad de almacenar fácilmente grandes cantidades de información digital, por lo que los sistemas digitales son más flexibles y adaptables a muchas situaciones.

Un ejemplo de un sistema digital con capacidad de almacenamiento de información es un computador digital. En la figura 1-1 se muestra la arquitectura general de un computador digital, en la que se pueden distinguir tres bloques fundamentales: Unidad Central de Proceso(CPU), Memoria Principal (MP) y Unidades de Control de Entrada/Salida.

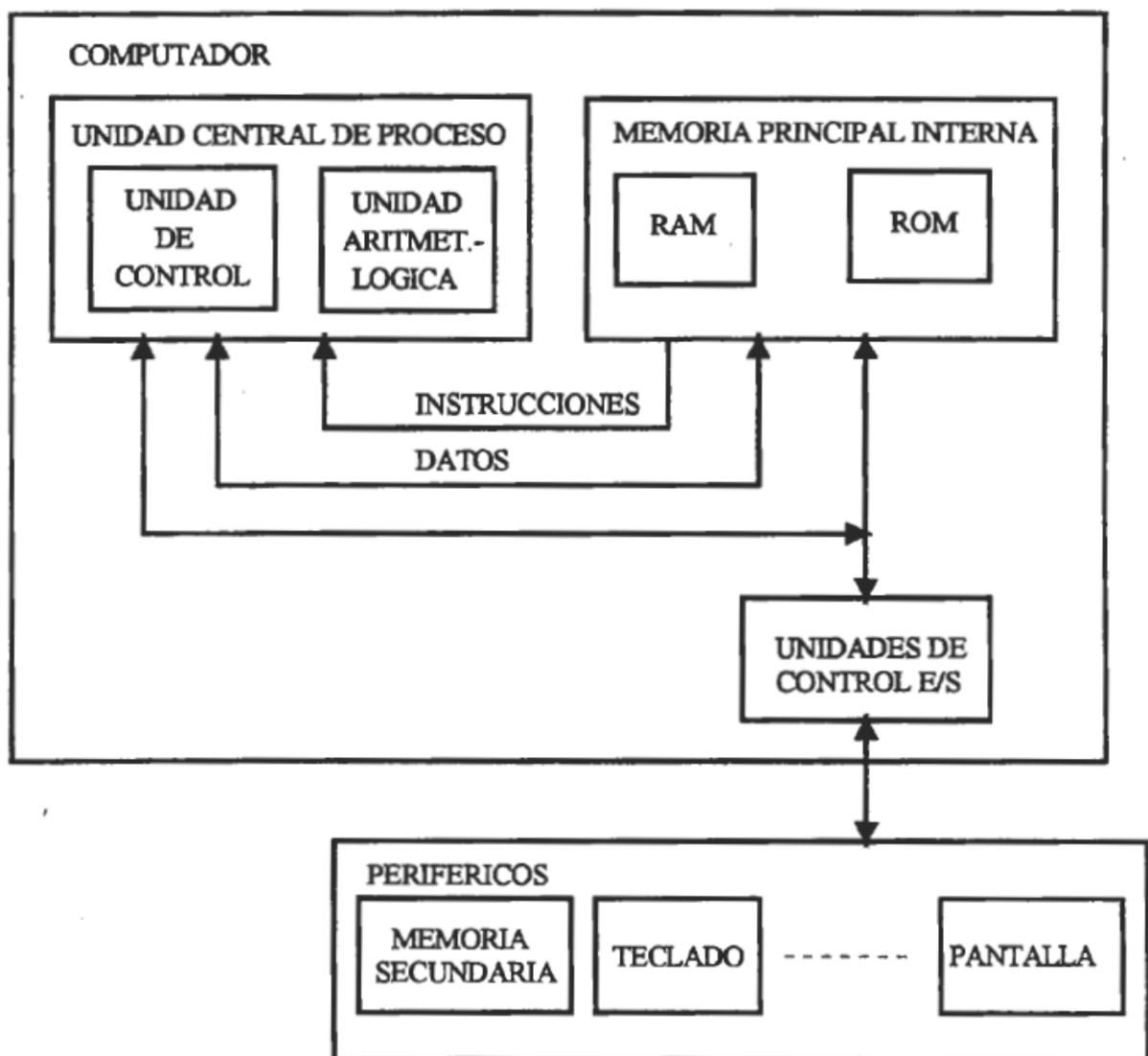


Figura 1-1 Arquitectura general de un Computador

La principal función de un computador es procesar información. Para ello, necesita saber qué tipo de procesamiento debe realizar y los datos que utilizará. El tipo se especifica mediante un programa, que es un conjunto de instrucciones u operaciones básicas, que ejecuta el computador (Aritméticas: suma, resta, multiplicación, división..., Lógicas: AND, OR, NOT, XOR ..., Transferencia de datos: cargar, almacenar, transferir ..., Saltos y Bifurcaciones condicionales, etc.).

Por tanto, se necesita un soporte que almacene tanto el programa como los datos. El bloque de Memoria Principal interna es el que realiza esta función.

La CPU se encarga de ejecutar las instrucciones del programa y de acceder a los datos cuando sea necesario. Ésta se divide en dos bloques: Unidad de Control (UC) y Unidad Aritmético-Lógica (ALU). La UC genera las señales de control que sincronizan el funcionamiento de todos los bloques del computador. La ALU realiza las operaciones básicas comentadas anteriormente.

Para ejecutar un programa, la UC captura las instrucciones de la MP, las decodifica y según el tipo de instrucción, genera las señales de control necesarias:

- Si la instrucción opera con datos deberá acceder a la MP para transferirlos a la ALU.
- Proporciona las señales necesarias a la ALU para que realice la operación correspondiente a la instrucción.
- Controlará las Unidades de Control de E/S si la instrucción accede a alguno de estos dispositivos.

Puede ocurrir que el programa y los datos tengan un tamaño elevado por lo que no se pueden almacenar en la MP. En este caso, se utiliza la memoria secundaria (MS), que se considera como un periférico o unidad de E/S del computador. La MS es más lenta que la principal, pero tiene una mayor capacidad de almacenamiento.

Como para ejecutar un programa debe estar en la MP, se deben establecer los mecanismos necesarios para transferir éste desde la MS a la MP.

Una forma de determinar el rendimiento de un computador es por el número de instrucciones que ejecuta por segundo (MIPS), de forma que generalmente un computador con un número MIPS elevado indica, que éste tiene una gran capacidad de procesamiento. Al ser la UCP el bloque encargado de la ejecución del programa, debe funcionar a la mayor velocidad posible. Por ello, se implementa con dispositivos semiconductores, que actualmente debido a los avances tecnológicos pueden trabajar a frecuencias muy elevadas (del orden de 150 MHz). Como la MP suministra las instrucciones y los datos a la UCP, debe ser lo más rápida posible, por lo que la MP debe implementarse también con dispositivos semiconductores.

En este tema se analizarán solamente las memorias semiconductoras. Según el tipo de memoria, el elemento básico de almacenamiento es un flip-flop o un condensador. Como ya se estudió, un flip-flop almacena un bit de información y un conjunto de éstos forman un registro, que puede almacenar una palabra, es decir un conjunto de bits.

Por otra parte, la información digital se puede almacenar como una carga en un condensador, que es un dispositivo acumulador de energía eléctrica. Este tipo de almacenamiento tiene la ventaja frente al anterior, que además de tener un consumo más reducido, su tamaño es menor, por lo que permite una mayor densidad de integración.

## 2. FUNDAMENTOS DE LAS MEMORIAS

Una memoria es un dispositivo que almacena información. Para que una memoria sea útil debe permitir dos operaciones:

- ✓ **Operación de almacenamiento.** Consiste en grabar la información en una posición de memoria. Esta se especifica mediante un conjunto de señales denominadas dirección. Se distinguen dos tipos de operaciones de almacenamiento:
  - ✓ **Operación de escritura.** Es una operación en la que se almacena información en una memoria, pero cuando ésta está funcionando en un sistema digital. Siempre que se escribe una palabra en una posición de memoria, ésta sobrescribe la palabra que se encontraba anteriormente.
  - ✓ **Operación de programación:** La operación de almacenar la información se realiza cuando la memoria no está trabajando en el sistema digital y debe hacerse mediante un equipo especial, denominado equipo programador. Otra característica es que antes de almacenar la información debe borrarse su contenido.
- **Operación de lectura.** Operación con la que se extrae de la memoria el contenido de la posición de memoria indicada por la dirección. También se denomina operación de extracción (fetch). En la figura 2-1 se representa mediante un bloque una memoria genérica.

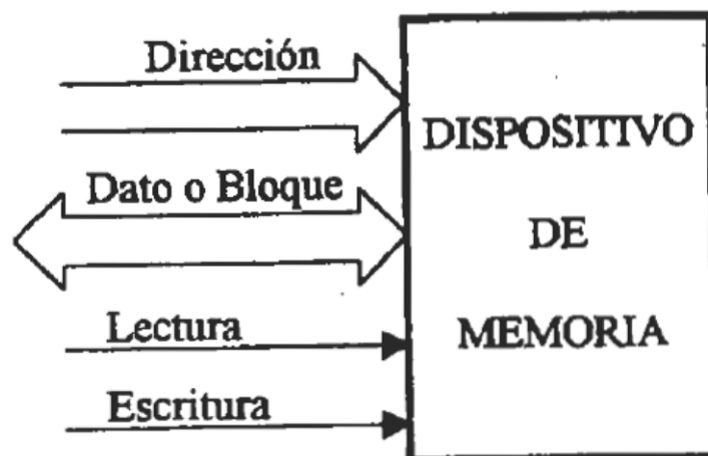


Figura 2-1 Bloque de una memoria genérica

- El tipo de operación se indica mediante las señales de control de lectura y escritura. Las señales de datos son bidireccionales, es decir son de entrada en la operación de escritura y de salida en la de lectura. Como posteriormente se verá, hay memorias que tienen señales de datos independientes para lectura y para la escritura.

## 2.1. TERMINOLOGIA UTILIZADA.

- Antes de estudiar los distintos tipos de memoria, vamos a analizar los términos básicos que se utilizan.
- **Celda de memoria.**
  - √ Es el elemento básico de memoria que también se denomina **punto de memoria**. Está constituido por un dispositivo que almacena un solo bit (1 o 0). Este dispositivo depende de la tecnología y del tipo de memoria. En las memorias de semiconductores está constituido generalmente por un flip-flop o un condensador. En las memorias secundarias como cintas y discos magnéticos está constituido por pequeñas áreas que se magnetizan.
- **Palabra de memoria.**
  - √ Como se puede ver en la figura 2-2, una memoria consta de varios registros, cada uno de los cuales almacena varios bits, es decir, está formado por varias celdas de memoria. Estos registros se denominan palabras. Por tanto, una palabra es un conjunto de bits que almacenan información: instrucciones o datos. Los accesos a una memoria se hacen a nivel de palabra, es decir, cuando se realiza una operación de escritura o lectura, se escriben o leen, respectivamente, todas las celdas de una única palabra. Por tanto, una definición más exacta sería: conjunto de celdas de memoria que se escriben o leen en una operación de lectura o escritura. El tamaño de las palabras en los computadores modernos varía de 4 a 64 bits. En la figura 2-2 cada palabra tiene un tamaño de 4 bits.

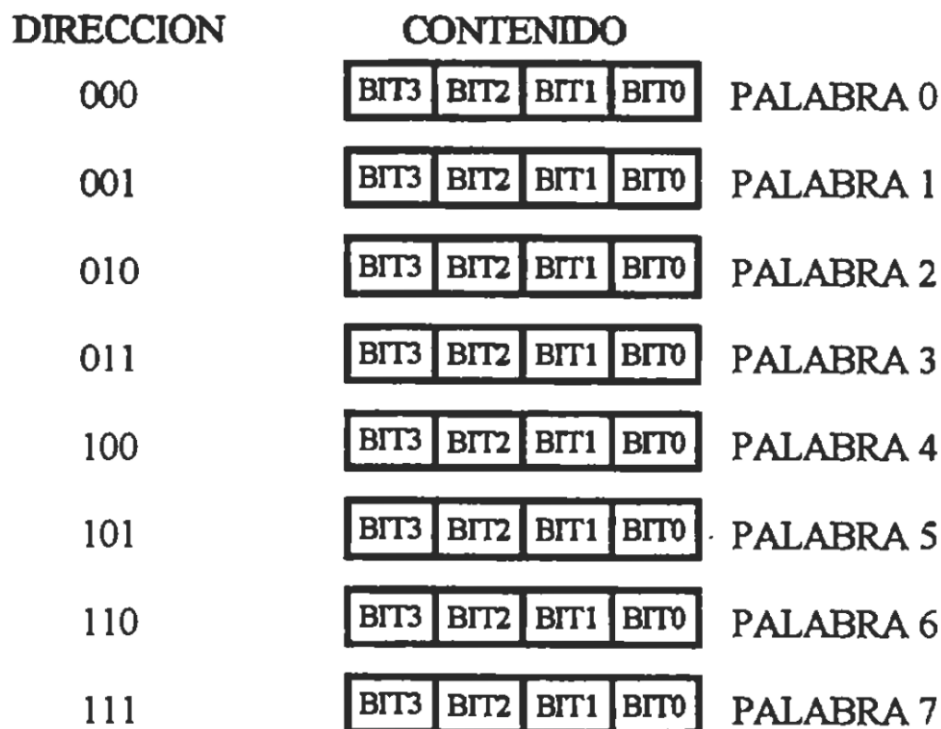


Figura 2-2 Memoria de 8x4

- Byte.
  - ✓ Es una palabra de 8 bits. Generalmente, el tamaño de las palabras se expresa en bytes. En este caso, se representa abreviadamente con **B**. Si se expresa en bits se utilizará **b** como notación abreviada.
- **Capacidad.**
  - ✓ Es el número de bits que se pueden almacenar en una memoria. La memoria de la figura 2-2 tiene 8 palabras de 4 bits, con lo que su capacidad será el resultado de multiplicar el número de palabras por el tamaño de la palabra, es decir 32 bits.
  - ✓ También se puede expresar la capacidad como **número de palabras x número de bits/palabra**. Para la memoria de la figura sería 8 x 4. Esta forma indica la **organización de la memoria**.
  - ✓ Las memorias actuales tienen una capacidad elevada, por lo que se utilizan anotaciones abreviadas para expresarlas. Se suelen usar los siguientes términos:
    - ❑ 1K Equivale a 1.024 ( $2^{10}$ ). Ejemplo: 4K = 4 x 1024 = 4096
    - ❑ 1M Equivale a 1.024K ( $2^{20}$ ) = 1.048.576. Ejemplo: 4M = 4 x 1048576.
    - ❑ 1G Equivale a 1.024M ( $2^{30}$ ).
    - ❑ 1T Equivale a 1.024G ( $2^{40}$ ).
- **Dirección.**
  - ✓ Número que identifica la posición de una palabra o un bloque en la memoria.
  - ✓ En las memorias semiconductoras de direccionamiento cableado hay tantas direcciones como palabras tenga la memoria, por lo que cada palabra tiene una dirección única que la identifica. Si una memoria tiene **m** palabras, se necesitan **n** bits de direcciones para poder seleccionar cada una de ellas, cumpliéndose que  $2^n = m$ . Por tanto, la capacidad de una memoria siempre será un múltiplo de 2, ya que también el tamaño de la palabra es una potencia de 2.
 

La memoria de la figura 2-2 consta de 8 palabras, por lo que deberá tener 3 bits de direcciones, ya que  $2^3 = 8$ . Estos 3 bits permiten el rango de 000 (0) a 111 (7). Para acceder a una palabra (palabra 3) hay que utilizar su dirección (011), que la identifica. Por ejemplo, una memoria de 16K x 8 tiene 16K palabras de 8 bits, y necesitará 14 bits de direcciones ( $16K = 2^4 \times 2^{10} = 2^{14}$ ), para identificar todas las palabras.
- **Tiempo de acceso ( $t_{acc}$ )**
  - ✓ Se define como el tiempo que una memoria tarda en realizar una lectura, y equivale al tiempo transcurrido entre el instante en el que se pone la dirección de la palabra que se desea leer y el instante en que la memoria pone el contenido de la palabra en sus salidas de datos.



- **Tiempo de ciclo de memoria ( $t_c$ )**

- ✓ Es el tiempo mínimo que debe transcurrir entre dos operaciones consecutivas a una memoria. Es siempre mayor o igual que el tiempo de acceso, ya que los componentes de algunas memorias necesitan un tiempo de recuperación entre dos accesos consecutivos.

- **Modos de acceso.**

- ✓ **Acceso Aleatorio.** Utiliza el mecanismo de direccionamiento cableado, por lo que cada palabra de memoria tiene asociada una dirección de memoria única. Así, se accede directamente a la información deseada independientemente de su posición, por lo que el tiempo de acceso es el mismo para cualquier dirección de memoria. La mayoría de las memorias de semiconductores son de acceso aleatorio y se denominan RAM.
- ✓ **Acceso Secuencial.** Una memoria de acceso secuencial se denomina SAM. Está dividida en unidades de datos, denominados registros. En la cabecera (inicio) de cada bloque se almacena la información de direccionamiento, que lo identifica. Por tanto, para operar con un registro se debe ir leyendo todos los registros precedentes hasta encontrar el que se desea acceder. Así, el tiempo de acceso no es constante, sino que varía según la posición del registro, ya que para acceder a él hay que pasar antes por todos los que le preceden. Por esto, los tiempos de acceso de la SAM son mayores que los de las RAM. Ejemplo: la cinta magnética.
- ✓ **Acceso Asociativo.** Utilizan el acceso aleatorio en la escritura, pero en la operación de lectura se accede por su contenido. Es decir, en vez de suministrar a la memoria la dirección de la palabra a leer, se le indica su contenido. La memoria compara el dato con el de todas sus palabras, y suministra la dirección de coincidencia o un contenido asociado a ella, o una indicación de que no hay equivalencia.

La comparación con todas las palabras se realiza simultáneamente, por lo que el tiempo de acceso es independiente de su posición y del contenido. Las memorias caché pueden emplear acceso asociativo.

- **Volatilidad.**

Una memoria es volátil cuando no mantiene la información al desaparecer la alimentación. Las memorias de semiconductores pueden ser volátiles o no volátiles. Las memorias magnéticas son no volátiles.

Por el contrario una memoria es no volátil, si mantiene permanentemente la información de los puntos de memoria, mientras no se realice una operación de escritura sobre ellos. Por ejemplo, los discos magnéticos.

- ***Memoria de lectura y escritura (RWM).***

Cualquier memoria en la que se puede leer y escribir información un número elevado de veces. Las memorias de semiconductores, tanto estáticas como dinámicas, son de este tipo.

- ***Memoria de sólo lectura (ROM).***

Es una memoria que sólo tiene la operación de lectura. En términos técnicos una ROM sólo puede escribirse (programarse) una vez y ésta operación normalmente se efectúa en fábrica. Algunas memorias semiconductoras no volátiles derivadas de las ROM como las EEPROM y EPROM FLASH, pueden escribirse pero en realidad se trata de una operación de borrado seguida de otra de programación y no una operación específica de escritura. Todos los tipos de memorias ROM son no volátiles.

- ***Memoria estática (SRAM).***

Es una memoria de semiconductores en los cuales los datos almacenados permanecen mientras se aplique la energía de alimentación, sin necesidad de escribir periódicamente los datos en la memoria. Utiliza generalmente flip-flop como celda de almacenamiento.

- ***Memoria dinámica.***

Es una memoria de semiconductores en las que los datos se van degradando paulatinamente aunque esté alimentada. Para que no desaparezca la información debe reescribirse periódicamente. Esta operación se conoce como de *refresco*. Utiliza condensadores como celda de almacenamiento.

### 3. MEMORIAS SEMICONDUCTORAS.

#### 3.1 INTRODUCCIÓN.

Todas las memorias que usan como soporte o medio de almacenamiento dispositivos semiconductores integrados en un circuito integrado se denominan memorias semiconductoras. La clasificación de las memorias semiconductoras se hace en base a varios criterios. Primero se clasifican según el tipo de operaciones que permiten realizar: sólo lectura y lectura/escritura. Las memorias de lectura/escritura se clasifican en función del tipo de acceso: aleatorio y secuencial.

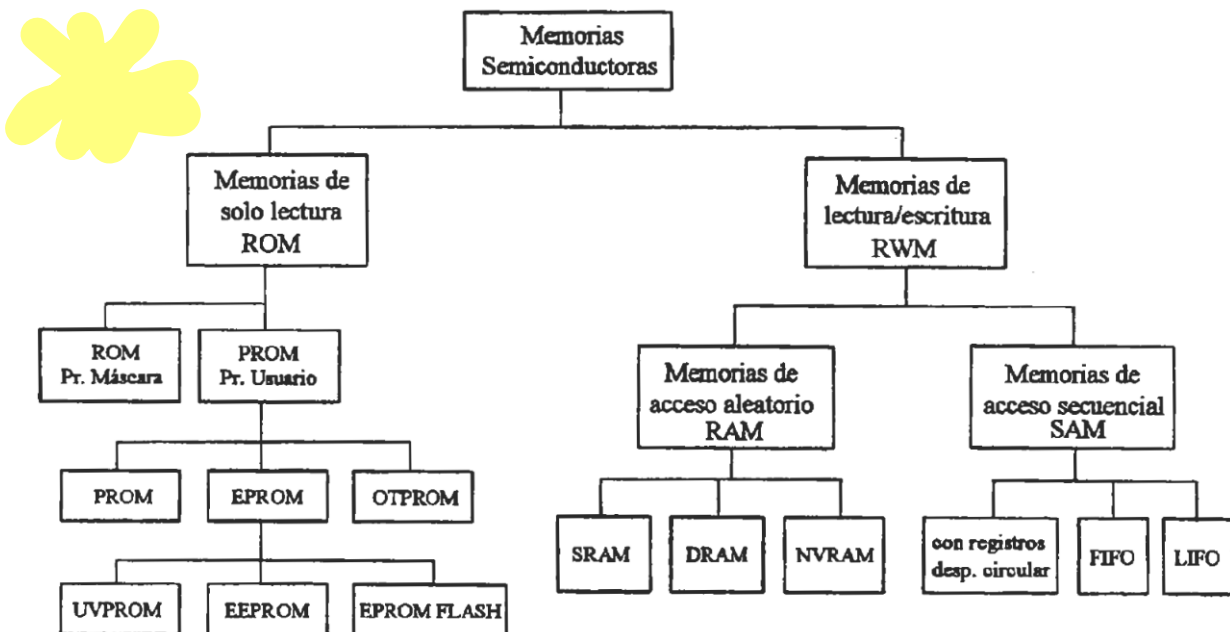


Figura 3-1 Clasificación de las memorias semiconductoras

#### 3.2. MEMORIAS DE ACCESO ALEATORIO (RAM).

##### 3.2.1. FUNDAMENTOS.

El término RAM significa Random Access Memory, es decir memoria de acceso aleatorio. Como ya se comentó anteriormente, en una RAM se accede directa e independientemente a cada una de las palabras de la memoria, por lo que el tiempo de acceso es siempre el mismo para cada una de ellas. En este tema se estudiarán las RAM de semiconductores.

