Caplítulo 4. Memoria Interna

Aunque podría parecer sencilla en concepto, la memoria exhibe el rango más grande de tipos, tecnología, organización, desempeño y costo que cualquier otra característica de una computadora. Ninguna tecnología, de manera única, es óptima para satisfacer los requerimientos de la computadora. Como consecuencia, un sistema típico está equipado con una jerarquía de subsistemas de memoria.

Algunos subsistemas son internos (accesibles directamente por el procesador) y otros externos (accesibles por el procesador a través de un módulo I/O). Mientras más "cercana" esté del procesador se dice que ocupa un nivel más alto en la jerarquía. Por lo tanto, en el nivel más alto se encuentran los registros del procesador. En seguida, continúan uno o más niveles de memoria caché. Cuando se utilizan múltiples niveles se denotan como L1, L2, etc. Después viene la memoria principal, que usualmente está hecha de memoria de acceso aleatorio dinámica o DRAM. Todas las anteriores se consideran como memorias internas de la computadora. La jerarquía continúa con la memoria externa, el siguiente nivel es tipicamente un disco duro fijo y uno o más niveles por debajo que consisten de medios físicos removibles como los discos ópticos.

Conforme se desciende en la jerarquía, decrece el costo por bit, aumenta la capacidad y aumenta el tiempo de acceso, es decir, disminuye la velocidad. Sería excelente utilizar solamente memoria de la más rápida, sin embargo, debido a que también es la más cara, se hace un intercambio de tiempo de acceso por costo y se utiliza más memoria de la lenta.

En general, es probable que la mayoría de los accesos futuros a memoria hechos por el procesador sea a locaciones de memoria que han sido recientemente accesadas. Así, el caché automáticamente retiene una copia de algunas de las palabras recientemente accesadas de la DRAM. Si se diseña el caché de manera adecuada, entonces la mayoría del tiempo el procesador solicitará palabras que ya se encuentran en el caché.

Las dos formas básicas de memoria semiconductora de acceso aleatorio son la RAM dinámica (DRAM) y la RAM estática (SRAM). SRAM es más rápida, más cara y menos densa que DRAM, y se utiliza para la memoria caché. DRAM se utiliza para la memoria principal.

Comunmente se utilizan técnicas de correción de errores en los sistemas de memoria. Estas técnicas involucran la adición de bits redundatnes en función de los bits de datos para formar un código de corrección. Si ocurre un error, el código lo detectará y, usualmente, lo corregirá.

Para compensar por la velocidad relativamente lenta de DRAM, una variedad de organizaciones de DRAM se han utilizado. Dos de las más comúnes son la DRAM síncrona y el RamBus DRAM. Ambas involucran el uso del reloj de sistema para la transferencia de bloques de datos.

4.1 Sistema de memoria de una computadora

Características de la memoria

La memoria se puede clasificar de acuerdo a sus características clave:

- **Ubicación:**
  - Interna
  - Externa
- **Capacidad**
  - Número de palabras
  - Número de bytes
- **Unidad de Transferencia**
  - Palabra
  - Bloque
- **Método de Acceso**
  - Secuencial
  - Directa
  - Aleatorio
  - Asociativa
- **Desempeño**
  - Tiempo de acceso
  - Tiempo de ciclo
  - Tasa de Transferencia
- **Tipo Físico**
  - Semiconductora
  - Magnética
  - Óptica
  - Magneto-óptica
- **Características Físicas**
  - Volátil/no volátil
  - Borrable/no borrable
- **Organización**
  - Módulos de memoria

La **ubicación** se refiere a si la memoria es interna o externa para la computadora. La memoria interna es directamente accesible por el procesador y usualmente se equipara con la memoria principal. Sin embargo, existen otros tipos de memoria interna: el procesador requiere de su propia memoria local, en la forma de registros; también la unidad de control del procesador puede requerir de su propia memoria local; la memoria caché es otro tipo de memoria interna. La memoria externa es accesible al procesador mediante controladores I/O. La **capacidad** para la memoria interna usualmente se expresa en términos de bytes o palabras, longitudes comunes de palabra son 8, 16 y 32 bits. Para la memoria externa, la capacidad se expresa en términos de bytes: MB, GB, TB. Un concepto relacionado es la **unidad de transferencia**, para la memoria interna, la unidad de transferencia es igual al número de lineas eléctricas que entran y salen del módulo de memoria. Puede ser igual a la longitud de palabra, aunque puede ser mayor. Considere los siguientes conceptos relacionados a la memoria interna:

- **Palabra:** la unidad natural de organización de memoria. Idealmente igual al número de bits utilizado para representar un entero y al tamaño de una instrucción. Aunque no siempre es así (Intel x86).
- **Unidades direccionables:** en algunos sistemas, la unidad direccionable es una palabra. Sin embargo, muchos sistemas son direccionables a nivel de byte.
- **Unidad de transferencia:** para la memoria principal, es el número de bits leídos o escritos cada vez. La unidad de transferencia no necesita ser igual a la palabra o a la unidad direccionable. Para la memoria externa, los datos se transmiten en unidades mucho más grandes que una palabra y se les llama bloques.

Otra distinción entre los tipos de memoria es el **método de acceso**, donde se incluyen:

- **Acceso secuencial:** los datos se organizan en unidades llamadas *records*. El acceso se debe realizar en una secuencia lineal específica. Información adicional se utiliza para separar los records y ayudar en el proceso de obtención. Se utiliza un mecanismo compartido de lectura/escritura que debe ser movido de su posición actual a la posición deseada, ignorando todos los records intermedios. Por tanto, el tiempo de acceso es muy variable. Ej. Cintas magnéticas.
- **Acceso directo:** también se utiliza un mecanismo compartido de lectura/escritura pero cada bloque individual tiene una dirección única basada en su posición física. El acceso se logra accesando directamente una vecindad general aunado a una búsqueda secuencial, para alcanzar la ubicación final. Ej. Unidades de Disco
- **Acceso aleatorio:** cada localidad de memoria direccionable tiene mecanismo único de direccionamiento cableado físicamente. El tiempo de acceso es constante puesto que es independiente de los accesos previos. Ej. Memoria principal y algunos sistemas de caché.
- **Asociativo:** es un tipo de acceso aleatorio que permite hacer una comparación utilizando localidades de bit específicas dentro de una palabra. Así, una palabra es obtenida basada en una porción de sus contenidos en lugar de por su dirección. El tiempo de acceso es constante e independiente de accesos previos. Algunas memorias caché utilizan accesos asociativos.