Antes de analizar la estructura de una memoria RAM de semiconductores, se expondrán las patillas que se necesitan para controlar su funcionamiento. En la figura 3-2 se muestra la representación básica de un circuito integrado de memoria de acceso aleatorio, que tiene las operaciones de lectura y escritura.

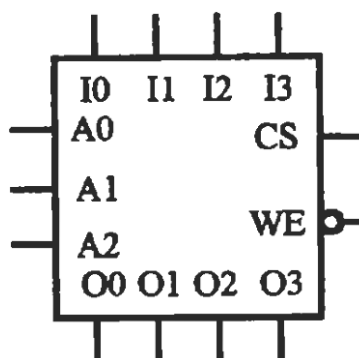


Figura 3-2. Representación básica de una RAM genérica

Las patillas A2, A1 y A0 son las líneas de direcciones. Una dirección es un número binario, que indica la posición de una palabra en la memoria. Como cada palabra tiene una dirección única que la identifica, deben haber tantas direcciones como palabras tenga la memoria. La memoria por tanto tendrá 8 palabras, ya que con tres bits de direcciones se obtienen 8 combinaciones binarias ( $2^3$ ). Para implementar la operación de escritura dispone de 4 patillas para los datos de entrada (I0 ..I3). Las patillas O3, O2, O1 y O0 son los datos de salida, que se utilizan en la operación de lectura. Al tener 4 patillas de datos la anchura de palabra será de 4 bits, y la memoria tendrá una organización de 8 x 4 y una capacidad de 32 bits.

La patilla CS es la abreviatura de la expresión inglesa *chip select*, que significa selección de pastilla. Según la representación de la figura es activa a nivel alto, de forma que para realizar una operación de lectura o escritura CS debe estar a 1 lógico. Si CS está a 0 lógico la memoria no se selecciona y no se puede leer ni escribir. En ese caso se dice que está en el modo de espera o standby en el que el consumo de potencia es muy reducido.

WE (Write Enable) determina el tipo de operación. WE significa habilitación de escritura, y como tiene un círculo en la patilla será activa a nivel bajo. Así, si WE es 0 la operación será de escritura y si es 1 será de lectura.

En la figura 3-2 se ha expuesto una RAM genérica, pero existen RAM comerciales con otra distribución de patillas:

- Dos patillas independientes para habilitar las operaciones de lectura y escritura (RD y WR respectivamente). En este caso ambas son activas a nivel bajo.
- Patillas de datos únicas, por lo que serán bidireccionales, es decir se utilizan para los datos de entrada en la escritura y para los datos de salida en la lectura.
- Dos patillas de selección (CS1 y CS2), una activa nivel alto y otra a nivel bajo.

Aunque en un apartado posterior se analizarán los cronogramas de las operaciones de lectura y escritura en una RAM, a continuación se indican los pasos necesarios, suponiendo que se accede a la palabra 3.

- **Operación de escritura.**

☐ Direccionar la palabra 3 poniendo A2 .. A0 al valor binario 011 (3).

☐ Poner el dato a escribir en los terminales de entrada de datos I3 .. I0.

- ☐ Seleccionar la memoria poniendo a nivel alto CS.
- ☐ Activar señal de escritura WE.
- ☐ La escritura de la palabra se realizará al desactivar CS o WE.
- **Operación de lectura.**
  - ☐ Direccionar la palabra 3 poniendo A2 .. A0 al valor binario 011 (3).
  - ☐ Seleccionar la operación de lectura desactivando WE.
  - ☐ Seleccionar la memoria poniendo a nivel alto CS.
  - ☐ La memoria pondrá el contenido de la palabra 3 en las patillas de datos de salida.

### 3.2.2. ESTRUCTURA DE UNA RAM SEMICONDUCTORA.

Una vez que se han expuesto las características básicas de una RAM de 8 x 4, estudiaremos su estructura para comprender porque se denomina memoria de acceso aleatorio, y porque el tiempo de acceso es igual para todas las palabras. Se analizará su estructura lógica sin tener en cuenta su construcción a nivel de semiconductores. En la figura 3-16 se muestra la estructura lógica de una RAM de 4 x 2. Para facilitar la comprensión se utilizan conmutadores lógicos para conectar cada palabra de la memoria con las patillas de datos para escribirla o leerla.

Los conmutadores lógicos utilizan una entrada de control que determina su estado del siguiente modo: si la entrada de control está a 0 el conmutador está abierto y si está 1 el conmutador se cierra. Por tanto, se selecciona una palabra de la memoria cuando sus correspondientes conmutadores se cierran. Estos conmutadores se implementan con dispositivos semiconductores, como por ejemplo transistores bipolares o MOS.

En la estructura de la memoria se distinguen cuatro bloques: las celdas de memoria, el decodificador, la lógica de control y las líneas de bits. Las celdas de memoria están constituidas por flip-flops básicos, es decir por dos inversores con realimentación cruzada. Como se dijo anteriormente sólo se analizará la estructura, por lo que únicamente se verá el flip-flop como un elemento de memoria, que almacena el valor lógico que haya en la línea de bit correspondiente.

Las líneas de direcciones se conectan al decodificador, que selecciona la palabra correspondiente. Como un decodificador activa solamente una de sus salidas y en concreto la correspondiente a la combinación binaria aplicada a sus entradas, debe tener un tamaño de 2 a 4. La relación entre cada dirección y la palabra seleccionada está impuesta por el funcionamiento del decodificador y se indica en la tabla 3-1.

Así, si se quiere acceder a la palabra 2, las direcciones deben tener la combinación binaria 10 (2). En este caso sólo la puerta G2 pondrá su salida a nivel alto, cerrando los conmutadores lógicos correspondientes a la palabra 2, y conectándola a las líneas de bits.

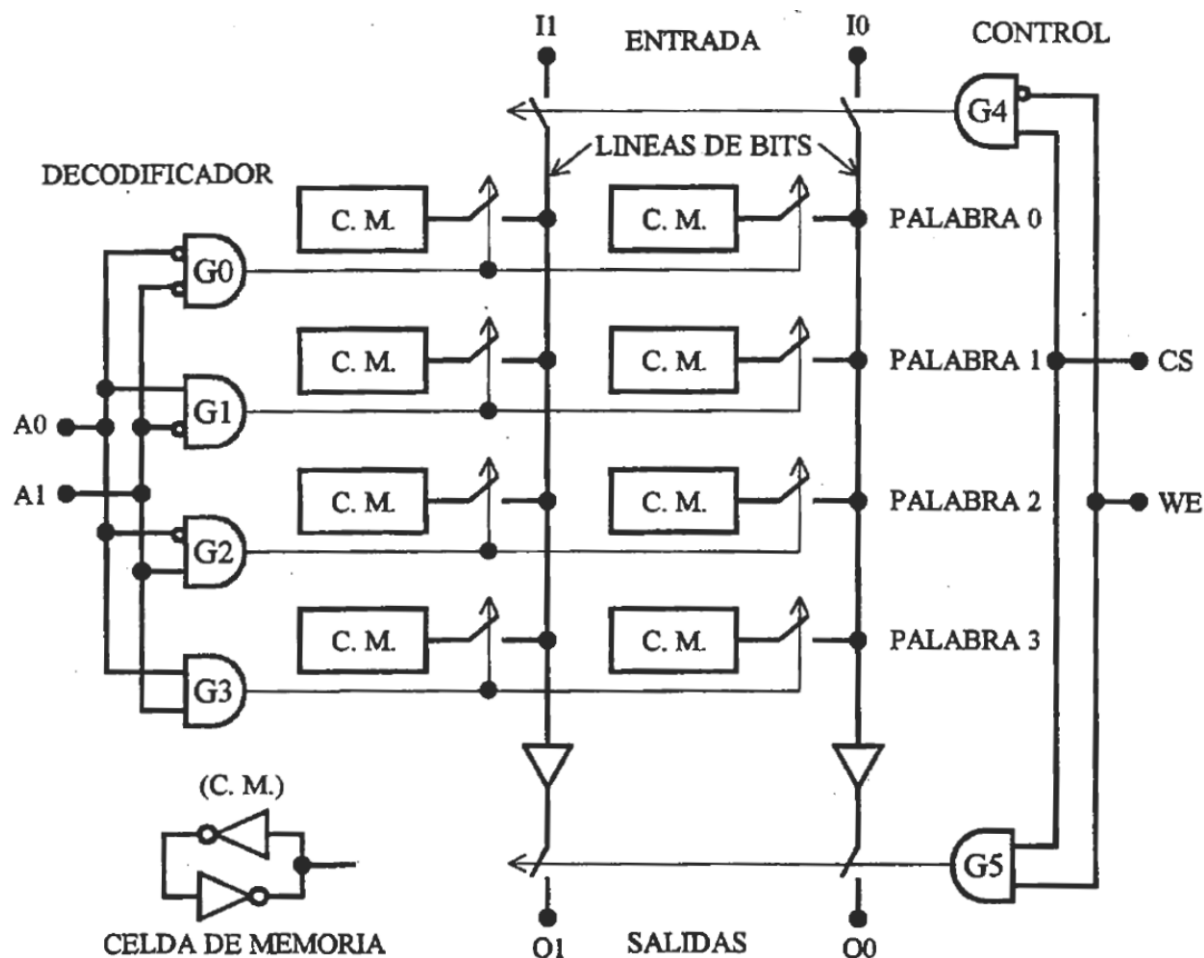


Figura 3-3. Estructura genérica de una RAM 4x2

DIRECCION A1 A0	PALABRA
0 0	PALABRA 0
0 1	PALABRA 1
1 1	PALABRA 2
1 0	PALABRA 3

Tabla 3-1

El bloque de lógica de control está compuesto por dos puertas AND, cuyas salidas controlan el estado de los conmutadores de entrada y salida de datos, según los valores de las patillas CS y WE.

- Si CS está a 0, es decir memoria no seleccionada, las salidas de ambas puertas AND estarán a 0, y por tanto, todos los conmutadores estarán abiertos. En esta situación, la palabra 2, aunque seleccionada por las direcciones, estará aislada del mundo exterior, no pudiéndose realizar ninguna operación con ella.
- Si CS está a 1, se pondrá a nivel alto la salida de una sola puerta AND, según el valor de WE, realizándose la operación de lectura o escritura:
  - ✓ Cuando WE está a 1, la puerta G5 pondrá a 1 su salida. Los conmutadores de salida de datos se cierran, conectando las líneas de bits a las patillas O1 y O0, que tendrán el contenido de la palabra 2. Por tanto, con WE a 1 y CS activo a 1 se realiza una operación de lectura en la memoria.

- ✓ Cuando WE está a 0, la puerta G4 pondrá a 1 su salida. Los conmutadores de entrada de datos se cierran, conectando las líneas de bits a las patillas I1 y I0. Los flip-flops de la palabra 2 almacenarán el valor de estas patillas, sobrescribiendo el valor anterior. Por tanto, con WE a 0 y CS activo a 1 se realiza una operación de escritura en la memoria.

En la estructura lógica analizada, las palabras de memoria forman una matriz unidimensional, necesitándose una señal para seleccionar cada palabra. Actualmente, debido al aumento de la capacidad de las memorias por las mejoras tecnológicas, se utiliza una matriz bidimensional para reducir la lógica necesaria. A continuación vamos a analizar estas dos estructuras partiendo de una memoria de 16 x 1. Como el tamaño de palabra es de 1 bit cada una estará compuesta por una celda de memoria, que se implementará con un flip-flop básico.

- En la figura 3-4 se muestra la estructura para una matriz unidimensional. Al tener una organización de 16 x 1, se necesitarán 16 señales para controlar los conmutadores lógicos de cada palabra. Por tanto, el decodificador debe tener un tamaño de 4 a 16, y estará formado por 16 puertas AND.
- Si las palabras de memoria se organizan en una matriz bidimensional, la estructura cambiará tal y como se puede ver en la figura 3-5. La dirección se divide en dos partes: fila y columna. Los bits A3 y A2 determinan la dirección de fila y los A1 y A0 la de columna. Se necesitarán dos decodificadores, el de fila y el de columna. Como cada palabra tiene una dirección única que la representa, tendrá igualmente una combinación única de dirección de fila y de columna. Así, para conectar solamente una palabra a la patilla de salida, se necesitarán dos conmutadores por palabra. Un conmutador estará controlado por una salida del decodificador de fila y el otro por una salida del decodificador de columna.

Para comprender mejor el funcionamiento, consideremos que se desea acceder a la palabra 9, por lo que las direcciones A3 .. A0 se pondrán a la combinación binaria 1001. Igualmente se considera que la señal CS está activada.

- ☐ A3,A2 = 10 activará la salida F2 del decodificador de fila, cerrándose los conmutadores CF8, CF9, CF10 y CF11. Las palabras 8, 9, 10 y 11 se conectarán a sus correspondientes líneas de bits de columna.
- ☐ A1,A0 = 01 activará la salida C1 del decodificador de columna, cerrándose el conmutador CC1. De esta manera solamente se conectará a la patilla de datos bidireccional E/S la palabra 9, pudiéndose leer o escribir su contenido, según la señal de habilitación de escritura (WE).
- La estructura bidimensional utiliza dos decodificadores de 2 a 4 por lo que se necesitan 8 puertas AND en lugar de las 16 de la estructura unidimensional, pero emplea 4 conmutadores más.

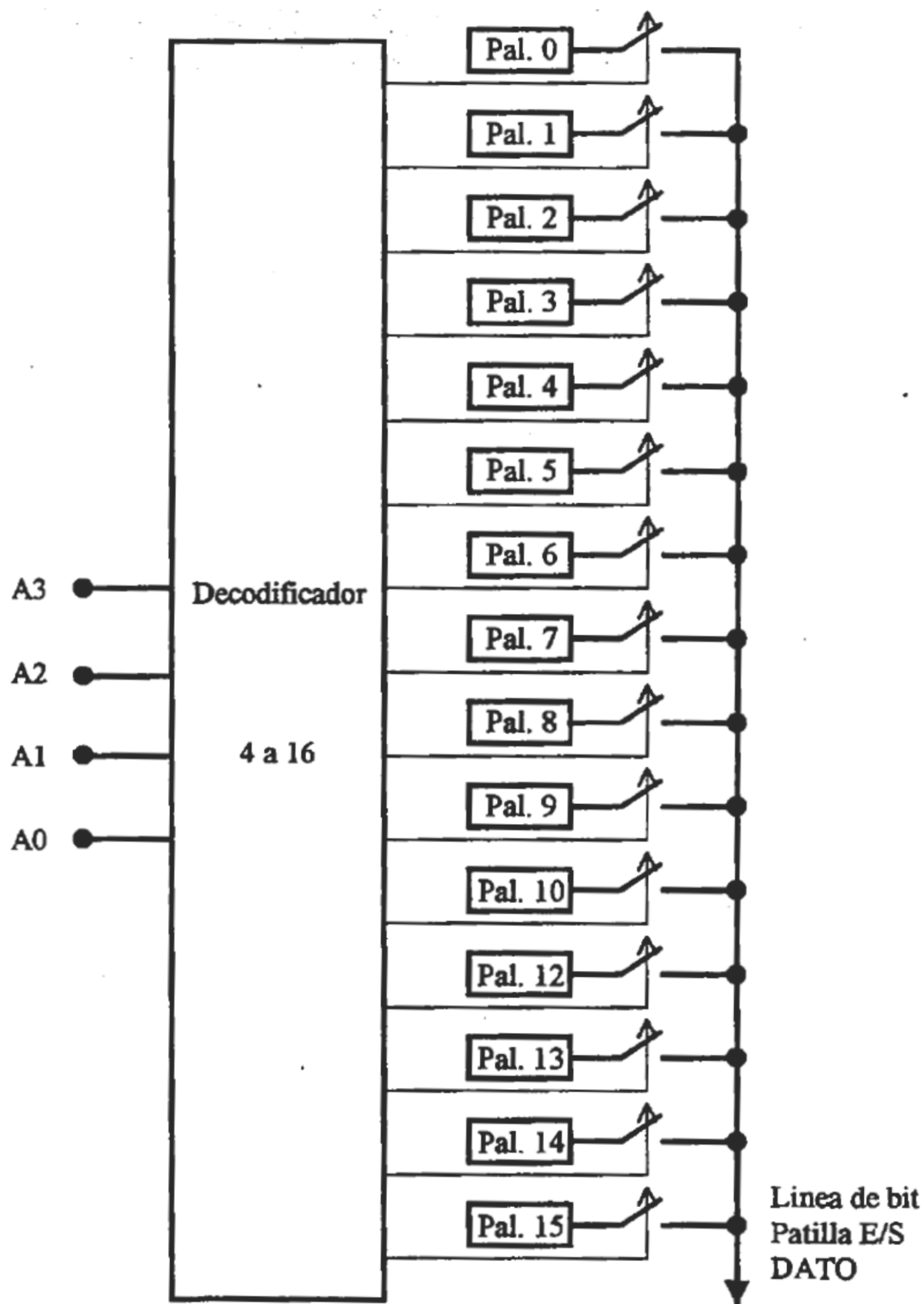


Figura 3-4. Matriz unidimensional de una memoria RAM



- Por tanto, con la estructura bidimensional se obtiene un sistema más simple. Este ahorro aumenta con la capacidad de la memoria. Las RAM actuales tienen capacidades del orden de 64 Mb. Si se considera una organización de 64M x 1 utilizarán 26 líneas de direcciones. Para implementarla con una estructura unidimensional se necesitarían 67.108.864 puertas AND ( $64 \times 1.048.576$ ) y 67.108.864 conmutadores. En una estructura bidimensional harían falta dos decodificadores de 13 a 8.192 ( $2^{13}$ ), empleándose 16.384 puertas AND ( $8.192 \times 2$ ), y 67.117.056 conmutadores ( $67.108.864 + 8.192$ ).

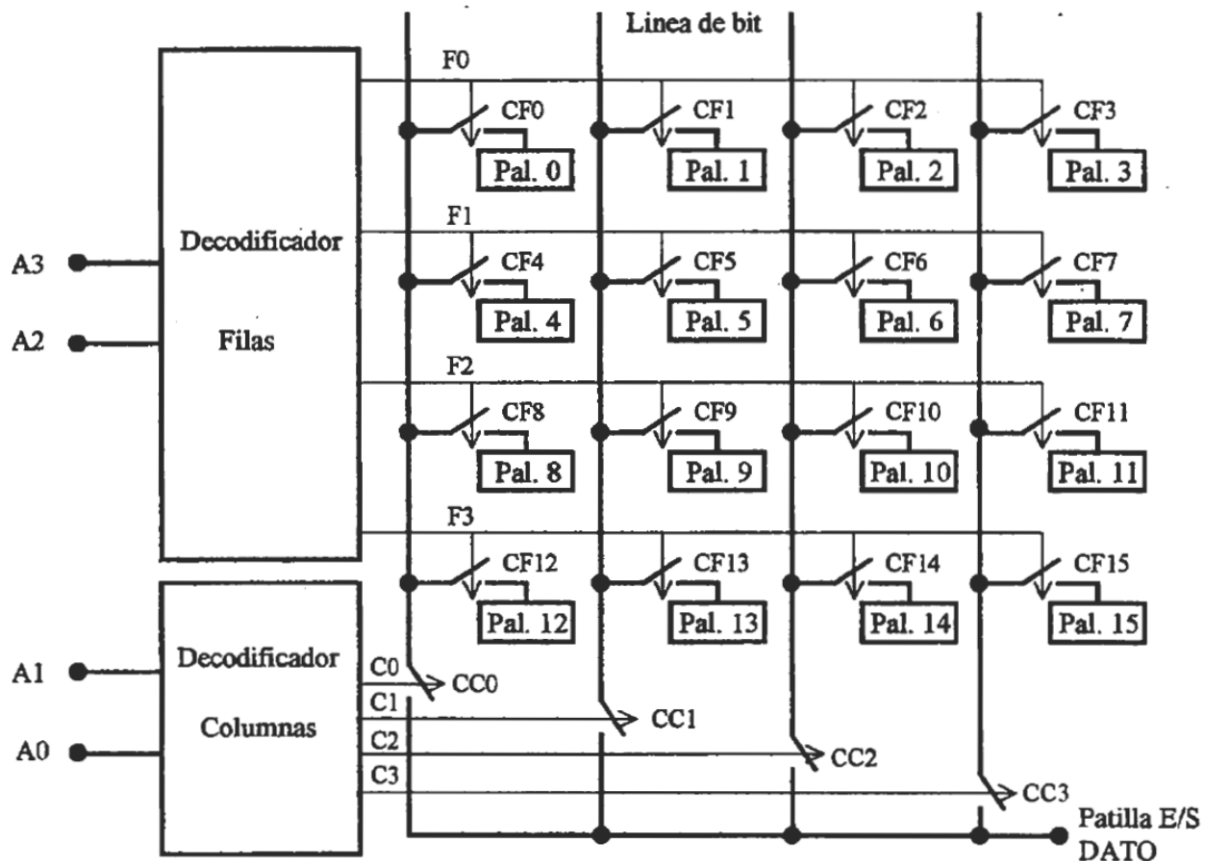


Figura 3-5. Estructura bidimensional de una RAM de 16 palabras

### 3.2.3. TIPOS DE RAM.

Fundamentalmente hay tres tipos de RAM, la RAM estática (SRAM), la RAM dinámica (DRAM) y la RAM no volátil (NVRAM). Además de éstas, aunque todos los autores las consideran un grupo aparte por tratarse un componente completamente combinacional, las memorias semiconductoras de sólo lectura (ROM) son también de acceso aleatorio.