Desde el punto de vista del usuario, las dos características más importantes de la memoria son la **capacidad** y el **desempeño**. Se utilizan 3 parámetros para medir el desempeño:

- **Tiempo de acceso:** para la memoria de acceso aleatorio, es el tiempo que toma efectuar una operación de lectura o memoria, desde el instante en que se presenta una dirección a la memoria al instante en que los datos se han almacenado o están listos para su uso. Para memorias de no acceso aleatorio, es el tiempo que toma posicionar el mecanismo de escritura/lectura sobre la posición deseada.
- **Tiempo del ciclo de memoria:** se aplica a la memoria de acceso aleatorio y consiste en el tiempo de acceso más el tiempo adicional requerido antes de que pueda comenzar un segundo acceso. El tiempo adicional puede ser requerido para que pasen los periodos transitorios en las lineas de señal o en regenerar los datos si la lectura es destructuva. Note que el tiempo de ciclo de memoria tiene que ver con el bus, no con el procesador.
- **Tasa de transferencia:** la tasa a la cual los datos pueden ser transferidos desde o hacia una unidad de memoria. Para la memoria de acceso aleatorio es 1.

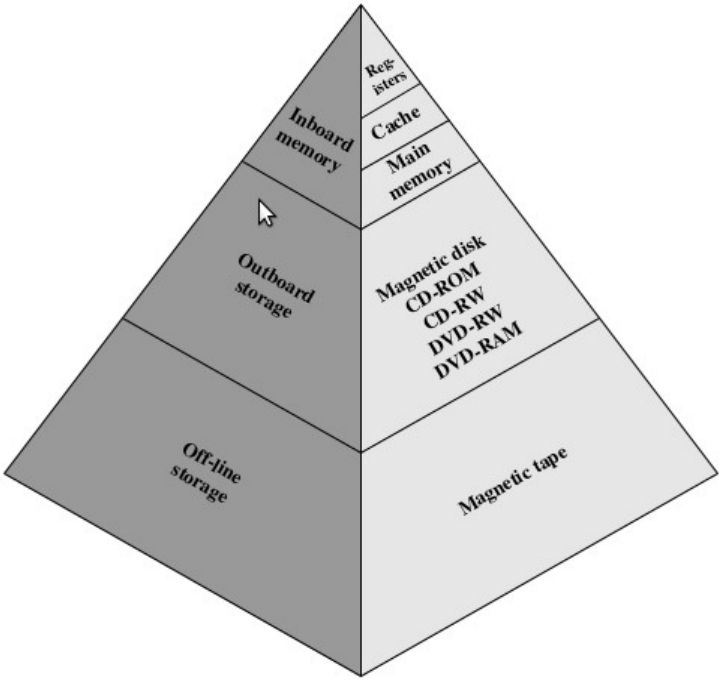
Una gran variedad de características físicas son importantes. En la memoria volátil, la información decae naturalmente o se pierde cuando el sumistro electrico se apaga. En una memoria no volátil, la información (una vez grabada) se mantiene sin deteriorarse hasta que es cambiada deliberadamente. Las memorias de superficie magnética son no-volátiles. Las memorias semiconductoras pueden ser de ambos tipos. Las memorias no-borrables no pueden ser alteradas, las memorias semiconductoras no-borrables se conocen como memorias de sólo lectura (ROM, *read-only-memory*).

La Jerarquía de Memoria

Existe un intercambio o compensación entre las 3 características claves de la memoria: capacidad, tiempo de acceso y costo. Las siguientes relaciones se han mantenido a través de los avances en implementación y tecnología de módulos de memoria:

- Menor tiempo de acceso, mayor costo por bit.
- Mayor capacidad, menor costo por bit.
- Mayor capacidad, mayor tiempo de acceso.

El dilema es claro, nos gustaría tener tecnologías de memoria que nos dieran la mayor capacidad, por su bajo costo por bit; así como el menor tiempo de acceso posible, para cumplir los requisitos de desempeños. Sin embargo, no se pueden obtener estas ventajas sin aumentar el costo por bit. La manera de resolver el problema planteado es no depender de un solo componente de memoria, sino emplear una jerarquía.



Conforme se desciende en la jerarquía: (a).disminuye el costo por bit, (b).aumenta la capacidad, (c).aumenta el tiempo de acceso y (d).**disminuye la frecuencia de accesos del procesador al tipo de memoria**. Así, las memorias rápidas, pequeñas y caras son suplementadas por memorias más largas, más lentas y más baratas. El uso de una jerarquía de memoria reduce el tiempo de acceso promedio sólo si se cumplen las condiciones (a)-(d).

Es fácil observar a partir del espectro de sistemas de memoria existentes que se satisfacen las condiciones (a) – (c). La condición (d) se cumple gracias a un principio conocido como **localidad de referencia.** Durante la ejecución de un programa, las referencias a memoria hechas por el procesador, tanto para datos como para instrucciones, tienden a agruparse. Por ejemplo, los programas contienen ciclos iterativos y subrutinas, una vez que uno de ellos se ejecuta, existen referencias repetidas a un pequeño conjunto de instrucciones. De manera similar, las operaciones sobre tablas o arreglas involucran acceso a un conjuto agrupado de datos. Es verdad que a lo largo de una cantidad considerable de tiempo los grupos de datos en uso cambian, pero considerando un periodo de tiempo corto, el procesador trabaja con grupos fijos de referencias a memoria.

Entonces es posible organizar los datos a través de la jerarquía de manera que el porcentaje de accesos a cada nivel subsecuente sea substancialmente menor que el del nivel superior. Considere una jerarquía de dos niveles: el nivel 2 (L2) de memoria contendrá la totalidad de instrucciones y datos. Los grupos de datos/instrucciones que se encuentran en uso se colocan temporalmente en el nivel 1 (L1). En algún punto, un grupo de L1 deberá ser movido a L2 para hacer lugar a un nuevo grupo de instrucciones/datos, pero en promedio, la mayoría de las referencias del procesador será a instrucciones y datos contenidos en L1.

Este principio puede ser aplicado a más de dos niveles de memoria. La memoria más rápida, pequena y cara son los registros internos del procesador. Dos niveles más abajo, la memoria principal es el sistema de

memoria más importante de una computadora. La memoria principal usualmente se extiende utilizando memoria caché de mayor velocidad y menor tamaño. La caché usualmente no es visible al programador, o siquiera, al procesador. Es un dispositivo diseñado para mejorar el desempeño en el movimiento de datos entre la memoria principal y el procesador. Estos 3 tipos de memoria son, típicamente, volátiles y utilizan tecnología semiconductora.

Los datos son guardados de manera más permanente en dispositivos externos, los más comunes son los discos duros y los medios removibles (discos y cintas magnéticos, dispositivos ópticos). La memoria externa, no volátil también se conoce como **memoria secundaria** o **memoria auxiliar.** Se utiliza para guardar programas y archivos de datos y son visibles al programador unicamente en términos de archivos y registros, en lugar de como bytes individuales o palabras. El disco duro también es utilizado para proveer una extensión de la memoria principal conocida como **memoria virtual**.

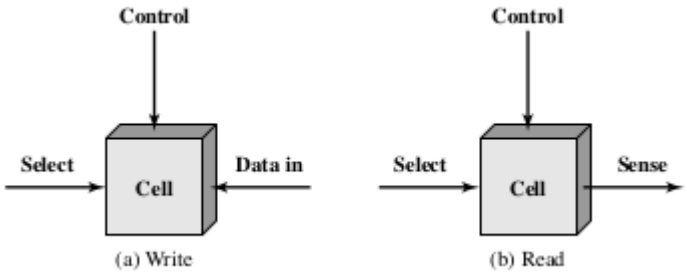
4.2 Memoria Principal Semiconductora

Organización

El elemento básico de la memoria semiconductora es la **celda** o **célula de memoria**. Aunque una variedad de tecnologías son usadas, todas las celdas de memoria semiconductora comparten algunas propiedades:

- Exhiben dos estados estables (o semiestables), que se utilizan para representar el 0 y 1 binarios.
- Son capaces de ser escritas (al menos una vez), para establecer el estado.
- Son capaces de ser leídas para conocer su estado.

Más comunmente, las celdas tienen 3 terminales funcionales capaces de llevar señales eléctricas. La terminal de selección *(select),* como su nombre lo indica, es capaz de seleccionar una celda de memoria para una operación de escritura o lectura. La terminal de control determina cuál de las dos operaciones se va a realizar. Para la escritura, la tercera terminal provee una señal eléctrica que establece el estado de la celda a 1 o 0. Para la lectura, la tercera terminal se utiliza para indicar el estado actual de la celda. Los detalles sobre la organización interna, funcionamiento y temporización de la celda de memoria dependen de la tecnología de los circuitos integrados utilizados para implementar las celdas. Para nuestros objetivos, tendremos por dado que una celda individual puede ser seleccionada para operaciones de lectura y escritura.

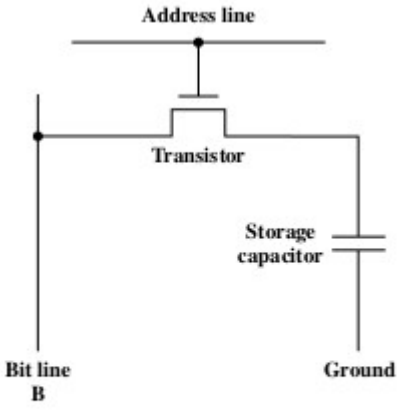


DRAM y SRAM

La siguiente tabla lista los tipos principales de memoria semiconductura. La más común se conoce como memoria de acceso aleatorio o memoria RAM. Éste es un uso incorrecto del término puesto que todos los tipos de memoria listados en la tabla son de acceso aleatorio. Una característica distinguible de la RAM es que es posible tanto leer datos de la memoria, como escribir datos en ella de manera fácil y rápida.

Otra característica de la RAM es que es volátil. Una memoria RAM debe ser proveída con una fuente de poder constante. Si se interrumpe el suministro, entonces los datos se pierden. Así, la RAM puede ser útilizada únicamente para almacenamiento temporal. Dos formas de RAM se tuilizan en las computadoras, la DRAM y la SRAM.

Memory Type	Category	Erasure	Write Mechanism	Volatility
Random-access memory (RAM)	Read-write memory	Electrically, byte-level	Electrically	Volatile
Read-only memory (ROM)	Read-only memory	Not possible	Masks	Nonvolatile
Programmable ROM (PROM)			Electrically	
Erasable PROM (EPROM)	Read-mostly memory	UV light, chip-level		
Electrically Erasable PROM (EEPROM)		Electrically, byte-level		
Flash memory		Electrically, block-level		



(a) Dynamic RAM (DRAM) cell

RAM Dinámica

La DRAM está hecha con celdas que guardan los datos como carga en capacitores. La presencia o ausencia de carga en el capacitor se interpreta como los 1 o 0 binarios. Debido a que los capacitores tienen una tendencia natural a descargarse, las DRAM requieren de una carga periódica que las "refresque" para mantener el almacenamiento de datos. El término *dinámica* se refiere a esta tendencia de la carga a perderse, incluso con una fuente de poder aplicada de manera continua.

En la figura se muestra la estructura de una celda que guarda un bit. La línea de dirección se activa cuando se va a leer o escribir el bit de la celda. El transistor actúa como un interruptor que está cerrado (permite el paso de corriente) si se aplica un voltaje a la línea de dirección; y abierto (no hay paso de corriente) si no hay voltaje presente en la línea de dirección.

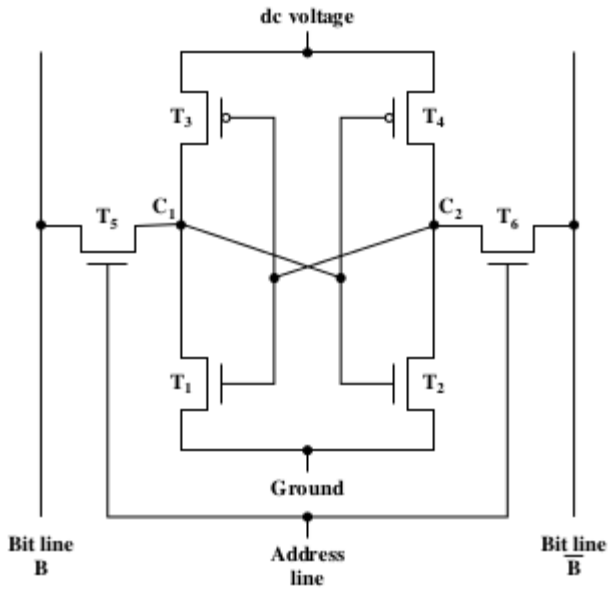
Para escribir, se aplica un voltaje a la línea de bit; un voltaje algo representa un 1 y un voltaje bajo representa un 0. Después se aplica una señal a la línea de dirección que permite que la carga se transfiera al capacitor.