Las SRAM y DRAM son memorias volátiles, frente a las NVRAM y ROM que son no volátiles. Ya se comentó, que una memoria era volátil, si perdía la información al quitarle la energía de alimentación. La diferencia entre la SRAM y la DRAM estriba en el dispositivo que se utiliza para implementar la celda de memoria. En la SRAM se emplea un flip-flop y en la DRAM un condensador. Un condensador es un componente eléctrico que almacena energía eléctrica, por lo que puede funcionar como celda de memoria, pero tiene el

inconveniente de que se descarga con el tiempo. Para evitar que pierda la información debe reescribirse periódicamente. Esta operación se denomina *refresco*. Por contra, una SRAM al utilizar flip-flop, mantiene la información siempre que se le suministre la energía de alimentación. Por eso se denomina a estas últimas memorias estáticas y a aquellas dinámicas. Estudiaremos primero las RAM típicas y posteriormente se analizarán las ROM.

### 3.2.4. RAM ESTATICA. SRAM.

**SRAM** es la abreviatura de la expresión inglesa *Static Random Access Memory*, es decir memoria estática de acceso aleatorio.

Utilizan como celdas de memoria flip-flops. Un flip-flop es un elemento básico de memoria, que almacena un bit. Se puede escribir con cualquiera de los dos valores binarios (0 o 1 lógico), y mantener ese estado indefinidamente siempre y cuando no se quite la energía de alimentación. Como ya se comentó al estudiar los términos básicos, una memoria que tiene esta característica se denomina *estática*. Las SRAM son memorias de lectura y escritura.

- **Celda de memoria.**

- √ En la figura 3-6 se indica la estructura de la celda de memoria. Las señales CS, RD y WR controlan su funcionamiento.

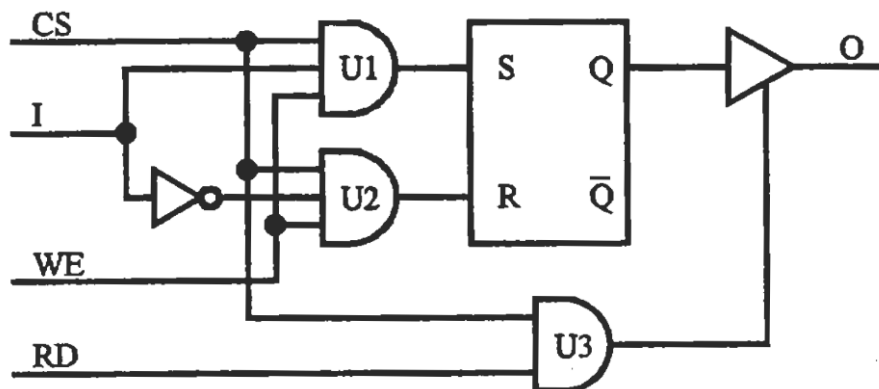


Figura 3-6. Celda básica de una memoria SRAM

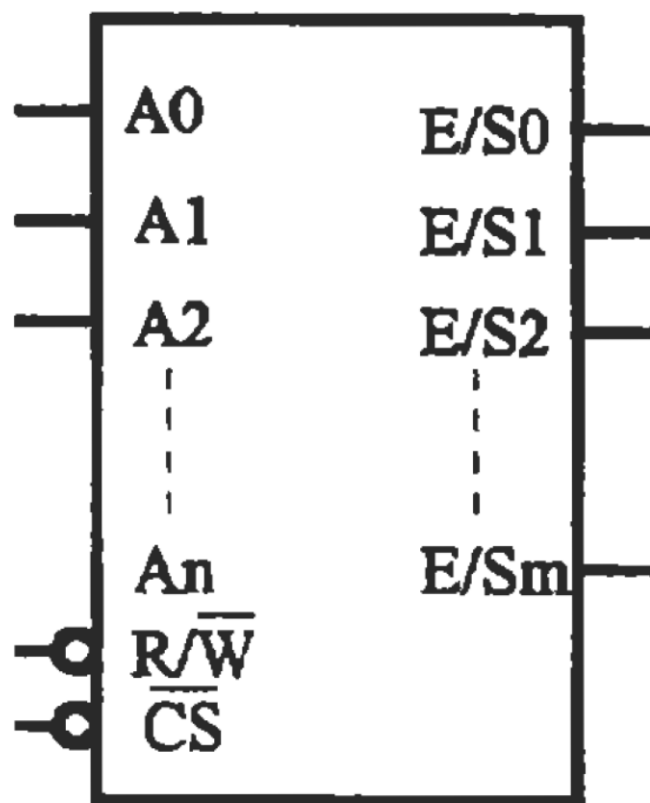
- ❑ CS es la entrada de selección de la celda de memoria. CS=1, celda seleccionada.
- ❑ WE y RD son las señales que determinan si la operación es de escritura o lectura, respectivamente. Son activas a nivel alto, y las generará el bloque de control de la SRAM.
- ❑ I es la entrada de datos y O la salida.
- √ El elemento básico es un flip-flop R-S NOR. El funcionamiento es el siguiente:
  - ❑ Si CS=0, las entradas S y R del flip-flop son 0, por lo que no cambia de estado, y la entrada de habilitación del buffer de salida es 0, poniendo la salida O en alta impedancia. Por tanto, en esta situación la celda no está seleccionada y no se puede realizar ninguna operación con ella.

- Si  $CS=1$ , se selecciona la celda y se realizará una operación de lectura o escritura según el valor de  $WR$  y  $RD$ .
- ⊗ Si  $WR=1$ , la entrada  $S$  del flip-flop es igual a  $I$  y la entrada  $R$  a  $I'$  (complemento de  $I$ ), haciendo que el flip-flop almacene el valor de  $I$ .
- ⊗ Si  $RD=1$ , se habilita el buffer de salida, de manera que en la línea de salida  $O$  se tendrá el valor almacenado en el flip-flop.
- ⊗ El bloque de control de la SRAM se diseña de forma, que nunca estarán activas al mismo tiempo  $RD$  y  $WR$ .

La estructura de una SRAM es la que se expuso para una RAM genérica en el apartado anterior. Las SRAM se fabrican con tecnología bipolar y CMOS. La bipolar es más rápida, pero la CMOS tiene un consumo mucho menor y permite una mayor densidad de integración, y por tanto una mayor capacidad. Esto último se debe a que un transistor bipolar es más complejo y tiene mayor tamaño que un MOSFET. Actualmente se utiliza casi exclusivamente la tecnología CMOS para fabricar las SRAM debido a las ventajas anteriores, y a que se ha reducido los tiempos de acceso por las mejoras tecnológicas.

Las SRAM se utilizan en sistemas de memoria de baja capacidad y tiempo de acceso reducido, como las memorias caché y buffer (intermedia) de un computador. Las SRAM actuales tienen una capacidad máxima de 2Mb (256K x 8).

A continuación vamos a estudiar el cronograma de acceso de las operaciones de lectura y escritura para la SRAM de la figura 3-7.



**Figura 3-7 Bloque de una memoria SRAM genérica**

- **Ciclo de lectura.**

- ✓ En la figura 3-8 se muestra el cronograma de lectura de la SRAM genérica anterior.
- El ciclo de acceso empieza en  $T_0$ . En este instante,
  - ⊗ Las entradas de direcciones tienen la dirección válida de la palabra que se va a leer.
  - ⊗  $R/\bar{W}$  está a nivel alto y permanecerá en él durante todo el ciclo.
  - ⊗  $\bar{CS} = 1$ , por lo que no se selecciona la memoria y las líneas de datos estarán en alta impedancia.
- En el instante  $T_1$  se activa  $\bar{CS}$ , con la que se habilita la memoria. A partir de este instante se habilitan los decodificadores internos de la SRAM.
- En el instante  $T_2$  la memoria pone en el bus de datos el contenido de la palabra direccionada.
  - ⊗ El tiempo entre  $T_0$  y  $T_2$  se denomina **tiempo de acceso  $T_{acc}$** , y equivale al tiempo que transcurre entre la recepción de una dirección en la entrada de la memoria y la disposición del dato válido en la salida.
  - ⊗ El parámetro de temporización  $T_{CO}$ , es el tiempo que tarda la SRAM en pasar su salida de datos del estado de alta impedancia al de dato válido, una vez que se active la señal  $\bar{CS}$ .
- En  $T_3$ ,  $\bar{CS}$  vuelve a alto por lo que la SRAM pone su salida en alta impedancia, después del tiempo  $T_{OD}$ . El dato está estable entre  $T_2$  y  $T_4$ .
- En  $T_5$  se cambia de nuevo la dirección para leer otra palabra. El tiempo entre  $T_0$  y  $T_5$  se denomina **tiempo de ciclo de lectura  $T_{rc}$** , y es que debe transcurrir entre dos accesos consecutivos de lectura.

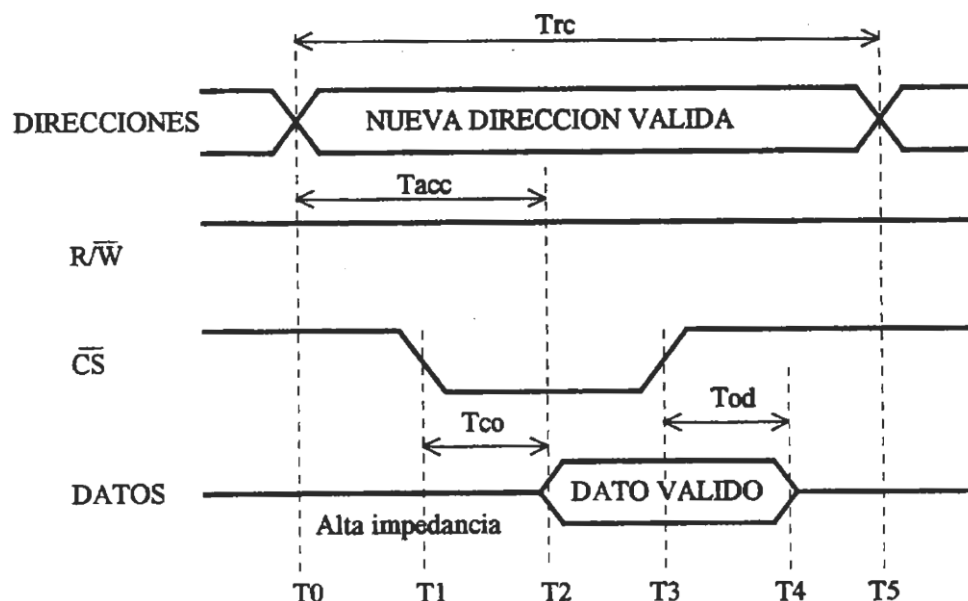


Figura 3-8. Cronograma de un ciclo de lectura de una SRAM

- **Ciclo de escritura.**

- ✓ En la figura 3-9 se muestra el cronograma de escritura de la SRAM genérica anterior.
- ❑ En ciclo comienza en T0 poniendo una dirección válida en las entradas de direcciones de la SRAM.
- ❑ Después de un tiempo  $T_{as}$ , denominado **tiempo de establecimiento de direcciones**, se activan tanto CS como R/W, con lo que se habilita el funcionamiento de los decodificadores de la SRAM. Las señales CS y R/W se mantienen a nivel bajo durante un intervalo de tiempo  $T_w$ , denominado **anchura de pulso de escritura**.
- ❑ En T1, durante el pulso de escritura, se debe poner en las entradas de datos el dato válido, que se va a escribir en la palabra de la SRAM. Los datos deben estar estables un tiempo  $T_{ds}$  antes de y un tiempo  $T_{dh}$  después de la desactivación de las señales R/W y CS.
- ⊗ El tiempo  $T_{ds}$  se denomina **tiempo de establecimiento del dato**.
- ⊗ El tiempo  $T_{dh}$  se denomina **tiempo de mantenimiento del dato**.
- ❑ Las direcciones deben mantenerse estables durante el tiempo  $T_{ah}$ , después de T2.
- ❑ Si no se satisface alguno de estos tiempos, la operación de escritura no se realizará correctamente.
- ❑  $T_{wc}$  es el **tiempo del ciclo de escritura**. Equivale al intervalo de tiempo entre T0 y T4, es decir durante el que deben estar estables las direcciones para realizar correctamente el ciclo de escritura. Por tanto, es el que define la duración del ciclo, y el que debe haber como mínimo entre dos operaciones consecutivas.

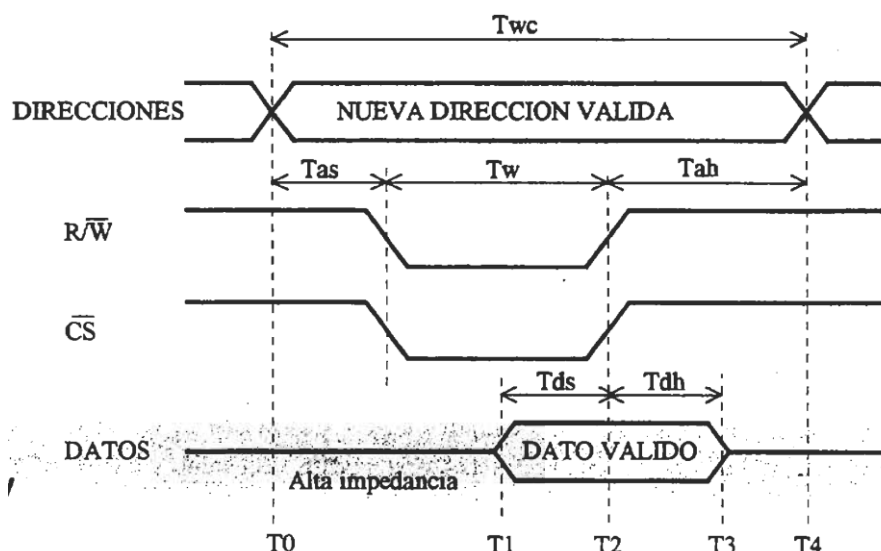


Figura 3-9. Cronograma de un ciclo de escritura de una SRAM

- ✓ La velocidad de la memoria está determinada por los tiempos  $T_{RC}$  y  $T_{WC}$ . Por ejemplo, una SRAM de  $T_{RC} = 50$  ns, quiere decir que puede leer una palabra cada 50 ns, es decir,  $1/50\text{ns} = 20$  millones de palabras/s.

### 3.2.5. RAM DINAMICAS. DRAM.

Las SRAM analizadas anteriormente almacenan la información en flip-flops. Sin embargo, las DRAM lo hacen en pequeños condensadores MOS. Un condensador es un componente que almacena energía eléctrica. Como ya se sabe en los sistemas digitales los valores lógicos se representan mediante niveles de tensión, de forma que un 1 lógico equivale a una tensión mayor de un determinado valor, y un 0 lógico a una tensión menor. Un condensador acumula energía, de forma que si entre sus extremos se le aplica una tensión continua (5V), este se carga a ese nivel (5V) y la mantiene durante un tiempo determinado. Es decir, un condensador almacena el nivel de tensión que hay entre sus extremos. Por tanto, un condensador almacena información binaria ya que tiene dos estados:

- Condensador cargado. Almacena un 1 lógico.
- Condensador descargado. Almacena un 0 lógico.

Por el modo de funcionamiento de un condensador, si entre sus extremos hay 5 V, se puede descargar para almacenar un 0, estableciendo una diferencia de potencial de 0V entre ellos. Para determinar el contenido del condensador se utiliza un circuito comparador, que indica si la diferencia de potencial entre los extremos del condensador es mayor o menor que un determinado umbral. Si es mayor almacena un 1 lógico y si es menor un 0 lógico.

Los condensadores almacenan carga eléctrica, pero no la mantienen por tiempo indefinido, ya que siempre existe un camino de descarga. Aunque la intensidad de descarga es muy pequeña, el condensador la perderá al cabo de cierto tiempo. Por tanto, las DRAM requieren la recarga periódica de las celdas de memoria; este proceso se denomina *refresco*. En general, cada celda debe refrescarse cada cierto tiempo para que no se pierda la información. Este tiempo varía entre 2 y 10 ms según la DRAM, y para que sea lo mayor posible debe utilizarse la tecnología CMOS.

Debido a que se debe rescribir periódicamente su contenido se les denomina memorias *dinámicas*.

El refresco de la celdas de memoria es una desventaja de la DRAM frente a las SRAM, ya que el sistemas de control se hace más complejo.

Por contra, las DRAM tienen la ventaja de la densidad de almacenamiento. Un condensador ocupa aproximadamente 4 veces menos espacio que un flip-flop, por lo que permite una mayor densidad de almacenamiento. Generalmente el coste por bit de almacenamiento en las DRAM se encuentra entre la quinta y la cuarta parte del coste de la SRAM.

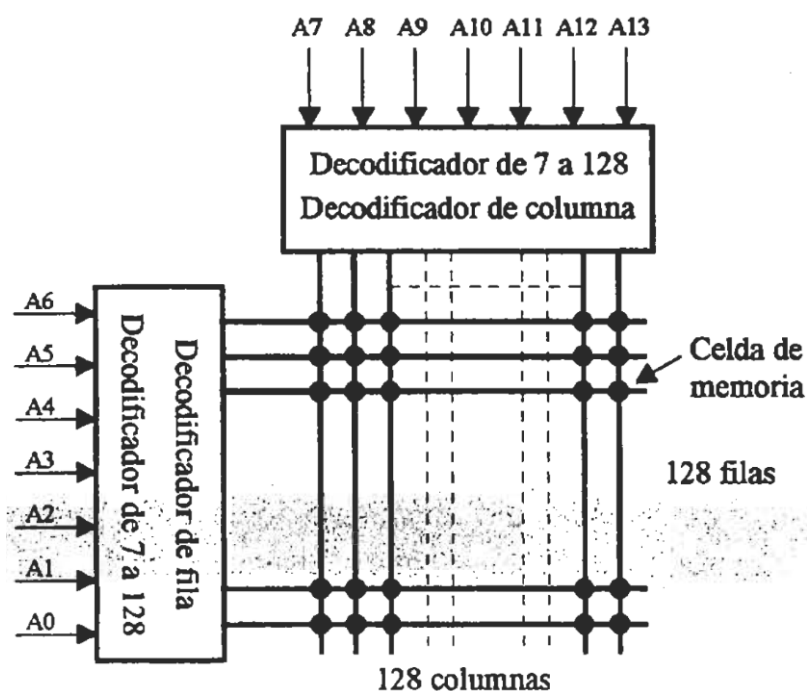
Además, tienen la ventaja de que consumen entre la mitad y un sexto de potencia que una SRAM, por lo que se necesitará una fuente de alimentación de menor potencia y más barata.

La capacidad máxima de las DRAM actuales es del orden de los 64Mb, frente a los 2 Mb de las SRAM. Las memorias SRAM y DRAM son las más utilizadas. La aplicación de una y otra depende fundamentalmente de la relación velocidad/capacidad que se desee.

- En aplicaciones en las que se valora la rapidez y la sencillez, en vez de la capacidad y el consumo, se utilizarán las SRAM, por ser más rápidas y no necesitar refresco. De esta manera, el diseño será más sencillo. En general se puede considerar que si la capacidad es menor que 512KB, y si la aplicación es de alta velocidad, como en los osciloscopios de almacenamiento digital y analizadores, se utilizan SRAM. En los computadores se utilizan en la memoria cache o como memorias intermedias, es decir en donde es muy importante la rapidez.
- Si la memoria del sistema tiene que cumplir las condiciones de una elevada capacidad y un bajo consumo, la mejor opción es la DRAM. En este caso aunque el circuito de control de refresco aumente la complejidad del diseño, y por tanto el número de circuitos integrados, no obstante la implementación final del sistema será más reducida al necesitarse menos circuitos de memoria. Las DRAM se utilizan generalmente para implementar la memoria principal de los computadores, por las características comentadas anteriormente.