Para leer, cuando se selecciona la línea de dirección, el transmisor se activa y la carga guardada en el capacitor se alimenta hacia la línea de bit que se conecta a un amplificador. El amplificador compara el voltaje del capacitor con un voltaje de referencia y determina si la celda contiene un 1 o un 0. El leer la celda descarga al capacitor, por lo que debe ser restaurado para completar la operación. Aunque una celda DRAM se utiliza para almacenar un bit, esencialmente es un dispositivo analógico. El capacitor puede almacenar cualquier valor de voltaje dentro de un rango así que se útiliza un valor límite de referencia para determinar si se interpreta la carga como un 0 o un 1.

RAM Estática

En contraste, la SRAM es un dispositivo digital que utiliza los mismos elementos lógicos que se utilizan en el procesador. Los valores binarios se almacenan usando configuraciones tradicionales de compuertas lógicas *flip-flop*. Una SRAM mantendrá sus datos mientras tenga alimentación de poder.

Se utilizan cuatro transistores (T1, T2, T3 y T4) conectados de forma cruzada en un arreglo que produce un estado lógico estable. En el estado lógico 1, el punto



(b) Static RAM (SRAM) cell

C1 es alto y el punto C2 es bajo; en este estado T1 y T4 se encuentran apagados mientras que T2 y T3 están prendidos. En el estado lógico 0, C1 es bajo y C2 es alto; en este estado T1 y T4 está prendidos mientras que T2 y T3 están apagados. Ambos estados son estables mientras exista un suministro de corriente directa. Contrario a DRAM, no se requiere "refrescar" la memoria para mantener los datos.

Como en DRAM, la línea de direccción se utiliza para abrir o cerrar un interruptor. Dicha línea controla dos transistores (T5 y T6). Cuando se aplica una señal a la línea de direcciones, ambos transistores se activan, permitiendo una operación de lectura o escritura. Para la escritura, el bit deseado se suministra a la línea B, mientras que su complemento se aplica a la línea B-negada, esto forza a los 4 transistores al estado deseado. Para la lectura, el valor del bit se lee de la línea B.

SRAM vs DRAM

Ambas son volátiles, es decir, se debe suministrar energía de manera constante a la memoria para preservar los valores almacenados. Una célula dinámica es mucho más simple y pequeña que una célula estática. Así, una DRAM es más densa y menos cara que una SRAM correspondiente. Por otro lado, una DRAM requiere el soporte de circuitería de refresco. Para memorias más grandes, el costo fijo de una circuitería de refresco es compensado de sobremanera por el costo bajo de las células DRAM. Debido a lo anterior, las DRAM se utilizan cuando se tienen requerimientos grandes de memoria. Un punto adicional es que las SRAM son más rápidas que las DRAM. Debido a estas características, SRAM se utiliza para implementar la memoria caché y DRAM se utiliza para memoria principal.

Tipos de ROM

La memoria de sólo lectura o ROM *(Read-Only Memory)* contiene un patrón permanente de datos que no puede ser cambiado. Una ROM es no volátil, es decir, no requiere de alimentación de energía para mantener sus bits en memoria. Es posible leer una ROM, pero es imposible escribirle nuevos datos. Una aplicación importante de las memorias ROM es la microprogramación. Otras aplicaciones incluyen:

- Subrutinas para funciones usadas frecuentemente
- Programas de sistema y tablas de funciones.

Una ROM se crea como cualquier otro circuito integrado, con la diferencia de que los datos se incluyen dentro del chip como parte del proceso de fabricación. Lo anterior presenta dos problemas:

- El paso de inserción de datos incluye un costo fijo grande, independientemente de si se va a fabricar una o miles de copias de la ROM.
- No hay cabida para errores. Si un bit está mal, todas las copias de la ROM deben ser descartadas.

Cuando se requiere únicamente un pequeño número de ROM's con un contenido en particular, existe una alternativa menos costosa: las **ROM programables** (**PROM**, *programmable read-only memory*). Las PROM también son no volátiles, sin embargo, pueden ser escritas, aunque sólo una vez. El proceso de escritura es hecho de manera eléctrica y puede ser realizado en un tiempo posterior al de fabricación. Se requiere de equipo especial para efectuar este proceso de escritura. Las PROM's proveen de flexibilidad y conveniencia, sin embargo, las ROM's siguen vigentes para producciones de grandes volúmenes.

Otra variación de las memorias de sólo lectura, son las memorias de mayoría de lecturas, las cuales son útiles para aplicaciones en las cuales las operaciones de lectura son mucho más frecuentes que las operaciones de escritura pero para las cuales se requiere de almacenamiento no volátil. Existen tres formas comunes: EPROM, EEPROM y Flash.

Las **memorias de sólo lectura borrables y programables** (**EPROM**, *erasable programmable read-only memory*) se leen y escriben eléctricamente como las PROM. Sin embargo, antes de una operación de escritura, todas las celdas de almacenamiento se borran y se dejan en el mismo estado inicial mediante la exposición del chip a radiación ultravioleta. El borrado se realiza utilizando una luz ultravioleta intensa a travez de una pequeña ventana incluída en el chip. El proceso de borrado puede ser repetido varias veces, cada borrado puede tomar hasta 20 minutos. Para cantidades comparables de almacenamiento, las EPROM son más caras que las PROM, pero tienen la ventaja de ser regrabables.

Una forma más atractiva de memoria de mayoría de lecturas son las **memorias de sólo lectura**

programables y borrables eléctricamente (**EEPROM**, *electricaly erasable programmable read-only memory*) Estas memorias pueden ser escritas en cualquier momento sin borrar los contenidos previos. La operación de escritura toma considerablemente más tiempo que la operación de escritura, en el orden de los varios cientos de microsegundos por byte. La EEPROM combina la ventaja de la no volatilidad con la flexibilidad de ser actualizable utilizando las líneas ordinarias de control, direcciones y datos del bus. Las EEPROM son más caras que las EPROM y también menos densas, por lo que soportan menos bits por chip.

Una última forma de memoria semiconductora son las memorias **Flash**, llamadas así por la velocidad a la que pueden ser reprogramadas. Introducidas a mediados de los 80's, las memorias flash son el intermediario entre las EPROM y las EEPROM tanto en consto como en funcionalidad. Utilizan borrados eléctricos como las EEPROM. Una memoria flash completa puede ser borrada en unos cuantos segundos, lo cual es mucho más rápido que las EPROM. Además, es posible borrar únicamente algunos bloques de memoria en lugar del chip entero. Sin embargo, las memorias flash no proveen borrado a nivel de byte. Utilizan únicamente un transistor por bit (como las EPROM), con lo cual logran una gran densidad.

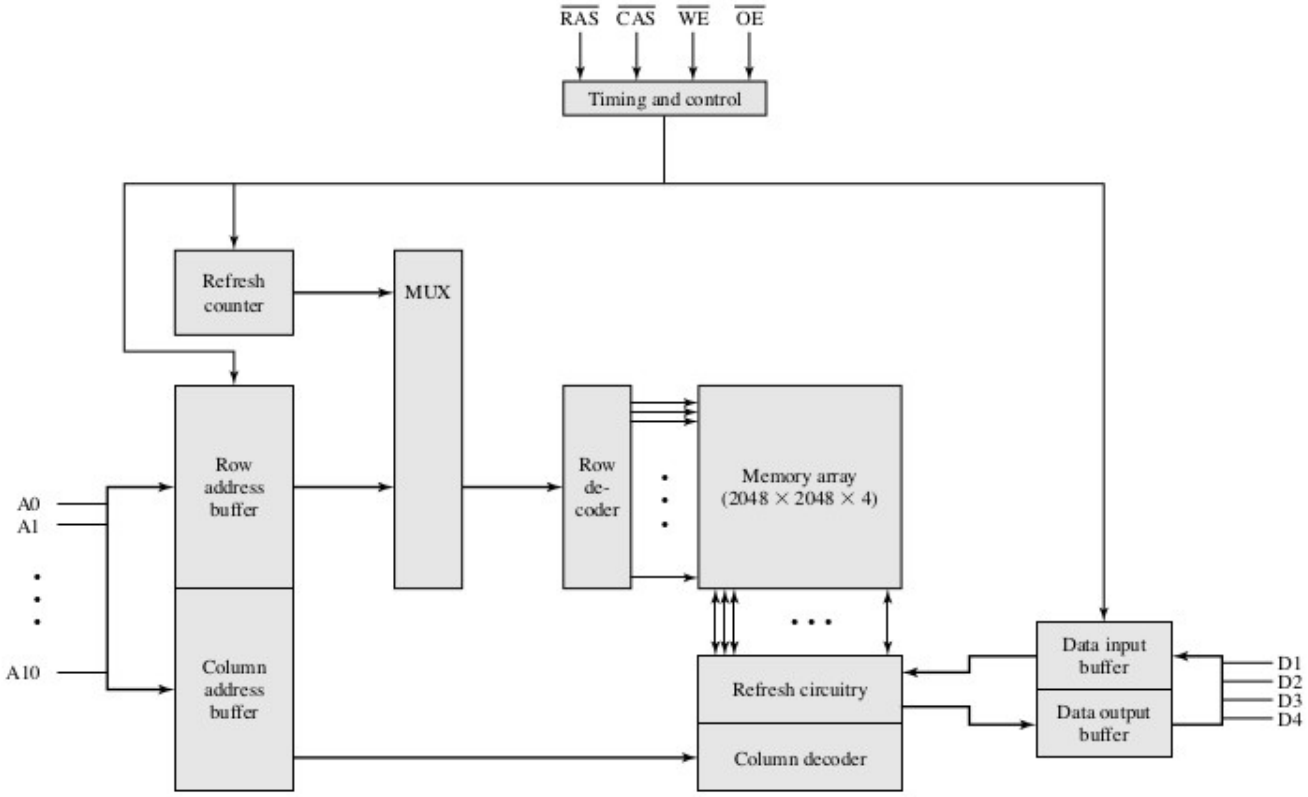
Lógica de circuito *(chip logic)*

Como con otros productos hechos de circuitos integrados, las memorias semiconductoras están empacadas en chips. Cada chip contiene un arreglo de celdas de memoria. La organización de estas celdas de memoria en el chip conllevan ciertos intercambios entre velocidad, capacidad y costo. Para las memorias semiconductoras, uno de los aspectos clave de diseño es el numero de bits que se pueden leer/escribir de una sola vez.

En un extremo, se tiene una organización en la cual el arreglo físico de las celdas sea igual al arreglo lógico de las palabras en memoria. El arreglo está organizado en W palabras de B bits cada una. Por ejemplo, un chip de 16 Mbits puede estar organizado como 1M palabras de 16 bits. En el otro extremo, existe la organización conocida como 1 bit por chip, en la cual los datos se leen/escriben un bit a la vez.

En la figura siguiente se muestra la organizaciño típica de una DRAM de 16 Mbits. Para este caso, 4 bits se leen o se escriben simultáneamente. Se pueden cablear varios arreglos físicos, pero en todos los elementos del arreglo estarán conectados tanto a líneas horizontales (filas) como líneas verticales (columnas). Cada línea horizontal se conecta con la terminal de Selección *(address line)* de cada celda en su fila; cada línea vertical se conecta con la terminal de datos/sensado *(bit line)* de cada celda en su columna. Las líneas de datos proveen la dirección de la palabra a ser seleccionada. Se requieren un total de $log2\ (W)$ líneas. En el ejemplo, 11 líneas de dirección se requieren para seleccionar una de las 2048 filas. Estas 11 lineas se conectan a un decodificador de fila, que tiene 11 líneas de entrada y 2048 líneas de salida. El decodificador activa una única línea de salida de las 2048 posibles dependiendo del patrón de bits presente en las 11 líneas de entrada.

Once líneas adicionales de dirección seleccionan una de las 20148 columnas. Cuatro líneas de datos son utilizadas para la entrada y salida de 4 bits hacia y desde un buffer de datos.

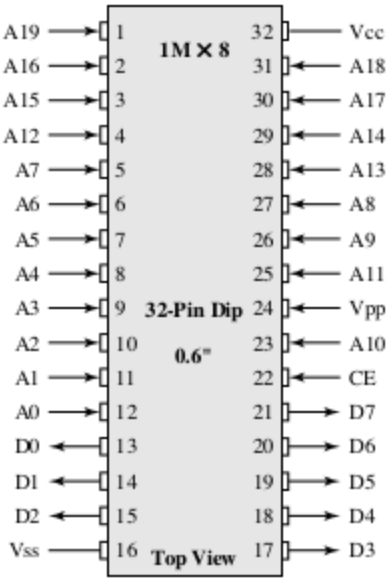


Para la escritura, cada línea de bit se activa con un 1 o un 0 de acuerdo al valor correspondiente presente en las líneas de datos; para la lectura, el valor de cada línea de bit se pasa a través de un amplificador y se presenta a las líneas de datos. Como sólo se leen/escriben 4 bits en esta DRAM, deben existir múltiples DRAM's conectadas al controlador de memoria para poder leer/escribir una palabra completa de datos del bus.