### 3.2.5.1. MATRIZ DE CELDAS Y FUNCIONAMIENTO DE LA CELDA DE MEMORIA.

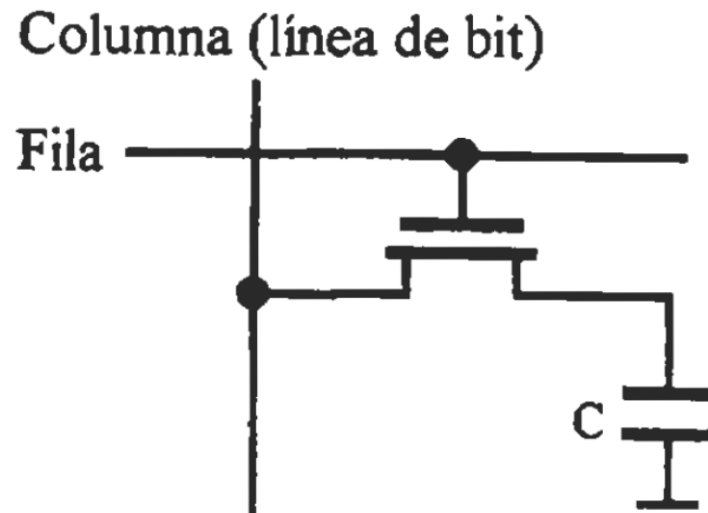
La arquitectura interna de una DRAM utiliza la estructura bidimensional. En la figura 3-10 se muestra la arquitectura de una DRAM de 16K x 1. Se parte de una matriz de celdas de memoria de 1 bit. Para direccionar 16K palabras se necesitan 14 bits de direcciones ( $16K = 2^4 \cdot 2^{10} = 2^{14}$ ), por lo que la matriz tendrá un tamaño de 128 x 128 ( $2^7$ ). Cada celda ocupa una posición única en la matriz, que estará determinada por sus direcciones de fila y columna.



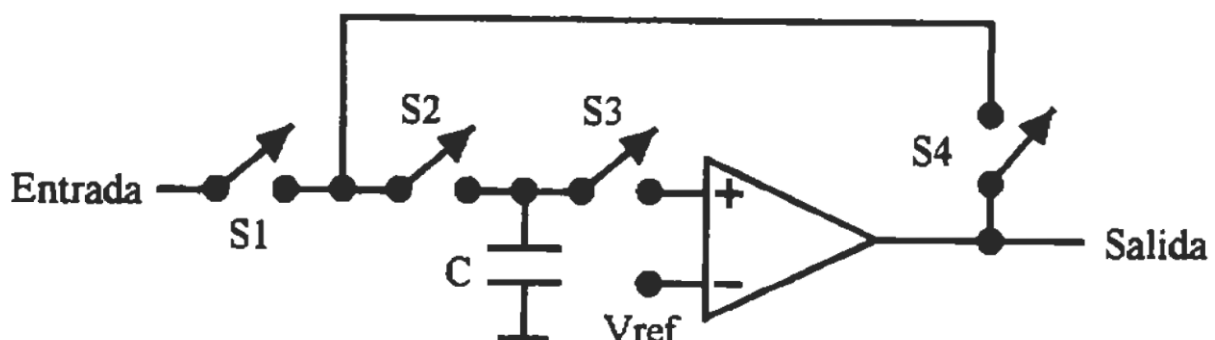
**Figura 3-10. Estructura de una DRAM**

Los bits A0 a A6 seleccionan la fila, mientras que los A7 a A13 seleccionan la columna. Se ha planteado la estructura de una DRAM de anchura de palabra de 1 bit, pero también existen DRAM con anchura de palabra de 4 bits. En este caso la estructura es la misma, salvo que cada posición dentro de la matriz contiene 4 celdas y cada dirección selecciona un grupo de celdas para la operación de lectura.

La figura 3-11 muestra la estructura de la celda de memoria dinámica, formado por un único transistor MOS y un condensador.

**Figura 3-11. Celda de memoria**

La figura 3-12 muestra de una manera simbólica la estructura de la celda de memoria dinámica, en la que se utilizan conmutadores para conectar la celda a la entrada o salida de datos. Estos conmutadores se implementan con transistores MOSFET y se controlan por los decodificadores de fila y columna y la señal R/W, que indica el tipo de operación. Como ya se comentó, la celda de memoria está constituida por el condensador C. Vamos a analizar el funcionamiento de la celda básica en las operaciones de lectura y escritura. Suponemos que R/W a 0 indica escritura y a 1 lectura.

**Figura 3-12. Celda de memoria**

- **Escritura.** El circuito de control a partir de las señales de los decodificadores y R/W, cierra los conmutadores S1 y S2 y mantiene abiertos S3 y S4, con lo que el condensador se conecta al dato de entrada. Si el dato de entrada es 1 el condensador se carga, y si es un 0 el condensador se descarga. Posteriormente los



conmutadores se abren, quedando aislado el condensador del resto del circuito. Desde un punto de vista ideal, C debe mantener la carga indefinidamente, pero en realidad hay un camino de descarga a través de los conmutadores, por lo que C va perdiéndola.

- **Lectura.** El circuito de control cierra S2, S3 y S4 y mantiene S1 abierto, conectándose el condensador C al amplificador de detección. Este compara el nivel de tensión con una tensión de referencia, y si la del condensador es mayor pondrá un 1 lógico (5V) en la salida, y si es menor un 0 lógico (0V). Si se observa la figura, se verá que la salida se conecta a C (S2 y S4 cerrados) en una operación de lectura. Con este esquema, el condensador refresca su contenido automáticamente cargándolo o descargándolo. Por tanto, al hacer una operación de lectura de una celda de memoria se refresca automáticamente. De esto se deduce que una operación de refresco es básicamente una operación de lectura.

### 3.2.5.2. ESTRUCTURA INTERNA DE UNA DRAM.

Como las DRAM actuales tienen una elevada capacidad se necesitará un número grande de patillas de direcciones para poder seleccionar cada una de las palabras. Por ejemplo, una DRAM actual puede tener una capacidad de 64 Mb y si suponemos que su palabra es de 1 bit, se necesitarán 26 líneas de direcciones ( $2^6 \cdot 2^{20}$ ).

Para reducir el número de patillas se emplea el multiplexado de direcciones, de forma que se utiliza un terminal para dos bits diferentes de direcciones, reduciéndose a la mitad el número de patillas de direcciones. En el ejemplo anterior se consigue una disminución bastante importante en el número de patillas (13).

En la figura 3-13 se muestra la estructura interna de la DRAM de 16K x 1. Contiene una matriz de celdas organizadas en 128 filas y 128 columnas y dos registros. Como se multiplexa una dirección de fila con otro de columna se necesita un registro para almacenar la dirección de fila y otro para la de columna. La DRAM tendrá las siguientes patillas:

- Una señal de **control R/W** que indique el tipo de operación. Para una operación de lectura R/W debe ponerse a 1, y para una operación de escritura a 0.
- **7 entradas de direcciones A0/A7 .. A6/A13**, en las que se multiplexan en el tiempo dos bits diferentes. Por ejemplo, la patilla A0/A7 corresponde al bit A0 en un instante y al A7 en otro posterior.
- **Entradas de selección de pastilla.** En este caso se utilizan dos señales, que además de hacer la función anterior sincronizan las direcciones de fila y columna en los registros. La **señal de selección de dirección de fila (RAS)** controla la entrada de reloj del registro de direcciones de fila, mientras que la **señal de selección de dirección de columna (CAS)** lo hace con el registro de direcciones de columna.
- Una patilla para la **entrada del dato** y otra para la **salida**.

La dirección se envía a la DRAM en dos pasos mediante las señales RAS y CAS. En la figura 3-13 se muestra el diagrama de temporización de estas señales.

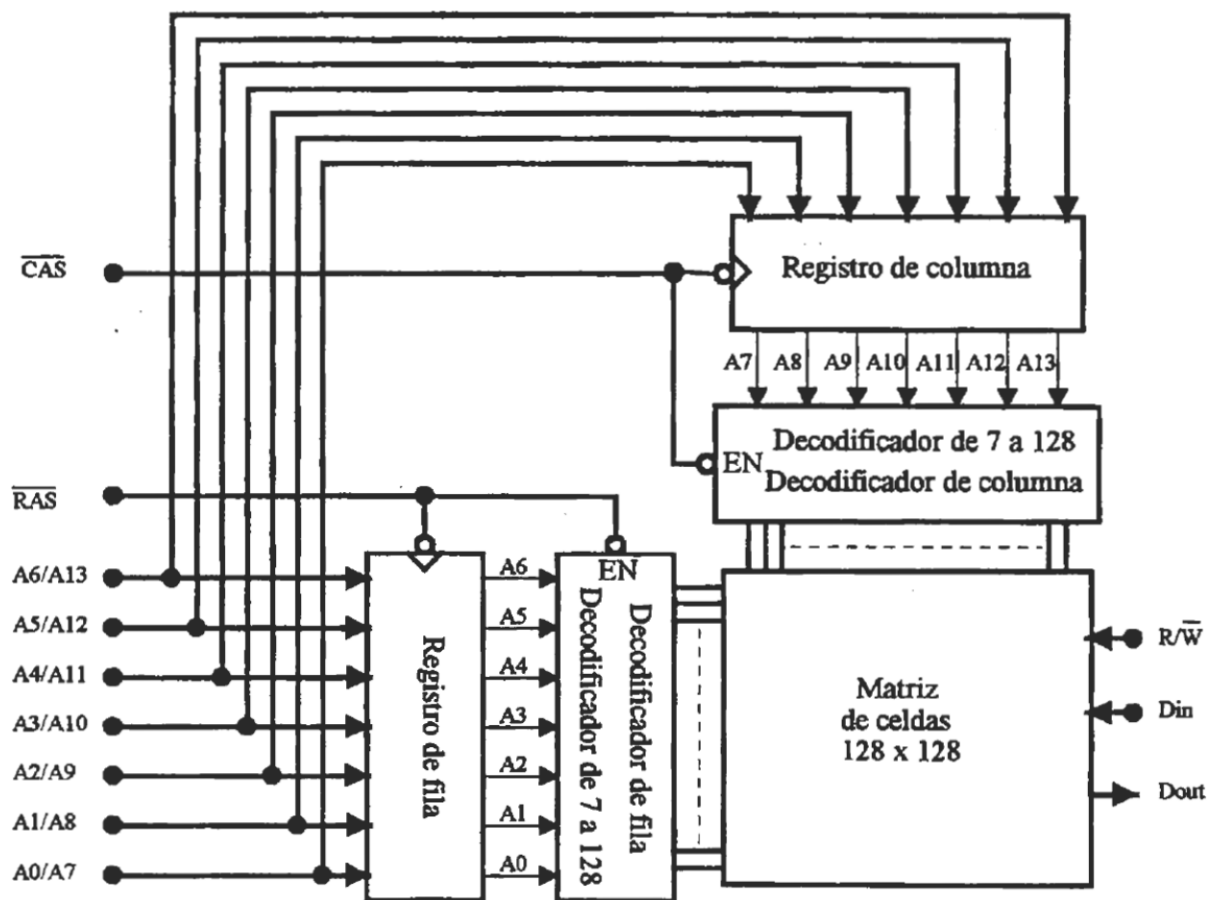
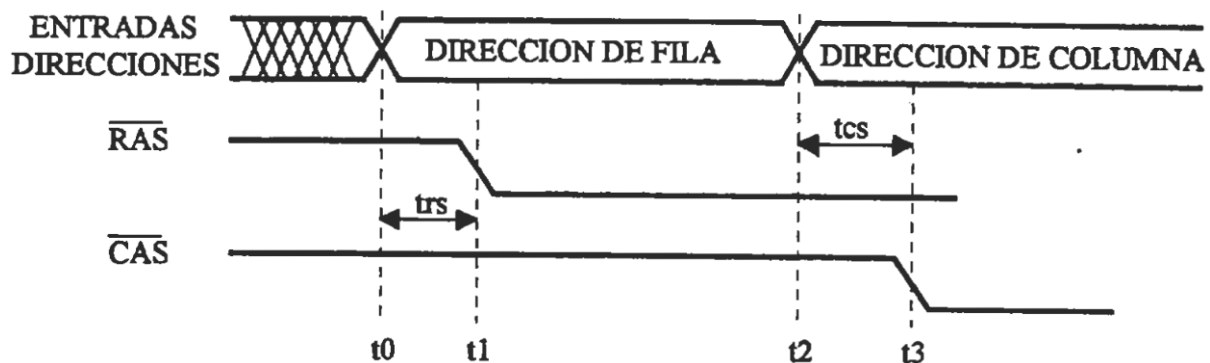


Figura 3-13. Arquitectura de una DRAM de 16Kx1

Figura 3-14. Cronograma de las direcciones,  $\overline{CAS}$  y  $\overline{RAS}$ 

- En el instante inicial RAS y CAS están a nivel alto, por lo que la memoria no se selecciona.
- En t0 las patillas de direcciones contienen la dirección de fila (A0 .. A6).
- Se espera el tiempo de establecimiento  $t_{rs}$ , que necesita el registro de fila. Después de ese tiempo en t1, se pasa a nivel bajo RAS. Con el flanco de bajada de RAS se almacena la dirección de fila (A0 .. A6) en el registro de fila, y por tanto, se aplica ésta al decodificador de fila. El nivel bajo en RAS habilita al decodificador, que selecciona la correspondiente fila de la matriz.

- En t2 se aplica a las entradas de direcciones la dirección de columna (A7 .. A13).
- En t3 la entrada CAS pasa a nivel bajo. Se almacena la dirección de columna en el registro de dirección de columna, y se habilita el decodificador, con lo que se selecciona la correspondiente columna de la matriz.
- Una vez que los dos decodificadores tienen sus salidas al valor adecuado, estará seleccionada la celda de memoria correspondiente a la dirección de 14 bits (A0 .. A13), y se podrá realizar una operación de lectura o escritura, según la entrada R/W.

La DRAM no tiene una señal de selección de pastilla (CS) como la SRAM. En este caso, la DRAM se selecciona cuando CAS y RAS están a nivel bajo, las cuales habilitan los dos decodificadores de forma que se seleccionará una celda de memoria. En la DRAM de 16K x 1 se ahorran 6 patillas, ya que la SRAM tendrá 15 pines (14 direcciones + CS) y una DRAM 9 (7 direcciones + RAS + CAS).

### 3.2.5.3. CONEXIÓN DE UNA DRAM.

Para comprender las diferencias en la implementación de un sistema de memoria con SRAM y DRAM, vamos a analizar cómo sería para un sistema de 16K palabras.

- Para direccionar 16K palabras se necesitan 14 señales de direcciones, A0-A13.
- **SRAM.** Si suponemos, que se conecta a una Unidad Central de Proceso (UCP), para el caso de una SRAM de 16K palabras la conexión es muy simple, ya que solamente tienen que conectarse entre sí las líneas de direcciones correspondientes. Ver figura 3-15.

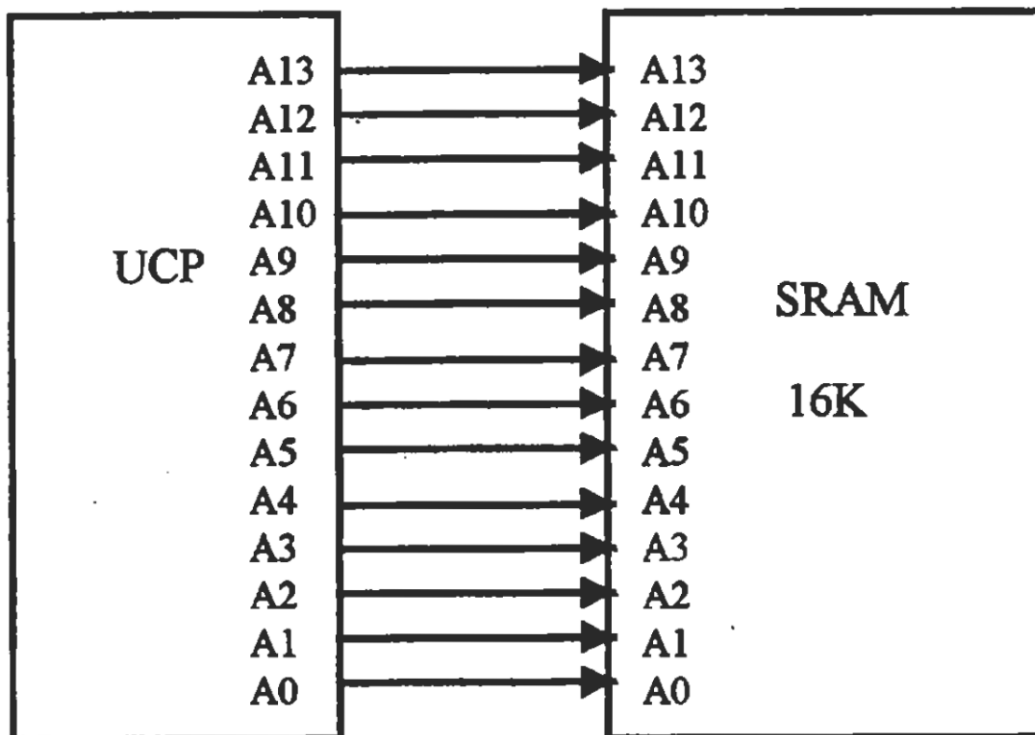


Figura 3-15. Conexión de una SRAM de 16K palabras a una UCP

- **DRAM.**

- √ Como la DRAM tiene 7 líneas de direcciones y la UCP 14, hay que usar un multiplexor para transmitirlos en dos grupos de 7. Como ya se ha estudiado primero se envía la dirección de fila (A0-A6) y posteriormente la de columna (A7-A13).
- √ Como se indica en la figura 3-16, el proceso anterior se realiza mediante 7 multiplexores de 2 a 1. La entrada MUX controla si se envía a la DRAM la dirección de fila o columna:
  - ☐ MUX = 0. Dirección de fila.
  - ☐ MUX = 1. Dirección de columna.

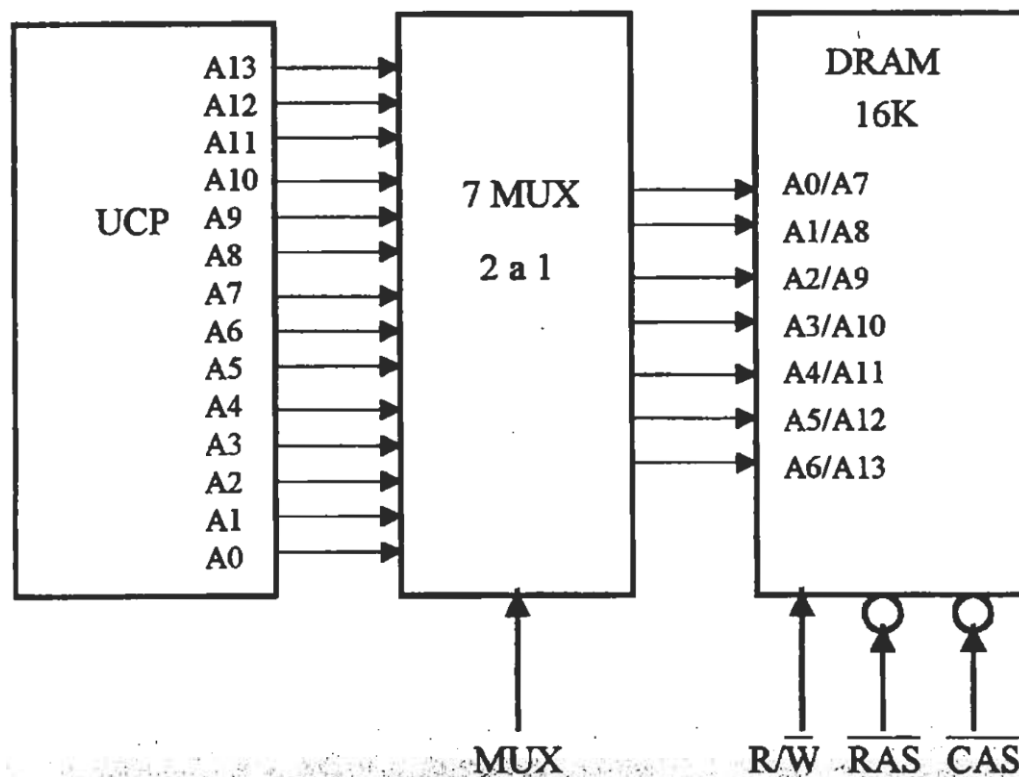


Figura 3-16. Conexión de una DRAM de 16K palabras a una UCP

- √ Las señales MUX, CAS y RAS deben estar sincronizadas entre sí. El cronograma se representa en la figura 3-17.
  - ☐ Inicialmente MUX debe estar a nivel bajo, de forma que se envíe la dirección de fila (A0-A6) a la DRAM.
  - ☐ Posteriormente se activa RAS. Con el flanco de bajada se almacena la dirección de fila en el registro de filas.
  - ☐ Se cambia a nivel alto MUX para transmitir la dirección de columna (A7-A13) a la DRAM.
  - ☐ Posteriormente se activa CAS. Con el flanco de bajada se almacena la dirección de columna en el registro de columnas.

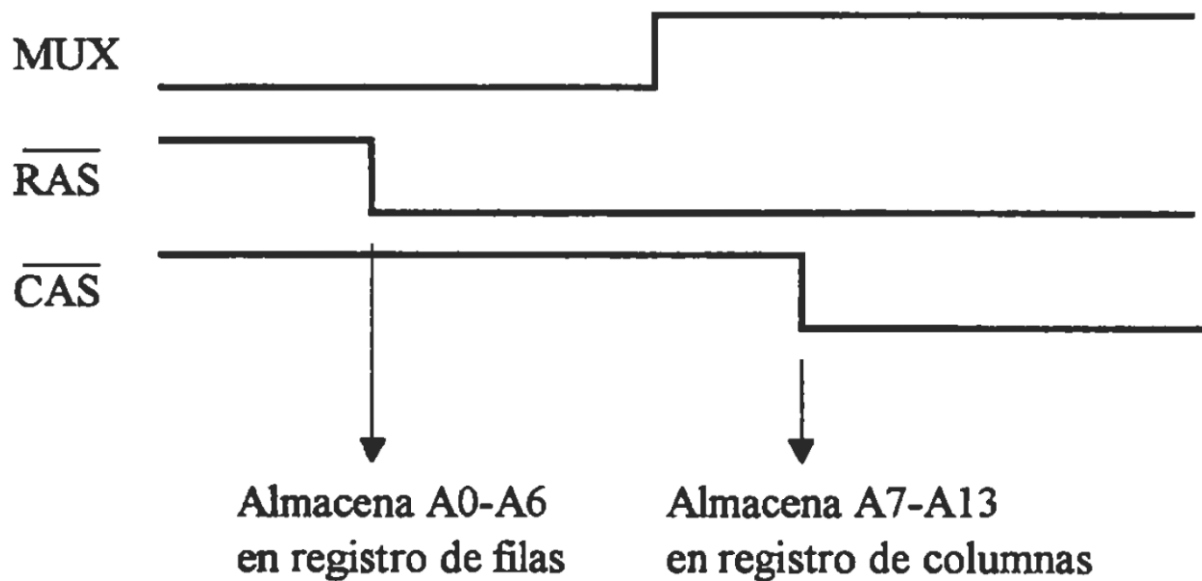


Figura 3-17. Cronograma de la multiplexación de direcciones

### 3.2.5.4. CRONOGRAMA DE LOS CICLOS DE LECTURA/ ESCRITURA.

El cronograma de los ciclos de lectura y escritura de una DRAM es más complejo que el de una SRAM. Se analizarán los ciclos de acceso partiendo de la configuración del apartado anterior.