Note que únicamente hay 11 líneas de dirección (A0 – A10) en lugar de las 22 necesarias. Ésto se hace para ahorrar espacio, las 22 líneas se multiplexan en las 11 líneas. Primero, 11 señales de dirección se le pasan al chip para definir la dirección de la fila, posteriormente otras 11 señales se presentan para la dirección de la columna. Esas señales se acompañan por una de dos señales de control que proveen la temporización: la Selección de Dirección de Fila (RAS, *row address select*) y la Selección de Dirección de Columna (CAS, *column address select*). Las señales de control WE *(write enable,* escritura*)* y OE *(output enable,* lectura*)* determinan el tipo de operación.

Como nota adicional, el multiplexado de las direcciones más el uso de arreglos cuadrados resulta en la cuadruplicación del tamaño de la memoria para cada nueva generación de chips de memoria. Si se dedica un nuevo pin al direccionamiento se duplica el número de filas y de columnas y, así, el tamaño del chip de memoria crece por un factor de 4.

En la figura también se indica el uso de circuitería de refresco. Todas las DRAM requieren de una operación de refresco. Una técnica simple es desactivar el chip DRAM mientras se refrescan las celdas. El contador de refresco (*refresh counter*) pasa por todos los posibles valores de fila. Para cada fila, las líneas de salida del contador se proveen al decodificador de fila y se activa RAS. Los datos se leen y se reescriben a la misma localidad. Ésto ocasiona que cada celda de la fila se refresque.



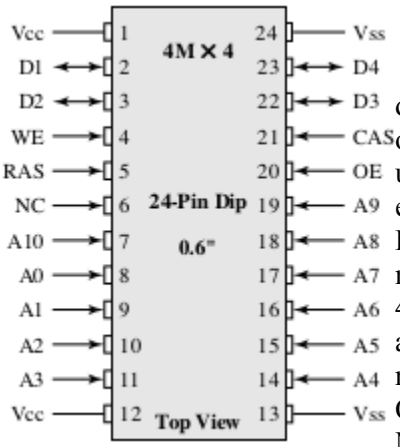
(a) 8-Mbit EPROM

Empaquetado del chip

En la figura siguiente se muestra un ejemplo del empaquetado de una memori EPROM. La memoria presentada es un chip de 8-Mbit organizado como 1M palabras de 8 bits. En este caso, la organización es de una-palabra-

por-chip. El empaquetado contiene 32 pins:

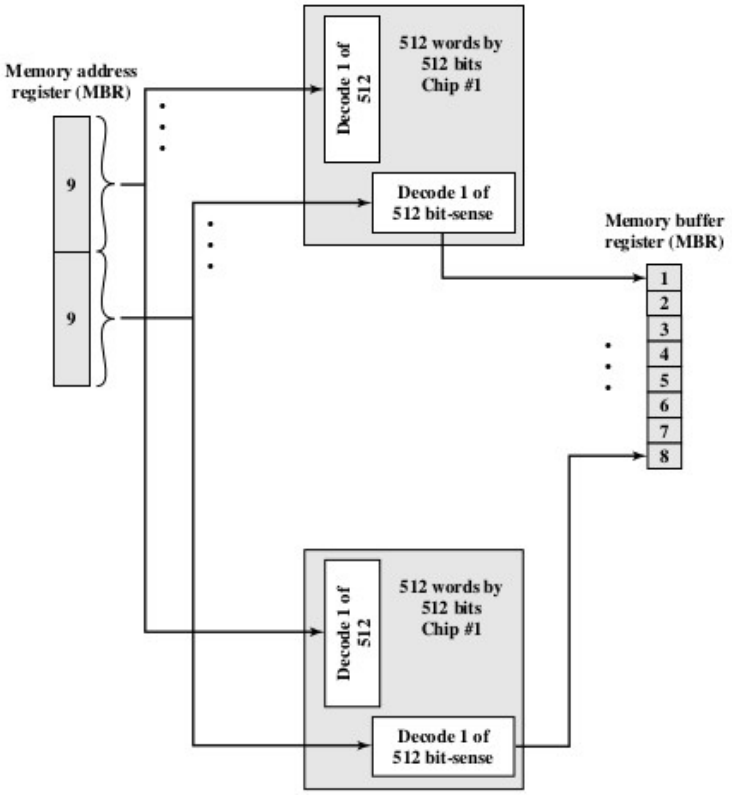
- 20 pins para la dirección de la palabra accesada (A0-A19). *2^20 = 1M*.
- 8 pins para los bits leídos (D0-D7).
- El pin para el suministro de voltaje (Vcc).
- El pin de aterrizaje (Vss).
- Un pin de habilitación de chip (CE, *chip enable*). Este pin existe debido a que puede haber más de un chip de memoria conectado al mismo bus de direcciones. El pin CE se utiliza para indicar si la dirección en el bus es válida o no para este chip.
- El pin de voltaje de programación (Vpp) que se provee cuando se realiza una operación de escritura.



Se muestra en la figura a la izquierda la configuración típica de los pins de una memoria DRAM. La memoria ejemplificada es chip de 16-Mbit organizada 4M x 4. Existen varias diferencias respecto a un chip ROM. Como una memoria RAM puede ser modificada, los pins de datos son de entrada/salida. Los pins WE y OE indican la operación a realizar. Como la DRAM se accesa indicando una fila y una columna, y la dirección se multiplexa, únicamente se necesitan 11 pins de direcciones para especificar las 4M combinaciones de fila/columna. Las funciones de RAS y CAS se discutió anteriormente. Finalmente, un pin de No Conexión (NC) para que exita un número par de pins.

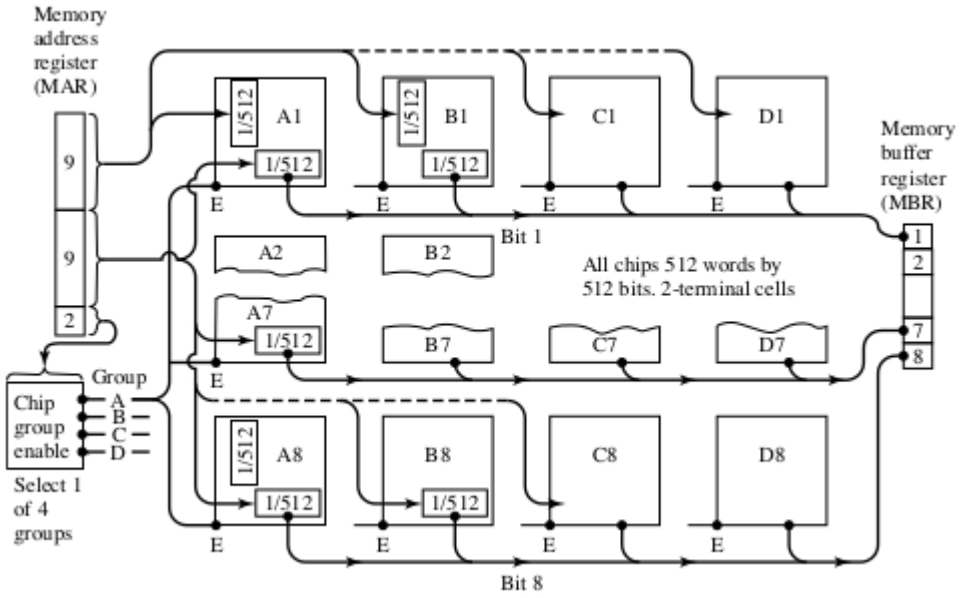
Organización Modular

Si un chip de memoria únicamente contiene 1 bit por cada palabra, es decir, se escriben/leen únicamente 1 bit por operación, entonces claramente se necestan al menos una cantidad igual de chips de memoria al número de bits en la palabra. Por ejemplo, la figura a la derecha muestra cómo se puede organizar un módulo de memoria que consiste de 256K palabras de 8 bits cada una. Para 256K palabras, se necesita una dirección de 18 bits. La dirección se presenta a 8 chips de 256K x 1, cada uno de los cuales provee una entrada/salida de 1 bit.



Esta organización funciona mientras el tamaño de la memoria sea igual al número de bits por chip. En el caso de que se requiera una memoria más grande, es necesario un arreglo de chips.

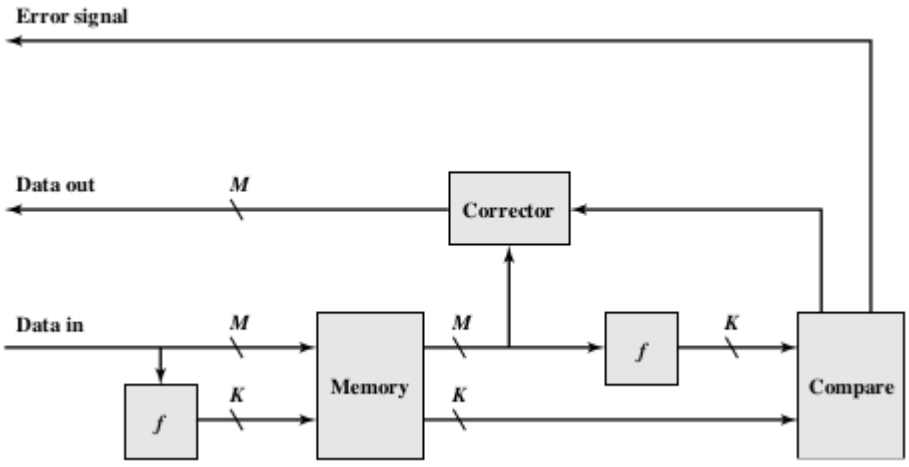
La figura siguiente muestra una posible organización de una memoria que consiste de 1M palabras de 8 bits cada una. En este caso, se tienen 4 columnas de chips cada comlumna contiene 256K palabras organizadas como en la figura anterior. Para 1M palabras se requieren 20 líneas de dirección. Los 18 bits menos significativos se conectan a los 32 módulos. Los 2 bits más significativos se utilizan para seleccionar el grupo o columna.



4.2 Corrección de errores

Una memoria semiconductora está sujeta a errores. Estos se pueden categorizar en errores **duros** y errores **suaves**. Los **errores duros** son un defecto físico permanente que ocasiona que las celdas de memoria no pueden almacenar datos de manera confiable y se atoran en 0 o 1 o cambian de manera errática entre estos dos valores. Los errores duros pueden ser causados por condiciones ambientales poco favorables, abuso de los componentes físicos, defectos de producción y uso. Un **error suave** es un evento aleatorio, no destructivo que altera los contenidos de una o varias celdas de memoria sin dañar los componentes físicos. Estos errores pueden ser causados por problemas en la fuente de alimentación o partículas alpha. Estas partículas resultan de la descomposición radioactiva y son bastante comunes porque los núcleos radioactivos se pueden encontrar en pequeñas cantidades en casi todos los materiales. Tanto los errores duros como los suaves son claramente indeseables, por tanto, la mayoría de los sistemas modernos de memoria incluyen lógica para detectar y corregir errores.

La figura siguiente ilustra, en términos generales, cómo se efectúa el proceso. Cuando los datos se guardan en la memoria, se efectúa un cálculo *f,* y se produce un código. Tanto el código como los datos son almacenados. Así, si una palabra de M bits se va a almacenar y la longitud del código son K bits, entonces la cantidad real de bits almacenados es M + K bits.



Cuando se lee una palabra previamente almacenada en memoria, el código se utiliza para detectar y posiblemente corregir errores. Un

nuevo conjunto de K bits de código se genera a partir de los M bits de datos y se compara con el código previamente almacenado. La comparación puede tener 3 posibles resultados:

- No se detectan errores. Los bits de datos se envían al bus de datos.
- Se detecta un error y es posible corregir el error. Los bits de datos más los bits de corrección de error se envían a un corrector, el cuál produce un conjunto corregido de M bits que son enviados al bus de datos.
- Se detecta un error, pero no es posible corregirlo. Se reporta la condición.

Estos códigos de denominan códigos de corrección de errores. Un código se caracteriza por la cantidad de errores en una palabra que es capaz de corregir y detectar.

El código corrector de errores más simple es el código de Hamming. En la figura a la izquierda se utilizan diagramas de Venn para ilustrar el uso de éste código en palabras de 4 bits (M = 4). Con 3 círculos intersectándose existen 7 compartimentos. Se asignan los 4 bits de datos a los compartimentos internos (a). El resto de los compartimentos se llenan con bits de manera que el número total de 1's en el círculo sea par, a estos bits se les llama *bits de paridad* (b). Así, como el círculo A incluye tres bits de datos con un 1, el bit de paridad en dicho círculo se pone en 1. Ahora, si un error cambia el valor de uno de los bits de datos (c), se encuentra fácilmente. Checando los bits de paridad, se encuentran discrepancias en los círculos A y C pero no en B. El error puede ser entonces corregido cambiando dicho bit.