- **Ciclo de lectura de una DRAM.**

- ✓ Durante todo el ciclo la señal R/W debe permanecer a nivel alto. En la figura 3-18 se describen los pasos necesarios para hacer un acceso de lectura a una DRAM.
  - ☐ t0. MUX cambia a nivel bajo para enviar la dirección de fila (A0-A6) a las entradas de direcciones de la DRAM.
  - ☐ t1. RAS cambia a nivel bajo para almacenar en la DRAM la dirección de fila.
  - ☐ t2. MUX se cambia a nivel alto para poner la dirección de columna (A7-A13) en las entradas de direcciones de la DRAM.
  - ☐ t3. CAS cambia a nivel bajo para almacenar en la DRAM la dirección de la columna.
  - ☐ t4. Una vez almacenadas la dirección de fila y columna, al estar activas RAS y CAS los decodificadores seleccionan la celda de memoria, y la DRAM pone el dato válido en la línea de salida SAL. DATO.
  - ☐ t5. Se desactivan CAS y RAS haciendo que SAL. DATO. pase a alta impedancia.
- ✓ Hay dos tiempos de acceso, uno respecto a RAS y otro respecto a CAS.

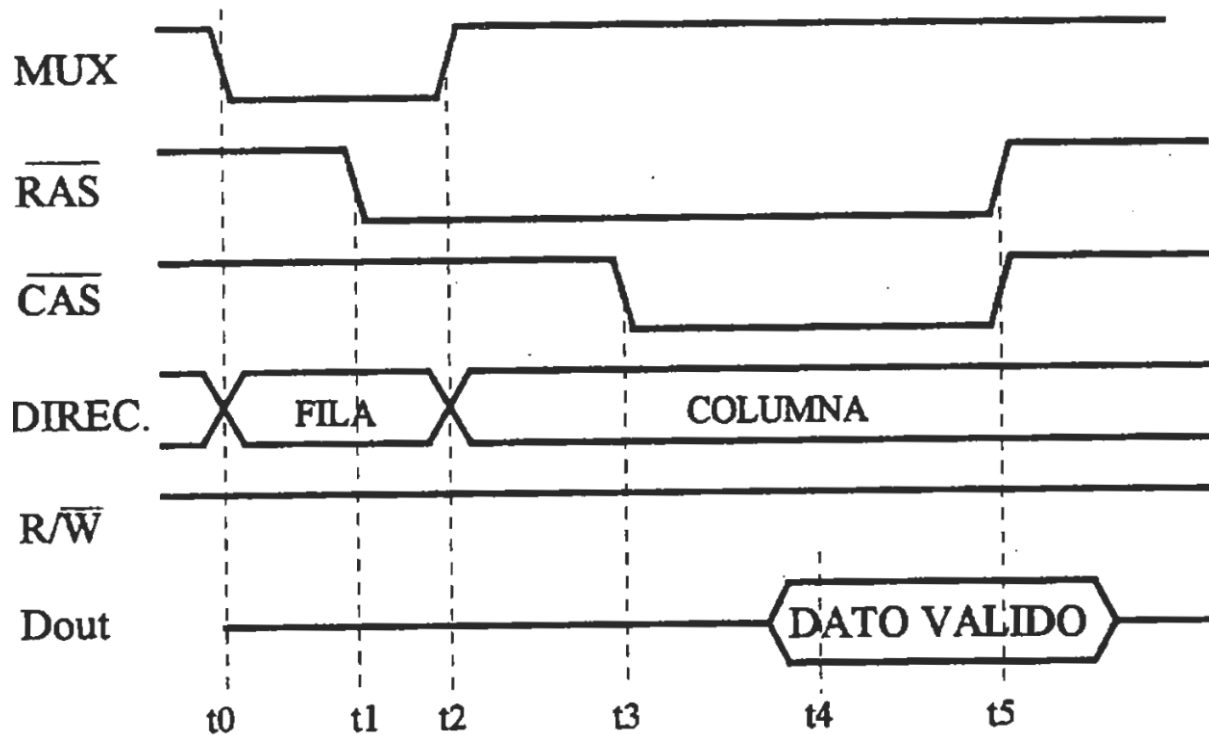


Figura 3-18. Cronograma del ciclo de lectura de una DRAM

- **Ciclo de escritura de una DRAM.**

- ✓ En la figura 3-19 se describen los pasos necesarios para realizar un acceso de escritura a una DRAM.

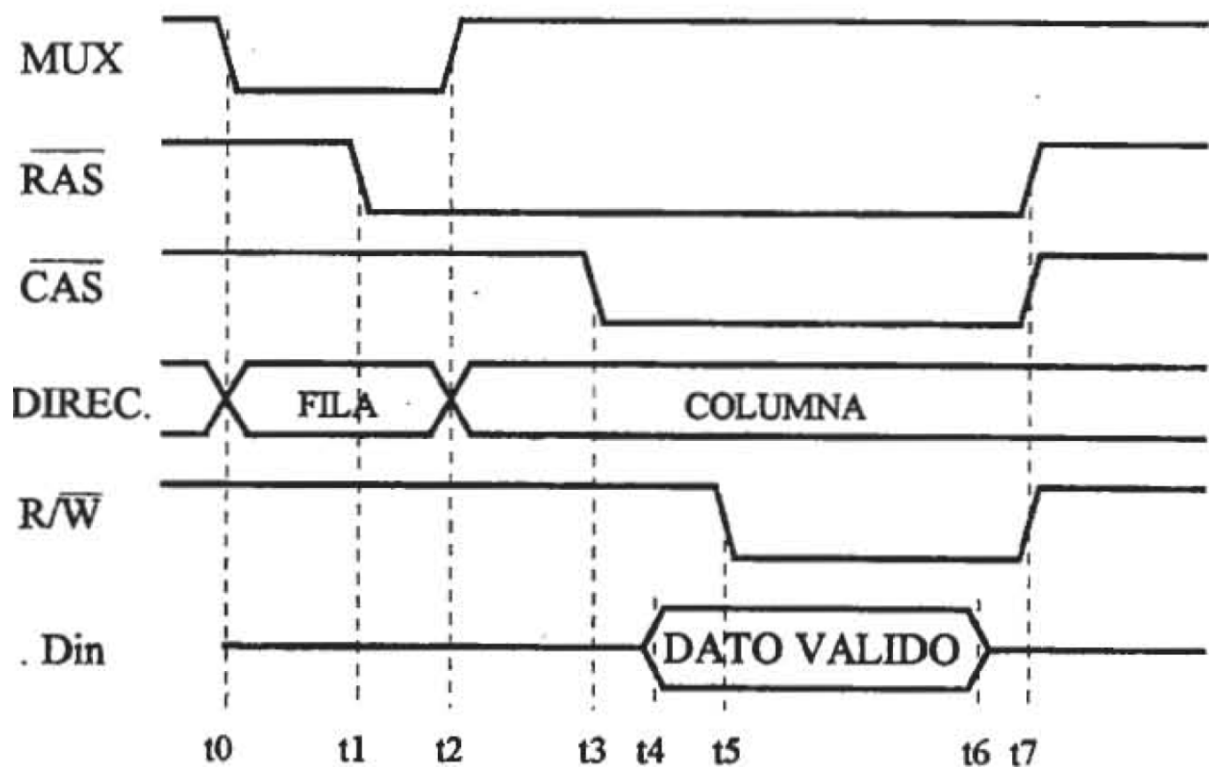


Figura 3-19. Cronograma del ciclo de escritura de una DRAM

- ☐ Los pasos  $t_0$  a  $t_3$  son los mismos que para un ciclo de lectura.
- ☐  $t_4$ . Una vez almacenadas la dirección de fila y columna, al estar activas RAS y CAS los decodificadores seleccionan la celda de memoria. El dato que se va a escribir en la memoria se pone en la línea ENT. DATO.
- ☐  $t_5$ . Después del tiempo de establecimiento del dato, R/W se pone a nivel bajo para escribir éste en la celda seleccionada.
- ☐  $t_6$ . Después del tiempo de mantenimiento del dato, que necesita la celda, se pone en alta impedancia la línea de entrada de datos. No obstante, el dato puede mantenerse hasta el final del ciclo.
- ☐  $t_7$ . El ciclo termina con la desactivación de CAS, RAS y R/W.

### 3.2.6. RAM NO VOLATIL. *NON VOLATIL RAM*.

Las memorias RAM estudiadas anteriormente (SRAM y DRAM) son memorias volátiles, es decir, que pierden su contenido al interrumpirle el voltaje de alimentación.

Si se necesita una memoria RAM no volátil, que mantenga su contenido aunque se le quite la tensión de alimentación, hay dos soluciones:

- Utilizar memorias que puedan alimentarse con baterías de *backup* o respaldo, cada vez que se le interrumpe la alimentación.
  - ✓ Esto obliga a que las memorias sean CMOS para que tengan un consumo muy reducido y las baterías no se descarguen demasiado pronto. Existen versiones de RAM de muy bajo consumo.
  - ✓ El sistema de memoria dispone de un circuito de control, que supervisa la tensión de alimentación.
    - ☐ En un funcionamiento normal, las memorias se alimentan de la tensión continua del sistema.
    - ☐ Al desconectar la tensión de alimentación del sistema, el circuito de control conecta la batería de backup a las patillas de alimentación de las memorias.
- Utilizar memorias realmente no volátiles, que incorporan la circuitería necesaria para no perder la información cuando se le interrumpa la alimentación. La memoria de este tipo más utilizada es la NVRAM con pila de litio interna.

#### 3.2.6.1. NVRAM CON PILA DE LITIO INTERNA.

- Están formadas por una SRAM a la que se le añade una pila de litio y un circuito de detección de fallo de alimentación.

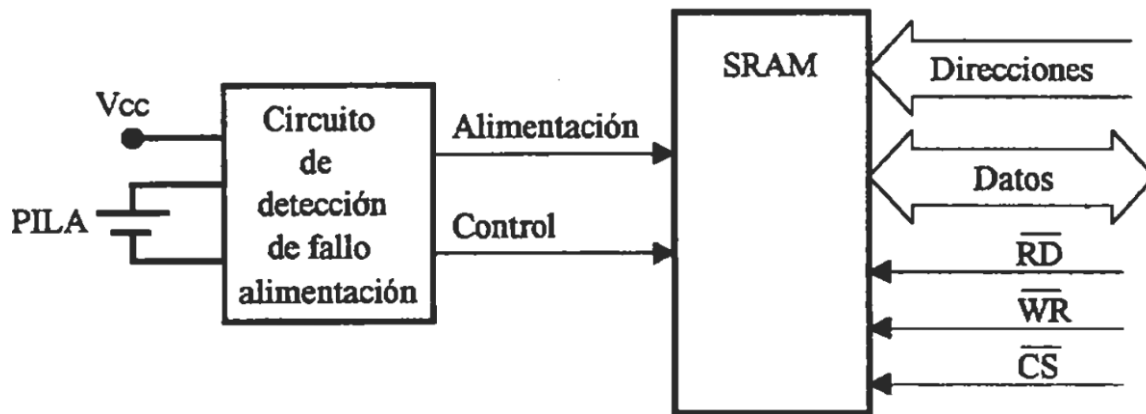


Figura 3-20. Estructura interna de una NVRAM con pila de litio

- El circuito de detección de fallo de alimentación está comprobando continuamente el nivel de la tensión de alimentación, y si éste cae por debajo de un cierto nivel realiza las operaciones necesarias para que no se pierdan los datos almacenados:
  - ✓ Inhibe la operación de escritura.
  - ✓ Si el nivel de la tensión de alimentación desciende por debajo de un valor, que generalmente es 3'3V, conmuta la alimentación a la pila de litio interna.
  - ✓ Cuando la tensión de alimentación se restablece se realizan las operaciones contrarias.

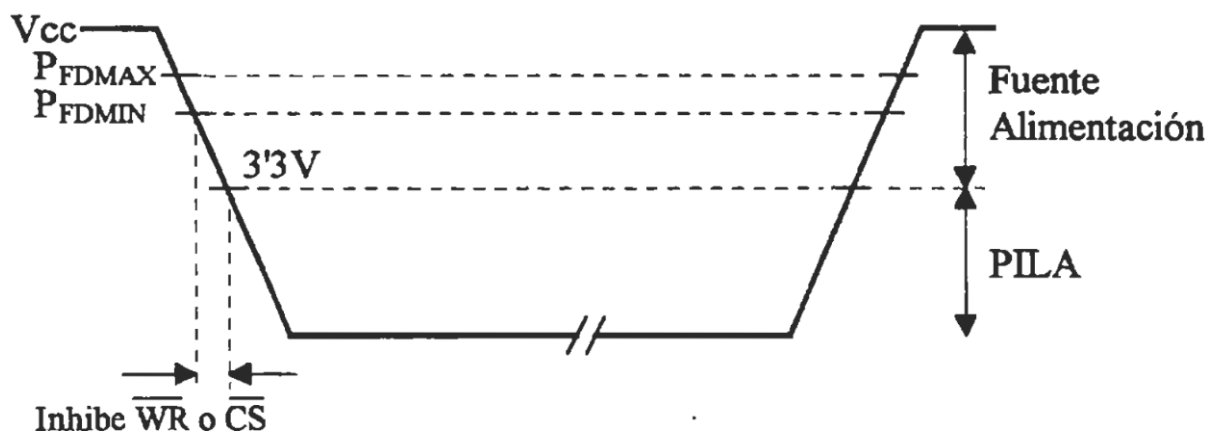


Figura 3-21. Temporización de la NVRAM

### 3.2.7. ASOCIACIÓN DE MEMORIAS DE ACCESO ALEATORIO.

No siempre es posible encontrar en el mercado memorias RAM de la anchura de palabra o capacidad deseada. Existen ya unos tamaños de anchura de palabra y capacidad estandarizados, y a partir de estos debemos construir la memoria deseada.

#### 3.2.7.1. ASOCIACIÓN DE SRAM.

Veamos con algunos ejemplos como podemos asociar memorias SRAM.

- *Aumento en tamaño de palabra.*

- ✓ La figura siguiente muestra cómo se asocian dos memorias de 2Kx8 para conseguir una de 2Kx16.



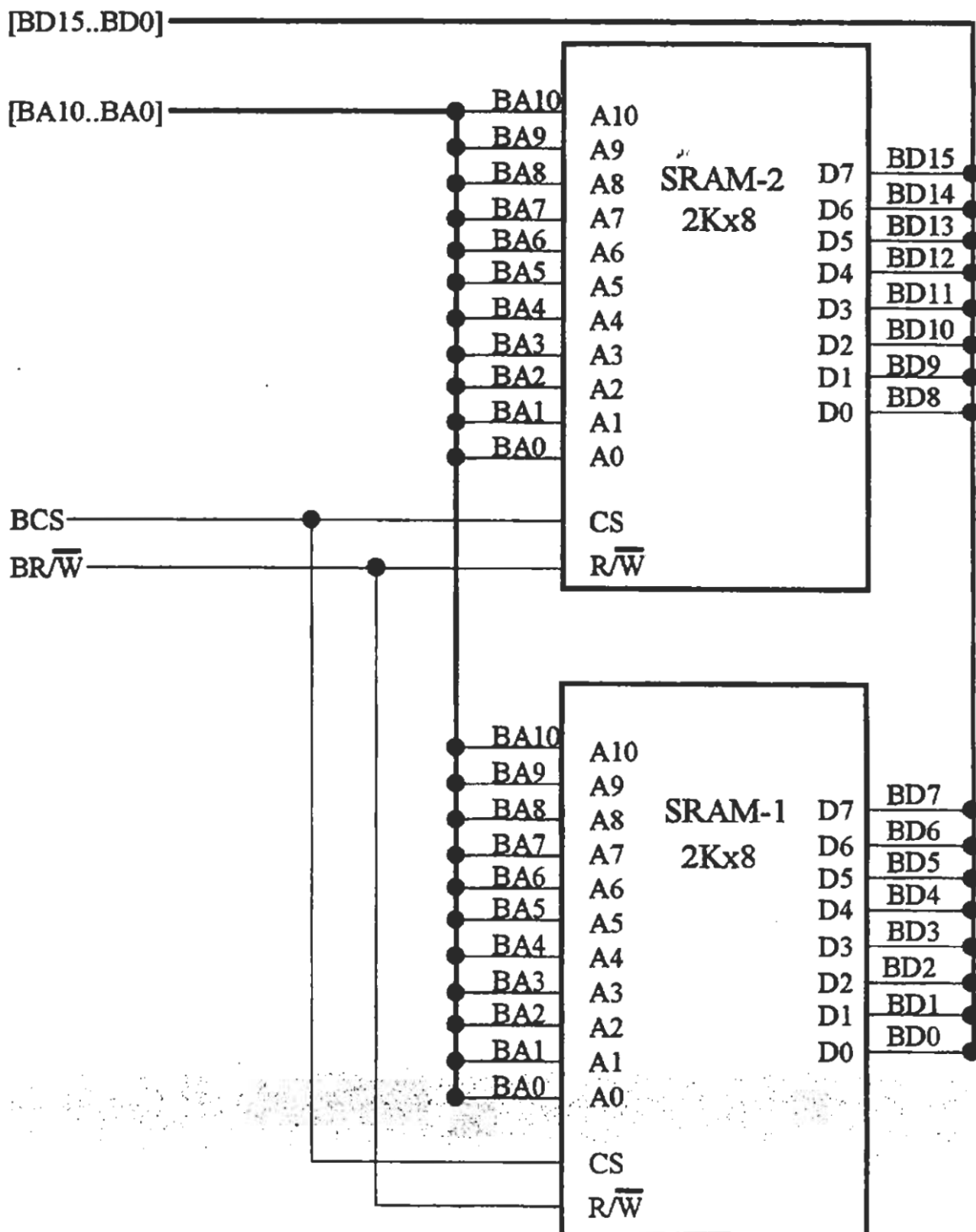


Figura 3-22. Asociación de dos SRAM de 2Kx8 para conseguir 2Kx16

- ❑ La memoria 1 queda conectada siempre a los 8 bits más bajos del bus de datos, y la memoria 2 a los 8 bits más altos.

BD15..BD8	BD7..BD0
D7 .. D0	D7 .. D0
Memoria 2	Memoria 1

Tabla3-2

- ❑ Los *chip select* de ambas memorias se han unido para que se activen a la vez, ya que se requieren 16 bits.

- La capacidad total de la memoria en número de bits se ha aumentado al doble sin embargo la capacidad en número de palabras no ha variado al unir las dos memorias.

- ***Aumento en capacidad o número de palabras.***

- √ La figura siguiente muestra cómo se asocian dos memorias de 2Kx8 para conseguir una de 4Kx8.

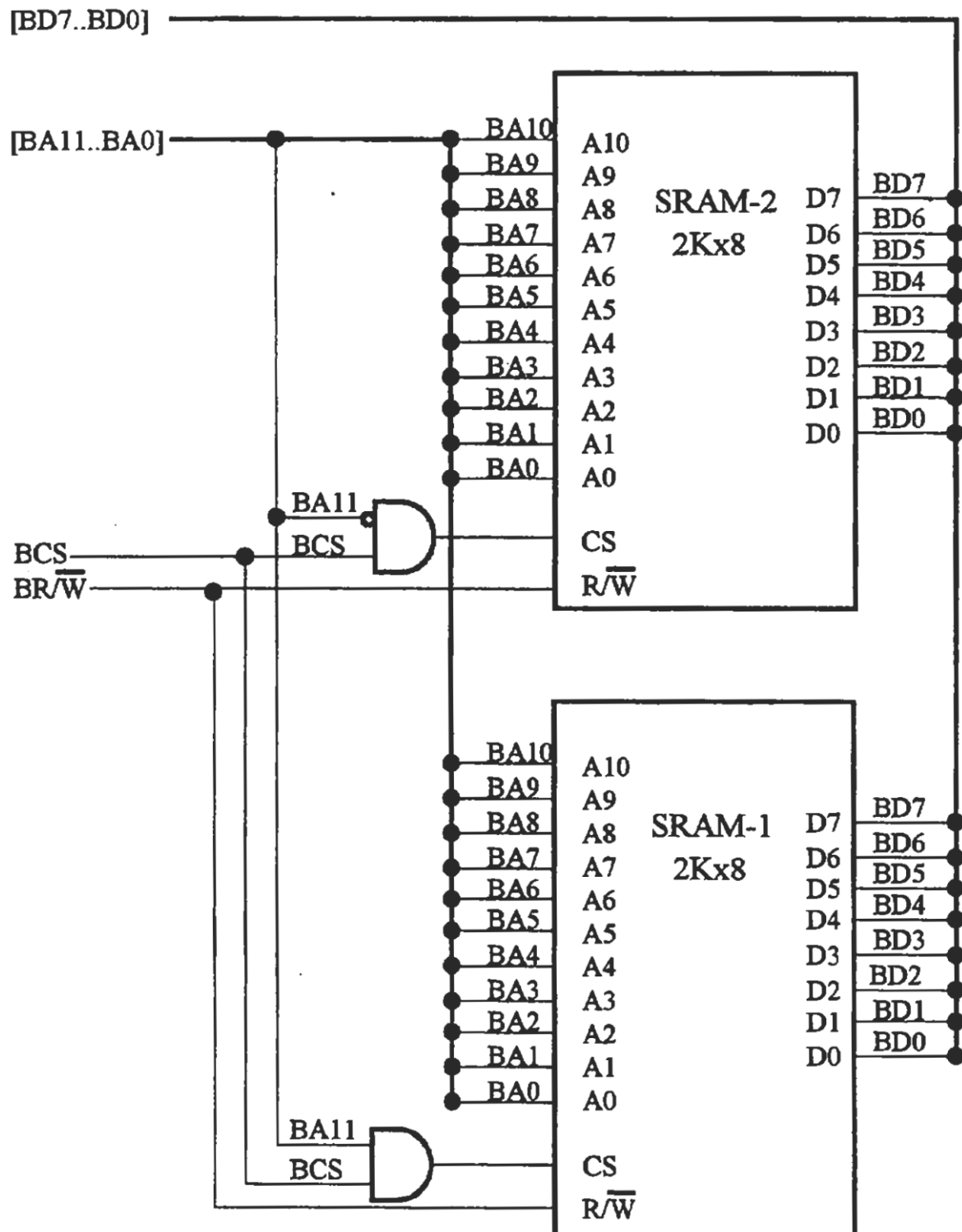


Figura 3-23. Asociación de dos SRAM de 2Kx8 para conseguir 4Kx8

- ❑ Las líneas de datos de ambas memorias se unen ya que el ancho de la palabra es de 8 bits, y nunca deberán activarse los dos chips selects a la vez.
- ❑ El chip select de cada memoria se activará en función de la dirección BA11, de forma que para BA11=0 se activará la memoria 2 y para BA11=1 se activará la memoria 1.

	4K	
BA11=1	Memoria 1	2K
BA11=0	Memoria 2	2K

Tabla 3-3

### 3.3. MEMORIAS DE SOLO LECTURA (ROM).

Una memoria de sólo lectura [*Read Only Memory* (ROM)] es una memoria en la que podemos leer, pero en la que no podemos escribir. Los contenidos de la memoria son fijos e inalterables, estableciéndose en el momento de la fabricación.

#### 3.3.1. ESTRUCTURA DE UNA ROM

Como una ROM no tiene operación de escritura, la información no tiene que cambiarse, por lo que no debe almacenarse la información en flip-flop o en otro elemento básico de memoria que permita esta operación. Por tanto, una ROM puede implementarse solamente mediante componentes combinacionales.

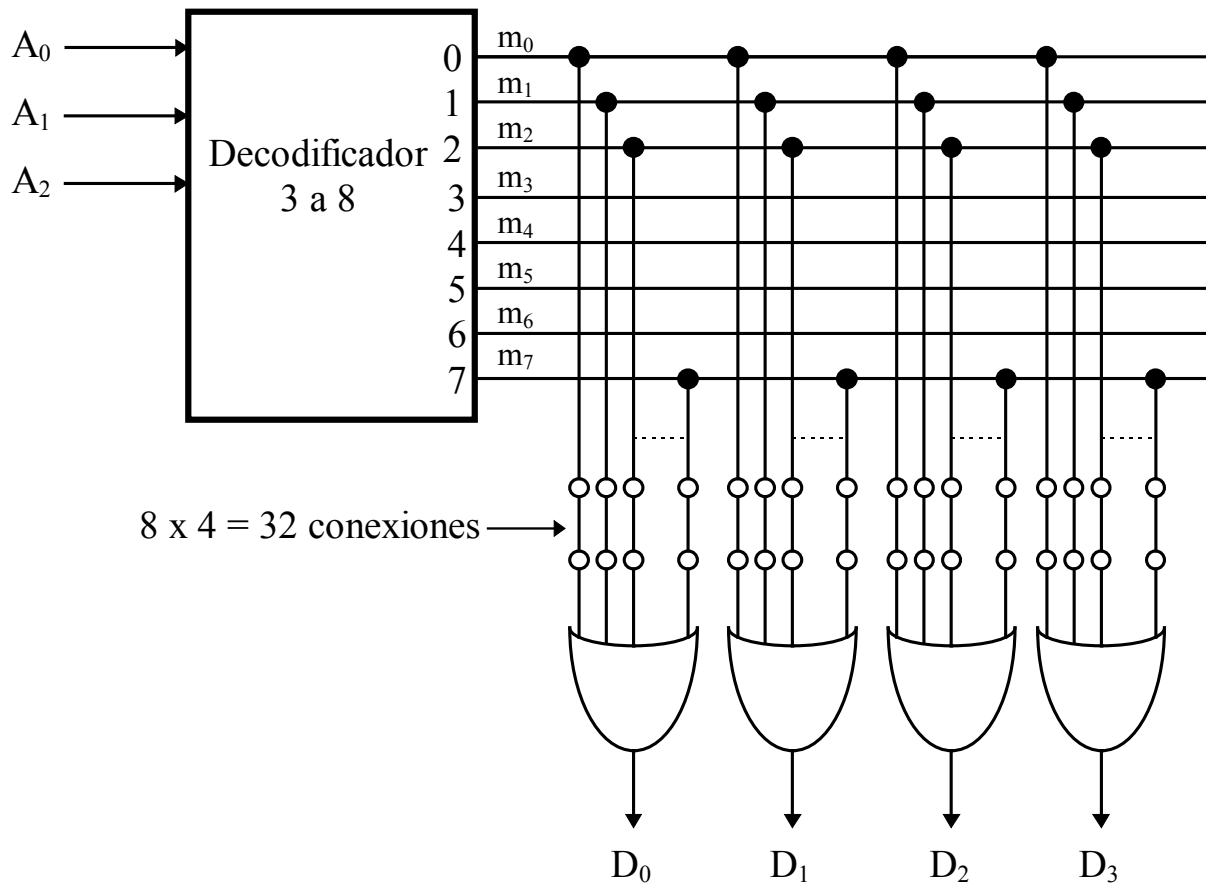
Una ROM es un circuito combinacional que consta de un decodificador, y varias puertas OR que constituyen un codificador. Las conexiones entre las salidas del decodificador y las entradas de las puertas OR son las que determinan su contenido.

En la figura 3-24 se muestra la estructura interna de una ROM con una organización de 8 x 4. Tiene 8 palabras ( $m_0, m_1, \dots, m_7$ ), que corresponden a cada una de las salidas del decodificador, por lo que necesita 3 líneas de direcciones [ $A_0, A_1, A_2$  ( $8 = 2^3$ )]. Al ser el tamaño de la palabra 4 bits, tendrá 4 líneas de datos ( $D_0, D_1, D_2$  y  $D_3$ ), correspondiendo cada una de ellas a la salida de una puerta OR. Las conexiones programables son las que determinan el contenido de la memoria.

Dependiendo de la dirección suministrada a la memoria solamente se activará (valor 1 lógico) una de las salidas del decodificador, mientras que todas las restantes salidas estarán en 0 lógico. Esta salida activa selecciona la palabra cuyo contenido se quiere leer, de forma que si está conectada a una puerta OR almacena un 1 para el bit de datos generado por esa puerta. Por tanto, para poder fijar el valor de las 8 palabras cada puerta OR debe tener tantas entradas como palabras contenga la memoria. En este caso serán de 8 entradas.

De forma genérica, una ROM con una organización  $n \times m$ , constará de un decodificador de  $p$  a  $n$  y de  $m$  puertas OR de  $n$  entradas, donde  $p$  se obtiene de la relación  $n = 2^p$ . Las salidas de una ROM pueden acoplarse a un bus mediante un buffer triestado

controlado por una señal que cuando tome valor 0 deja pasar el valor de la celda de memoria al bus y cuando tome valor 1 pone la salida a alta impedancia.



**Figura 3-24 Estructura interna de una memoria ROM**

Para ilustrar el proceso de programación de una ROM, vamos a indicar las conexiones necesarias para que la memoria de la figura anterior tenga el contenido de la siguiente tabla.

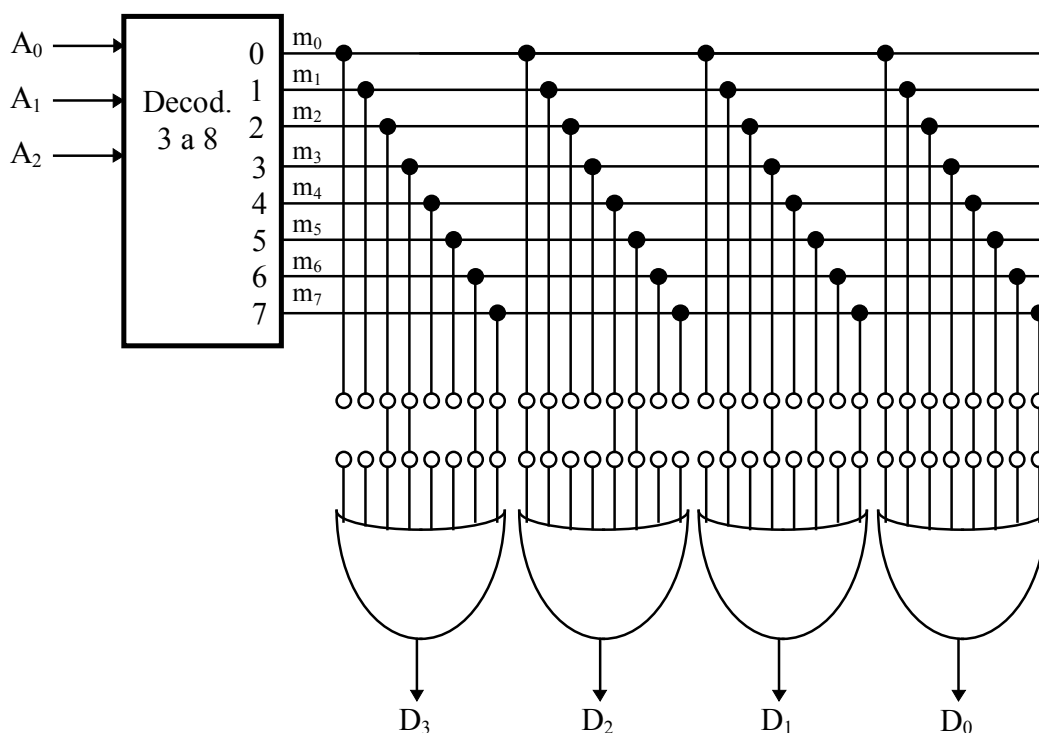
Posición de memoria	Dirección			Palabra de datos			
	A <sub>2</sub>	A <sub>1</sub>	A <sub>0</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>
m <sub>0</sub>	0	0	0	0	1	0	1
m <sub>1</sub>	0	0	1	0	1	1	1
m <sub>2</sub>	0	1	0	1	0	0	1
m <sub>3</sub>	0	1	1	1	0	1	1
m <sub>4</sub>	1	0	0	0	1	0	1
m <sub>5</sub>	1	0	1	0	1	1	1
m <sub>6</sub>	1	1	0	1	0	0	1
m <sub>7</sub>	1	1	1	1	0	1	1

**Tabla 3-4**

Según la tabla 3-4, por ejemplo  $D_2 = 1$  cuando y sólo cuando  $m_0 = 1$ ,  $m_1 = 1$ ,  $m_4 = 1$  ó  $m_5 = 1$ . Por tanto, habrá que conectar las entradas de la puerta OR correspondiente a la salida  $D_2$  a las salidas del decodificador  $m_0$ ,  $m_1$ ,  $m_4$  y  $m_5$ . De igual forma se obtendrían las

conexiones necesarias para el resto de salidas. Si nos fijamos en la salida  $D_0$ , ésta está a 1 siempre, ya que todas las palabras tienen este bit a 1. Por esto, como ya se comentó, las puertas OR tienen que ser de  $n$  entradas. Las conexiones también pueden deducirse de la siguiente forma. Consideremos por ejemplo la dirección de entrada  $A_2A_1A_0 = 011$ . Para esta dirección se selecciona la palabra  $m_3$ , cuyo contenido es  $D_3D_2D_1D_0 = 1011$ . Como para este valor de direcciones solamente se pone a 1 la salida  $m_3$  del decodificador, para fijar el contenido anterior basta con conectar  $m_3$  a las entradas de las puertas  $D_3$ ,  $D_1$  y  $D_0$ . Se supone que la ROM se diseña de forma que una entrada sin conectar a las puertas OR, éstas la consideran como 0 lógico.

En la figura 3-25 se indican las conexiones internas de la ROM.



**Figura 3-25 Conexiones internas de la memoria ROM que corresponde a la tabla 3-1**

La ROM debe programarla el fabricante según las especificaciones del cliente. Las conexiones internas se indican mediante un negativo fotográfico denominado máscara. Cada ROM tendrá una máscara en función de la información almacenada. Como las máscaras son costosas, éste tipo de ROM es sólo económico si se fabrican en cantidades elevadas. Para solucionar este inconveniente se introdujeron en el mercado las ROM PROGRAMABLES, las cuales se analizarán posteriormente. Como ya se comentó anteriormente, la operación de almacenar información en una memoria ROM se le denomina programación. Por esto, a la ROM se le suele denominar ROM programable por máscara (MROM).

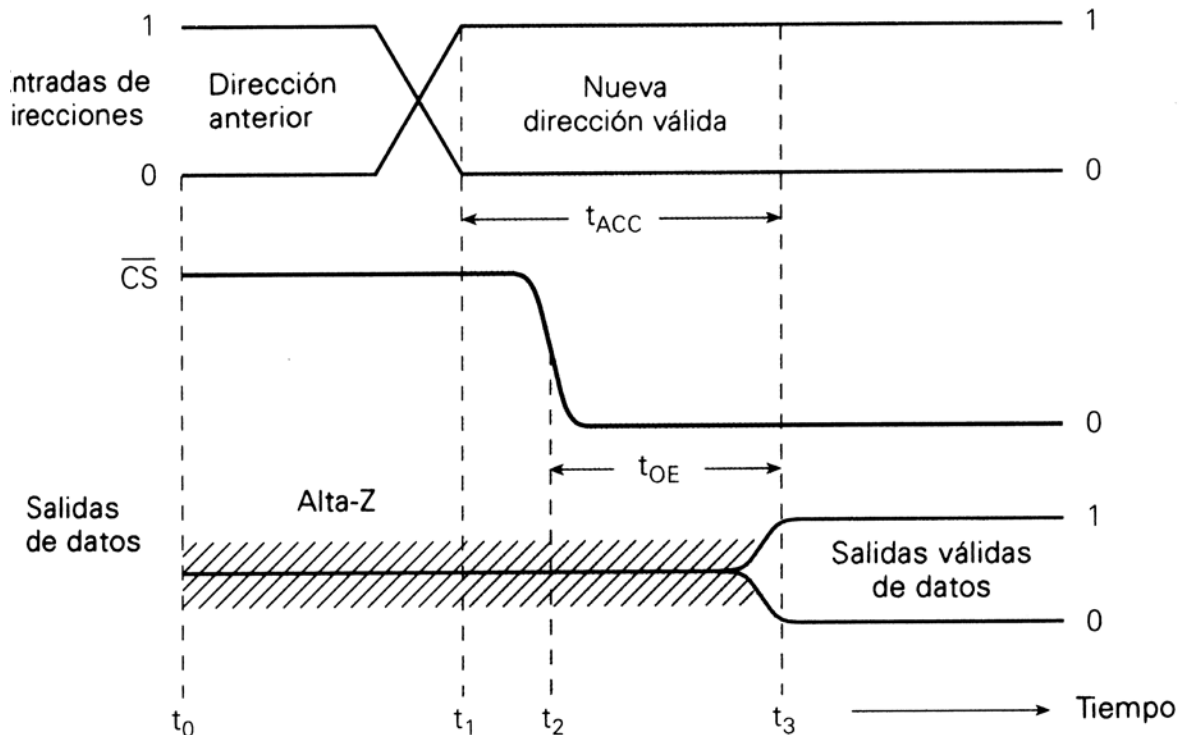
Se suele implementar de dos formas diferentes el codificador de una ROM: mediante una matriz de diodos o mediante una matriz de transistores.

## CICLO DE LECTURA DE UNA ROM

En la figura 3-26 se muestra la temporización de una operación de lectura en una ROM. La primera forma de onda representa las entradas de dirección, la del medio es una selección

de CI activa en BAJO,  $\overline{CS}$ , y la de más abajo representa las salidas de datos. En el tiempo  $t_0$ , las entradas de dirección están en un nivel específico, algunas en ALTO y algunas en BAJO.

$\overline{CS}$  es ALTA, de manera que las salidas de datos de la ROM se encuentran en su estado Alta-Z (representado por la línea sombreada).



**Figura 3-26 Ciclo de lectura de una memoria ROM**

Antes de  $t_1$ , las entradas de dirección cambian a una nueva dirección para realizar una nueva operación de lectura. En  $t_1$ , la nueva dirección es válida; es decir, cada entrada de dirección está en un nivel lógico válido. En este punto la circuitería interna de la ROM empieza a decodificar las nuevas entradas de dirección para seleccionar el registro que enviará sus datos a los buffers de salida. En  $t_2$ , la entrada  $\overline{CS}$  es activada para habilitar los buffers de salida. Finalmente, en  $t_3$ , las salidas cambian del estado Alta-Z a los datos válidos que representan los almacenados en la dirección especificada.

El retraso entre  $t_1$  y  $t_3$ , cuando la nueva dirección y las salidas de datos se vuelven válidas, es el tiempo de acceso  $t_{ACC}$ . En general, las ROMs bipolares tienen un tiempo de acceso que va de 30 hasta 90 ns; para los dispositivos NMOS este tiempo varía de 35 hasta 500 ns.

Otro importante parámetro de temporización es el *tiempo de salida*,  $t_{OE}$ , que es el retraso entre la entrada  $\overline{CS}$  y la salida de datos válida. Valores comunes de  $t_{OE}$  son de 10 a 20 ns para ROMs bipolares y de 25 a 100 ns para ROM MOS. Este parámetro de temporización es importante en situaciones donde las entradas de dirección ya están en sus nuevos valores, pero las salidas de la ROM aún no han sido habilitadas. Cuando  $\overline{CS}$  pasa a BAJA para habilitar las salidas, el retraso será  $t_{OE}$ .

### 3.3.2. CLASIFICACIÓN DE LAS ROM

En la figura 3-27 se muestra una clasificación de las memorias ROM, también denominadas *memorias inertes*:

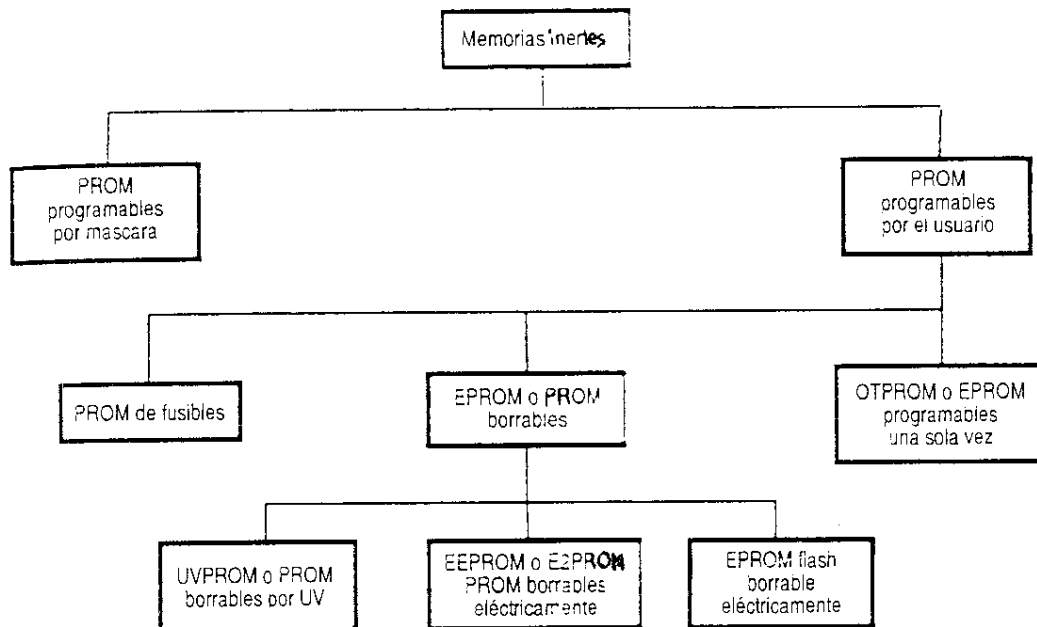


Figura 3-27. Clasificación de las memorias ROM

### 3.3.3. ROM PROGRAMABLES (PROM).

Las PROM (Programmable Read Only Memory) se emplean en sistemas digitales cuyo volumen de producción es bajo, y cuando éstos están en la fase de prototipo (diseño y puesta a punto de un sistema dado). De esta forma se evita el inconveniente de las ROM cuya fabricación implica un elevado coste inicial por el diseño de la máscara.

Dependiendo del elemento de interconexión que se emplee las PROM se pueden programar una sola vez o varias. Las primeras se denominan genéricamente PROM y las segundas EPROM, las cuales a su vez se clasifican en varios tipos. Dentro de las PROM se distinguen dos tipos: la PROM de fusibles y la OTPROM. A continuación se analizan estos dos tipos de PROM.

#### **PROM DE FUSIBLES.**

La estructura de una PROM de fusibles es la misma que la de la ROM, excepto que el elemento de interconexión es programable. Este consiste en un fusible que puede fundirse. Una PROM no programada (virgen) tiene todos los fusibles intactos, y para programarla (almacenar la información) se funden los fusibles que fijan un 0 lógico, aplicando un voltaje elevado (aproximadamente 21V). Este proceso se realiza mediante equipos comerciales de programación, que permiten la introducción del programa por teclado o desde un disco, y luego fundir los fusibles y verificar automáticamente, que la PROM está correctamente programada. Esta operación de verificación consiste en leer el contenido de la PROM, una vez programada, y posteriormente compararlo con el original.

Como se puede deducir de lo expuesto en el apartado anterior, una PROM de fusibles es una ROM programable una sola vez, ya que una vez fundido un fusible, éste no puede reconstruirse.

### **OTPROM.**

OTPROM corresponde a las siglas de One Time Programmable Read Only Memory, es decir, se trata de una memoria de sólo lectura programable una sola vez. Según lo anterior, sería una memoria semejante a la PROM de fusibles analizada anteriormente. Se diferencia de ésta en que el elemento de conexión es el mismo que el de las UVPRM, que se comentarán posteriormente. Se trata pues de una UVPRM a la que se le ha inhibido el mecanismo de borrado para reducir su coste de fabricación.

### **3.3.4. ROM PROGRAMABLES Y BORRABLES. EPROM.**

Una memoria EPROM (*Erasable Programmable Read Only Memory*) es un grupo de varios tipos de memorias de sólo lectura reprogramables, por lo que se puede programar un número elevado de veces. Para poder programarla antes debe borrarse su contenido, y después se establecen las conexiones necesarias según la información que tenga que almacenar.

El mecanismo de programación es eléctrico y semejante al de las PROM, ya que debe aplicarse una tensión elevada y tiene que hacerse generalmente mediante un equipo de programación. Evidentemente se diferencia de las PROM de fusibles en el elemento de conexión que emplea. Las EPROM utilizan un componente electrónico semiconductor cuyo estado (cerrado y abierto) se puede programar. Una EPROM virgen tiene estos componentes en estado abierto. La programación consiste en cerrar los componentes necesarios, pero en todos los tipos de EPROM para poder cerrarlos, antes deben estar en estado abierto. La operación de poner en el estado abierto los distintos componentes de conexión, se denomina borrado de la EPROM. Por tanto, cualquier EPROM para programarla, antes debe borrarse.

Dependiendo del mecanismo de borrado se distinguen los siguientes tipos de EPROM:

- UVPRM
- EEPROM
- FLASH EPROM

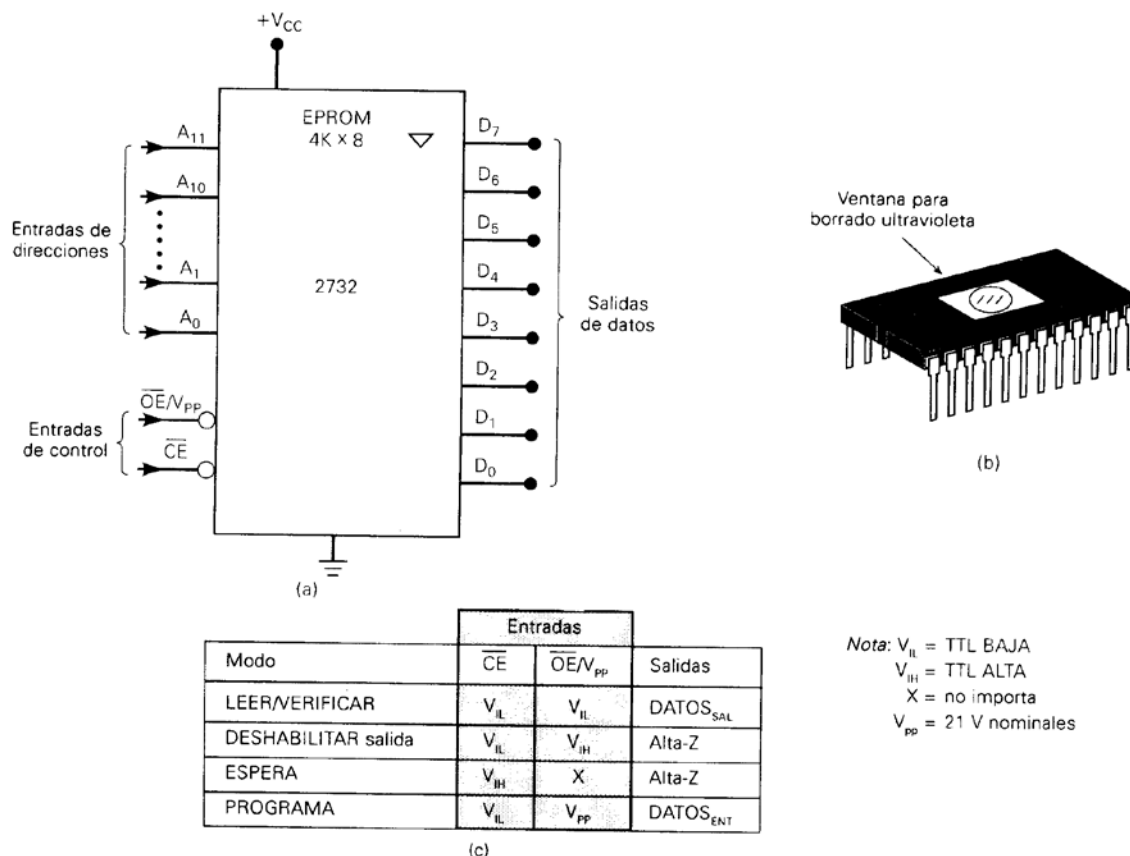
#### **3.3.4.1. UVPRM**

Se programan eléctricamente y se borran exponiéndolas a una radiación ultravioleta de longitud adecuada. Por ello, su encapsulado está provisto de una ventana de cuarzo, que permite ver el chip y deja pasar la luz. Cuando se borra una UVPRM se borra todo su contenido. Se les denomina generalmente EPROM, por ser el primer tipo de memoria ROM borrrable introducida en el mercado.

En la figura 3-28 se muestra una memoria EPROM de 4Kx8 con 12 líneas de direcciones y 8 salidas de datos. Además tiene dos señales de control  $\overline{CE}$ , que sirve para



activar y desactivar la memoria, y  $\overline{OE}/V_{PP}$ , para leer poniendo la señal a 0 o para escribir la memoria colocando la señal a una tensión  $V_{PP}$ . En la figura 3-28 se muestra también la tabla de funcionamiento.



**Figura 3-28. Memoria EPROM de 4Kx8 y su tabla de funcionamiento**

Si CE está a nivel alto no selecciona la EPROM y no se puede leer ni programar. En esta situación las salidas están en alta impedancia y el consumo de potencia es muy reducido.

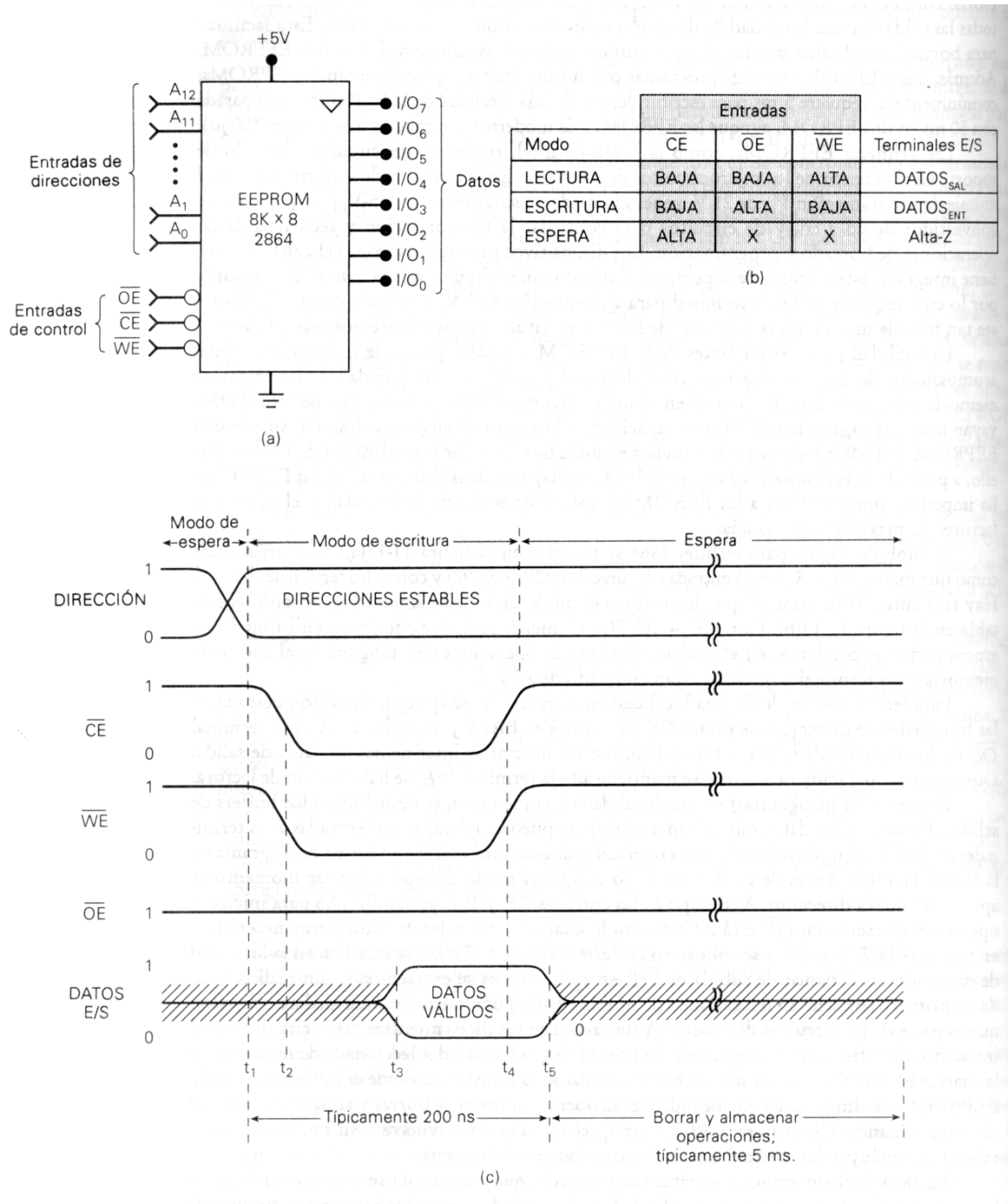
Para realizar una operación de lectura normal debe activarse CE y OE, y desactivarse PGM. La EPROM pone en sus salidas de datos el contenido de la palabra seleccionada por las líneas de direcciones. Si OE está desactiva las líneas de datos estarán en alta impedancia.

Para programar el dispositivo se debe activar CE y PGM y desactivar OE. Por otra parte, se tiene que aplicar la tensión de programación en la patilla  $V_{PP}$ .

### 3.3.4.2. PROM BORRABLES ELÉCTRICAMENTE.EEPROM

EEPROM corresponde a las siglas *Electrically Erasable PROM*, que significa PROM borrable eléctricamente. Para su grabación no es necesario ningún equipo de programación, pudiéndose programar dentro del propio sistema digital. Para el diseñador se trata de una memoria de lectura y escritura (tiene patillas de control para ambas operaciones), pero en la realidad no es así, ya que la operación de escritura es de programación al tener que borrarse previamente el contenido de la palabra que se desea escribir.

En la figura 3-29 se muestra una EEPROM de 8Kx8 y su tabla de funcionamiento según las distintas señales de control, así como su cronograma.



**Figura 3-29. Memoria EEPROM de 8Kx8. Tabla de funcionamiento y cronograma**

A diferencia de la UVPROM el borrado es selectivo, es decir, solamente se borra el contenido de una palabra, en concreto la que se va a escribir.

Por permitir la programación dentro del propio sistema digital se suele emplear para almacenar información que el software puede cambiar en cualquier momento, como es el caso de los datos de configuración del sistema.

### 3.3.4.3. FLASH EPROM

También se denominan memorias instantáneas o de ráfaga. Se diseñaron para obtener una memoria no volátil de gran capacidad de almacenamiento (igual o mayor que el de las

UVPROM), pero que pudiesen programarse dentro del sistema. Es decir, el objetivo fue diseñar una memoria que tuviese las ventajas de las UVPROM y EEPROM. Por tanto, son memorias programables y borrables eléctricamente. Se diferencian de las EEPROM en los siguientes aspectos:

- Las operaciones de borrado y programación son más rápidas. De ahí su nombre FLASH.
- Hasta hace poco tiempo para programarlas era necesario emplear una tensión elevada al igual que para las UVPROM. En la actualidad existen fabricantes que suministran FLASH EPROM que puede programarse con la misma tensión de funcionamiento (+5V).
- El borrado no es selectivo. Permiten el borrado completo o a nivel de página (un bloque de palabras).

La celda de memoria está compuesta por un solo transistor, como el de la EPROM, en vez de dos como el de la EEPROM. Tiene una capa de óxido más delgada en la puerta, por lo que se puede borrar eléctricamente y de una forma más rápida que la EEPROM. Esta estructura de la celda básica le da las siguientes ventajas respecto a las UVPROM y EEPROM:

- Tienen una mayor densidad de almacenamiento por su celda más sencilla (respecto a la EEPROM) y UVPROM (tecnología más avanzada). Por tanto, su coste por bit es bastante menor.
- Se fabrican con capacidades superiores.
- Menores tiempos de borrado y programación.

A continuación se describe el circuito integrado 28F256A.

### Memoria Flash EPROM 28F256A

El CI 28F256A es una memoria FLASH EPROM CMOS de 32K x 8. En la figura 3-30 se muestra su representación a nivel de bloque.

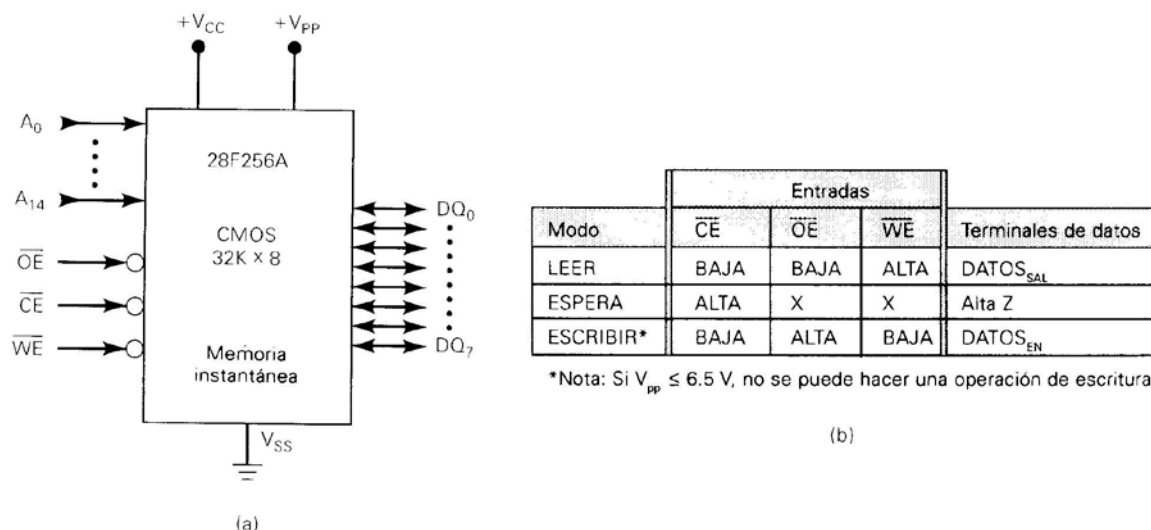


Figura 3-30. Diagrama y tabla de funcionamiento de la memoria 28F256A

### 3.3.5. APLICACIONES DE LAS MEMORIAS ROM.

Páginas 651, 652, y 653. *Sistemas digitales. Principios y fundamentos.* Ronald J. Tocci

- **IMPLEMENTACIÓN DE LÓGICA COMBINACIONAL**

Como ya se sabe una función lógica de  $n$  variables de entrada se puede implementar mediante un decodificador de  $n$  a  $2^n$  y una puerta OR. Como la estructura interna de una ROM de  $2^n \times m$  consta de un decodificador de  $n$  a  $2^n$  y  $m$  puertas OR, se puede implementar un circuito combinacional de  $n$  entradas y  $m$  salidas. Para determinar el contenido de la ROM, primero se obtiene de la tabla de verdad de las funciones lógicas, y posteriormente se asignan variables de entrada y de salida a las líneas de direcciones y datos, respectivamente, teniendo en cuenta los pesos de cada una. El contenido de la memoria vendrá indicado por la propia tabla de verdad, en la que cada combinación binaria de las variables de entrada selecciona cada una de las palabras, y el valor de todas las funciones lógicas para esa combinación el contenido de la palabra correspondiente.

### 3.4. MEMORIAS DE ACCESO SECUENCIAL (SAM).

En las memorias RAM estudiadas anteriormente el acceso se realiza directamente a la palabra deseada mediante su dirección. El tiempo de acceso es el mismo cualquiera que sea la dirección accedida. Por estos motivos y otros muchos comentados anteriormente este tipo de memorias se suelen utilizar como memoria principal de una computadora.

Existen otro tipo de memorias a las que no se puede acceder (para escritura o lectura) directamente a la posición deseada, sino secuencialmente. Esto quiere decir que para llegar a leer una palabra determinada debemos acceder secuencialmente a todas las palabras que se encuentran desde la última palabra accedida y la que deseamos acceder. Estas memorias son las que genéricamente se les denomina de acceso secuencial.

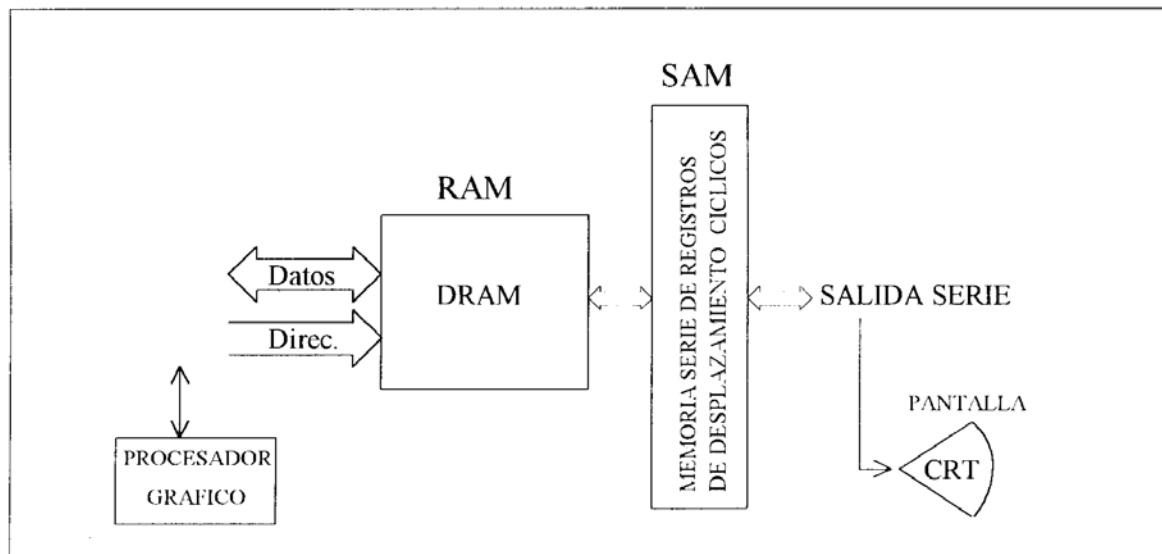
El tiempo de acceso en las memorias secuenciales no es fijo si no que depende del número de palabras que se encuentran entre la actual y la nueva a la que queremos acceder. Por este motivo y otros que se irán viendo a lo largo del capítulo hacen muy complejo el uso de estas memorias como memoria principal de una computadora, y por lo general este tipo de memorias suelen utilizarse para aplicaciones específicas.

En este apartado trataremos las memorias secuenciales más utilizadas: Memorias serie de registros de desplazamiento, memorias FIFO y memorias LIFO.

#### 3.4.1. MEMORIA SECUENCIAL CON REGISTRO DE DESPLAZAMIENTO CIRCULAR.

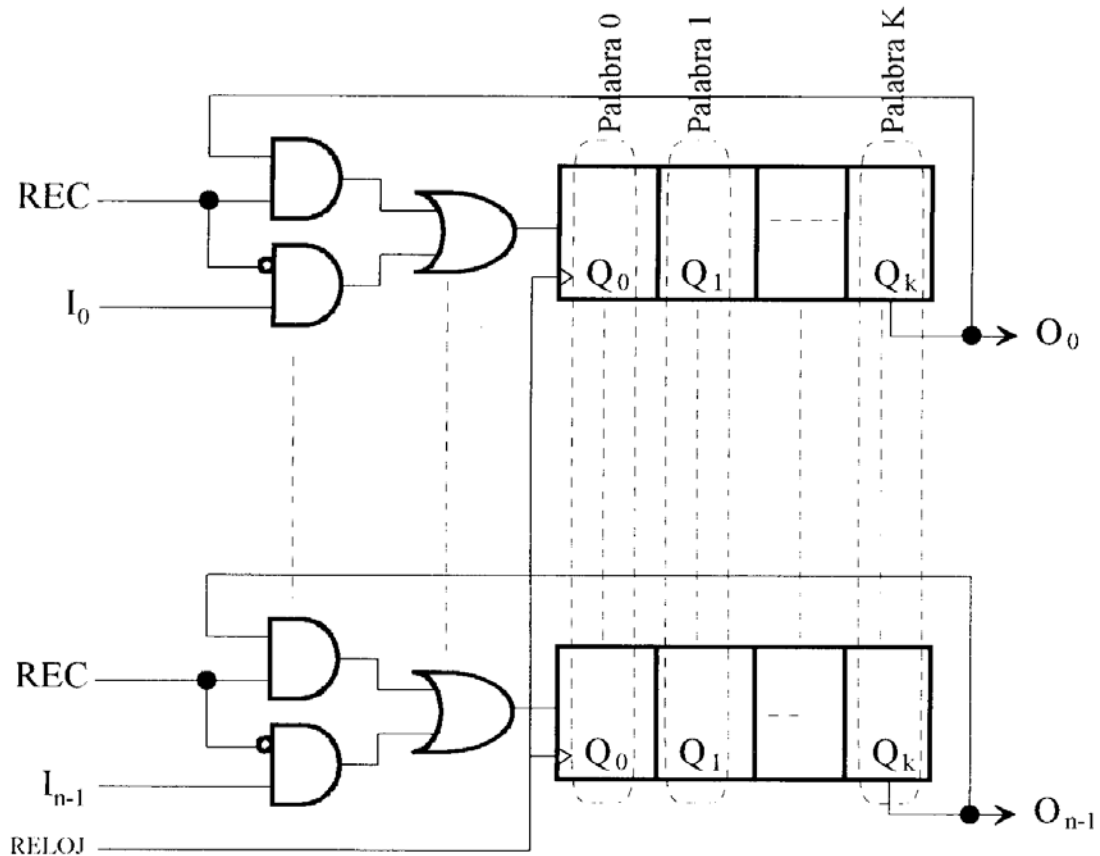
Se les suele llamar también memorias de registros de desplazamiento con recirculación, o memorias de registros de desplazamiento cíclicos.

Estas memorias suelen utilizarse como memoria de video, en donde los datos deben presentarse periódicamente al sistema de video para refrescar la pantalla. La figura siguiente muestra un diagrama de bloques del sistema donde suelen integrarse este tipo de memorias.



**Figura 3-31. Diagrama de bloques de un sistema de vídeo donde puede apreciarse la utilización de las memorias serie de registros de desplazamientos**

- ❑ A la memoria serie suele acompañarla en los sistemas de vídeo otra memoria RAM que se utiliza como almacenamiento masivo.
- La estructura de esta memoria se presenta en la figura 3-32:



**Figura 3-32. Memoria serie basada en registros de desplazamientos cíclicos**

- ❑ En la figura 3-32 las palabras se introducen a través de las líneas "I" y se leen a través de las líneas "O".
- ❑ Este registro de desplazamiento cíclico permite dos operaciones en función de la señal de control REC. Cuando  $REC = 1$ , permite el desplazamiento cíclico a la derecha de los datos (recirculación), y cuando  $REC = 0$  permite que se introduzcan los datos por la entrada serie I.
- ❑ La anchura de palabra de la memoria estará determinada por el número de registros, y la capacidad de la memoria por el tamaño de los registros.

### 3.4.2. MEMORIAS LIFO

Este tipo de memoria secuencial opera de forma que al leer, se lee el dato que se escribió en último lugar. Los datos se van leyendo en orden inverso a como se escribieron.

El nombre corresponde a las iniciales del texto en inglés "Last In First Out" (último en entrar primero salir) que describe como opera esta memoria.

Se les suele llamar también memorias PILA.

La operación de escritura en una memoria Pila se le llama PUSH (empujar) y a la operación de lectura se le llama POP (sacar).

Se utilizan dos estructuras básicas para construir una memoria LIFO.

- **LIFO basada en registros de desplazamiento.**

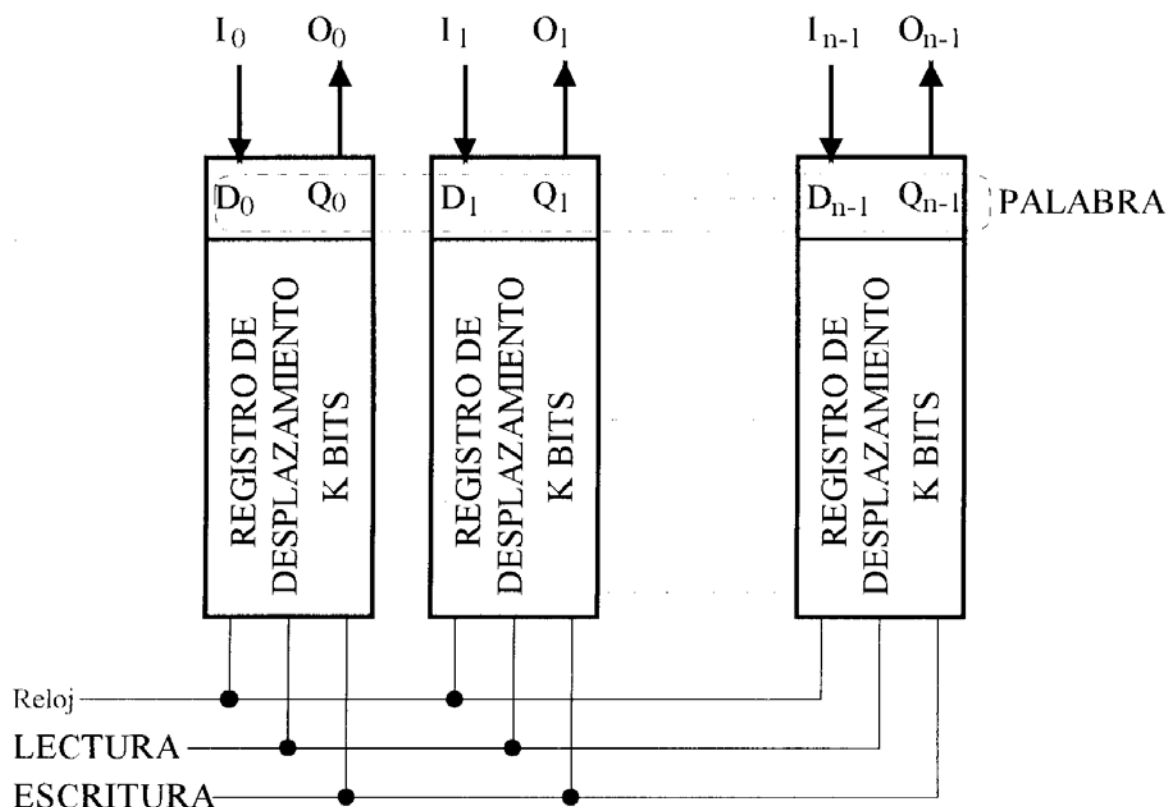
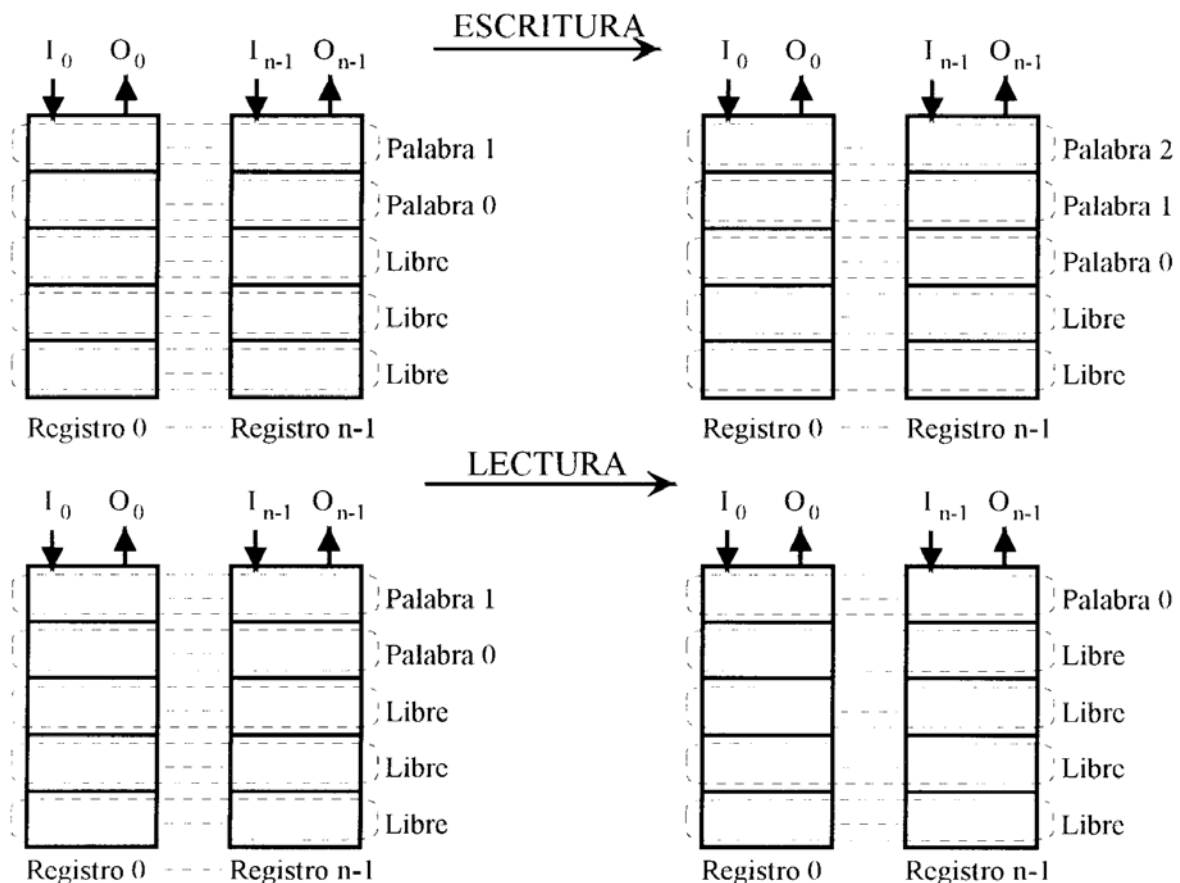


Figura 3-33. Estructura de una LIFO basada en registros de desplazamiento

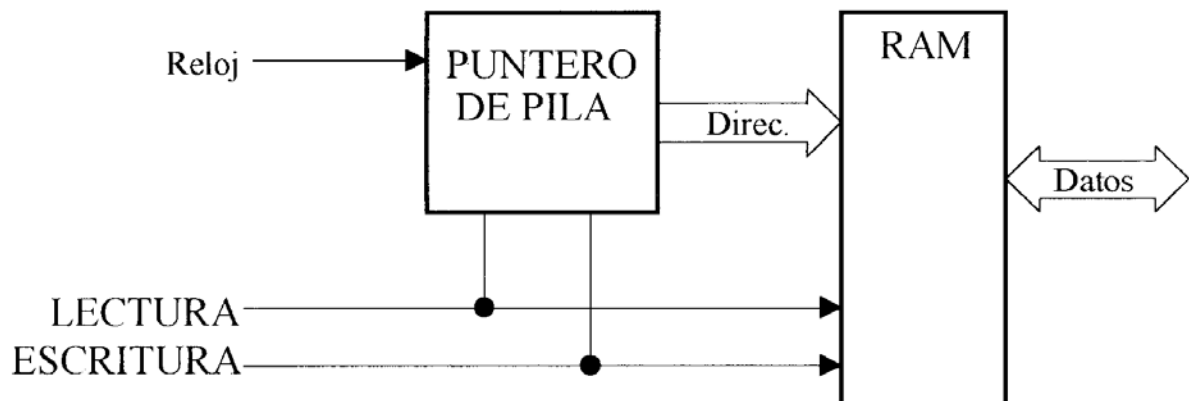
- ❑ Cada registro contiene un bit de la palabra total. Si la LIFO tiene anchura de palabra de 8 bits contendrá 8 registros de desplazamiento.
- ❑ Las palabras se introducen a través de las líneas "I" y se leen a través de las líneas "O".
- ❑ Los registros de desplazamiento deberán realizar las operaciones de desplazamiento a derecha e izquierda, para escribir y leer, y bloqueo si no se realiza ninguna operación sobre la LIFO.



**Figura 3-34. Operación de escritura y lectura sobre una LIFO con arquitectura basada en registros de desplazamiento**

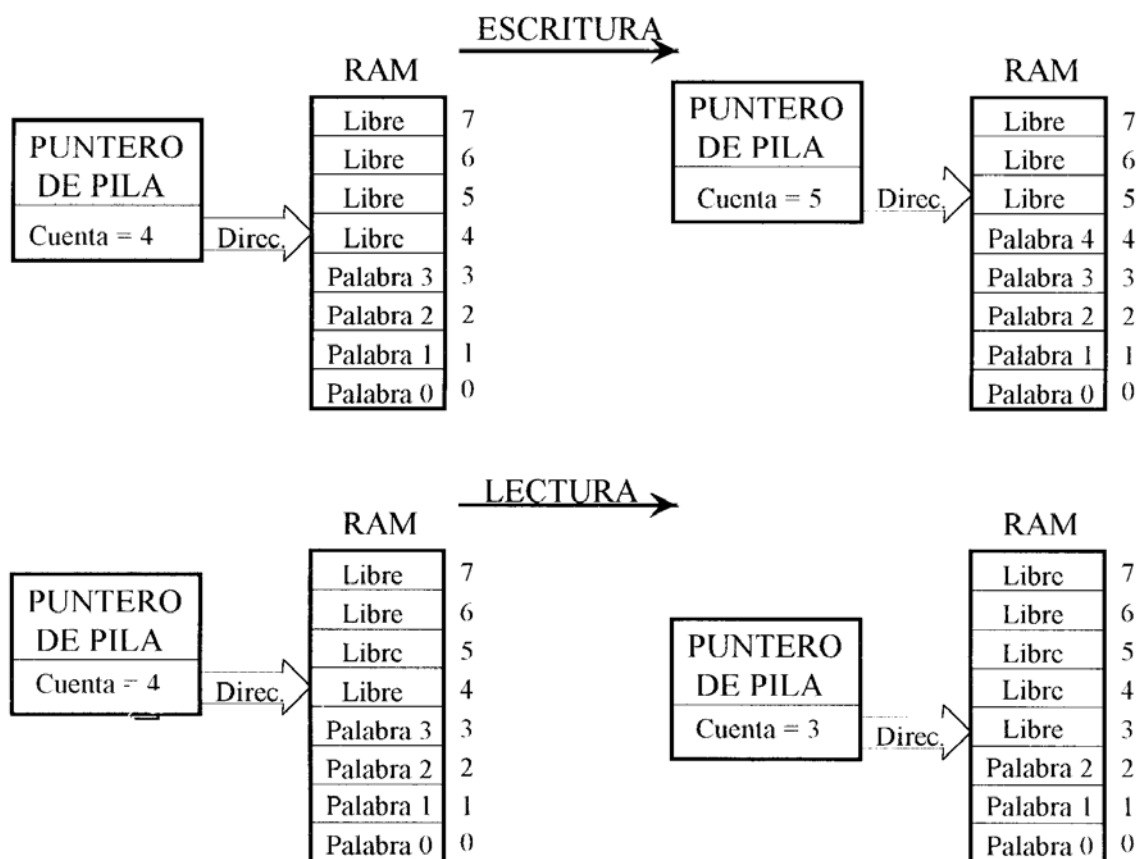
- ❑ Las palabras (datos) se introducen a través de las líneas "I" y se leen a través de las líneas "O" y están numeradas en forma creciente al orden en como se han introducido siendo la "palabra 0" la que primero se escribió.
- ❑ En una operación de escritura se produce un desplazamiento de los datos hacia abajo, y en una operación de lectura se produce un desplazamiento de los datos hacia arriba.

- **LIFO basada en una RAM y un puntero.**



**Figura 3-35. LIFO construida con una RAM y un puntero**

- ❑ El puntero de Pila es un contador que permite una cuenta ascendente y descendente controlada por las señales de lectura y escritura, y genera las líneas de direcciones de la RAM.
- ❑ El puntero apunta generalmente a la última posición libre.
- ❑ La profundidad de la memoria Pila así como la anchura de palabra vendrá determinada por la capacidad y anchura de la RAM.



**Figura 3-36. Operación de escritura y lectura sobre una LIFO construida con una RAM y un puntero**



- ❑ Las palabras (datos) están numeradas en forma creciente al orden en como se han introducido siendo la "palabra 0" la que primero se escribió.
- ❑ En una operación de escritura el dato se escribe en la última libre y se incrementa el puntero. En una operación de lectura se decrementa el puntero y se lee el dato.
- ❑ En el ejemplo anterior los datos se van introduciendo en posiciones ascendentes consecutivas, de igual forma existen memorias Pila en la que los datos se van introduciendo en posiciones decrecientes consecutivas, de forma que al escribir se decrementa el puntero y al leer se incrementa.

Las memorias Pila son fundamentales en los sistemas basados en microprocesadores. En este caso el puntero de pila este integrado en el propio microprocesador , y la memoria RAM de la Pila es una parte de la memoria total del sistema.

### **3.4.3. MEMORIAS FIFO.**

Este tipo de memoria secuencial opera de forma que al leer, se lee el dato que se escribió en primer lugar. Los datos se van leyendo en el mismo orden a como se escribieron.

El nombre FIFO corresponde a las iniciales del texto en inglés "**F**irst **I**n **F**irst **O**ut" (primero en entrar primero en salir), que describe como opera esta memoria.

Las memorias FIFO permiten que los datos se escriban a una determinada velocidad y se vayan leyendo a otra, lo que las hace muy útil para la transferencia de datos entre sistemas que operan a distinta velocidad. Es decir la memoria FIFO actúa como "buffer" de datos entre sistemas distintos.

Un sistema puede escribir datos rápidamente en la memoria FIFO y dedicarse a otras tareas, mientras el otro sistema los va leyendo a otra velocidad más lenta, o viceversa.

Un ejemplo donde una FIFO aumentaría el rendimiento de la transmisión de datos entre dos sistemas sería la transmisión de datos de un computador a una impresora. El computador escribiría rápidamente los datos a imprimir en la memoria FIFO, y mientras la impresora los lee puede dedicarse otras tareas.

Se pueden encontrar muchos ejemplos de utilización de estas memorias FIFO como "buffer" de datos: entre el computador y el disco, entre un sistema de adquisición de datos y un computador, etc.

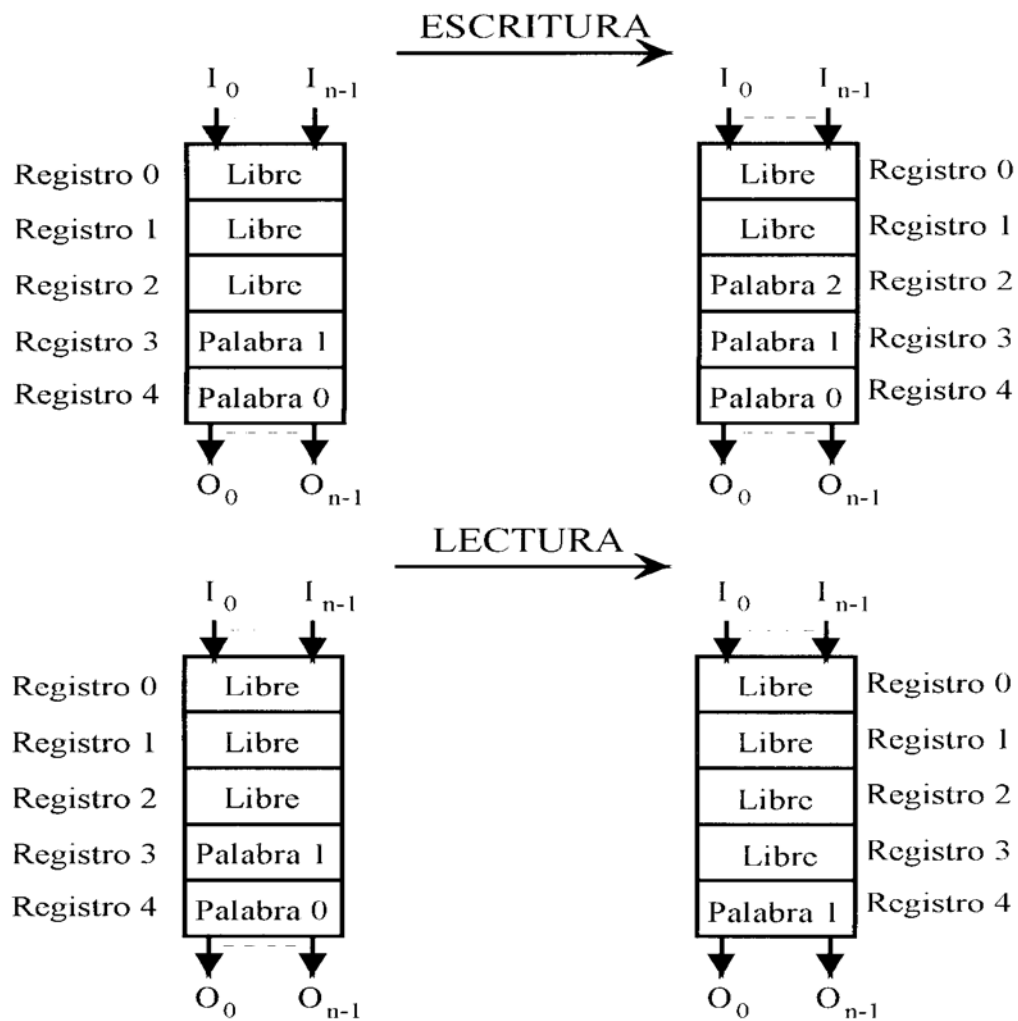


Figura 3-37. Operación de escritura y lectura de una FIFO de arquitectura con desplazamiento de datos (TOGGLE)