Ahora desarrollaremos un código para detectar y corregir errores de 1 bit para palabras de 8 bits. Primero, es necesario determinar qué tan grande debe ser el código. La lógica de comparación de códigos recibe dos códigos de K bits cada uno. Se efectúa una comparación bit por bit obteniendo el XOR de estos códigos. Al resultado se le llama *síndrome*. Así, cada bit del síndrome es un 0 o un 1 dependiendo de si existe o no una coincidencia en cada posición de bit para las 2 entradas.

El síndrome tiene también entonces K bits y tiene un rango de valores entre 0 y $2^k - 1$. El valor 0 indica que no existe un error, lo cual deja el resto de los valores para indicar que sí hubo un error. Debido a que el error puede ocurrir en cualquiera de los M bits de datos o los K bits del código se debe cumplir la desigualdad:

$$2^k - 1 >= M + K$$

Esta desigualdad determina el número de bits necesarios para corregir un error de 1 bit en una palabra que contiene M bits de datos, por ejemplo para 8 bits de datos:

- K = 3, 2^3 - 1 < 8 + 3
- K = 4, 2^4 – 1 > 8 + 4

Por lo tanto se requieren 4 bits de código para 8 bits de datos. Las primeras 3 columnas de la tabla siguiente muestran la cantidad de bits de código necesarios para diversas longitudes de palabra.

	Single-Error Correction		Single-Error Correction/ Double-Error Detection	
Data Bits	**Check Bits**	**% Increase**	**Check Bits**	**% Increase**
8	4	50	5	62.5
16	5	31.25	6	37.5
32	6	18.75	7	21.875
64	7	10.94	8	12.5
128	8	6.25	9	7.03
256	9	3.52	10	3.91

Por conveniencia, nos gustaría generar un código con las siguientes características:

- Si el síndrome contiene únicamente 0's, no se ha detectado un error.
- Si el síndrome contiene únicamente un 1, entonces ha ocurrido un error en uno de los bits de código. No es necesesaria una corrección.
- Si el síndrome contiene más de un bit en 1, entonces el valor númerico del síndrome indica la posición del error en los bits de datos. Para corregir el error se invierte el bit en dicha posición.

Para lograr las características anteriores, los bits de datos y los bits de códigos se arreglan en una palabra de 12 bits (8 + 4) como la que se muestra a continuación. Las posiciones de los bits se han numerado del 1 al 12. Aquellas posiciones que sean potencias de 2 se designan como bits de código. Los bits de código se calculan efectuando operaciones XOR sobre los valores de los bits de datos cuyas posiciones contengan un 1 en la misma posición de bit que la posición del bit de código. Así, los bits en las posiciones 3, 5, 7, 9 y 11 (D1, D2, D4, D5, D7) todos contienen un 1 en el bit menos significativo al igual que C1; los bits en las posicioens 3, 6, 7, 10 y 11 todos contienen un 1 en la segunda posición de bit al igual que C2; y así consecutivamente.

$$
\begin{aligned}
C1 &= D1 \oplus D2 \oplus D4 \oplus D5 \oplus D7 \\
C2 &= D1 \oplus D3 \oplus D4 \oplus D6 \oplus D7 \\
C4 &= D2 \oplus D3 \oplus D4 \oplus D8 \\
C8 &= D5 \oplus D6 \oplus D7 \oplus D8
\end{aligned}
$$

Bit position	12	11	10	9	8	7	6	5	4	3	2	1
Position number	1100	1011	1010	1001	1000	0111	0110	0101	0100	0011	0010	0001
Data bit	D8	D7	D6	D5		D4	D3	D2		D1		
Check bit					C8				C4		C2	C1

Visto de otra manera, la posición de bit *n* es revisada por aquellos bits *Ci* de manera que la sumatoria de *i* = *n*. Por ejemplo, la posición 7 se revisa por los bits en las posiciones 4, 2 y 1 (4 + 2 + 1 = 7).

Revisemos que el esquema es adecuado mediante un ejemplo. Asuma que la palabra de 8 bits es 00111001, con el bit de datos D1 en la posición de más a la derecha. Los cálculos son:

$$
\begin{aligned}
C1 &= 1 \oplus 0 \oplus 1 \oplus 1 \oplus 0 = 1 \\
C2 &= 1 \oplus 0 \oplus 1 \oplus 1 \oplus 0 = 1 \\
C4 &= 0 \oplus 0 \oplus 1 \oplus 0 = 1 \\
C8 &= 1 \oplus 1 \oplus 0 \oplus 0 = 0
\end{aligned}
$$

Suponga ahora que el bit 3 de datos tienen un error y su valor cambia de 0 a 1. Cuando se recalculan los bits de código se obtiene:

$$
\begin{aligned}
C1 &= 1 \oplus 0 \oplus 1 \oplus 1 \oplus 0 = 1 \\
C2 &= 1 \oplus 1 \oplus 1 \oplus 1 \oplus 0 = 0 \\
C4 &= 0 \oplus 1 \oplus 1 \oplus 0 = 0 \\
C8 &= 1 \oplus 1 \oplus 0 \oplus 0 = 0
\end{aligned}
$$

Al comparar los nuevos bits de código con los anteriores, se forma el síndrome:

	C8	C4	C2	C1
	0	1	1	1
⊕	0	0	0	1
	0	1	1	0

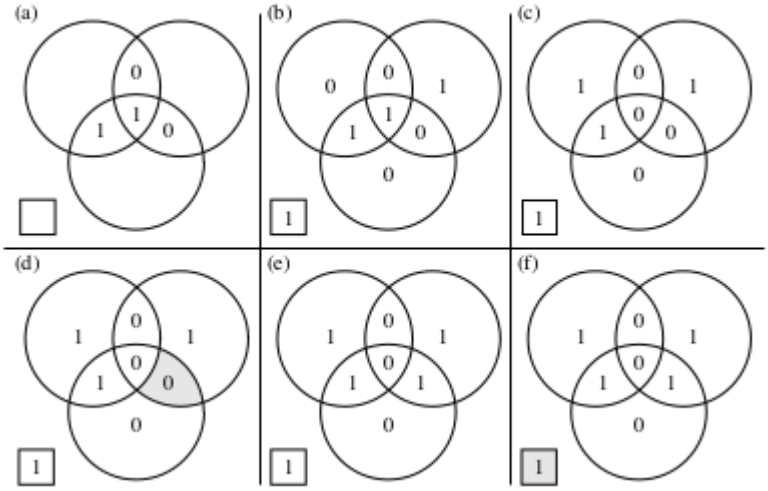
El resultado es 0110, indicando que el bit en la posición 6, que contiene el bit de datos 3, tiene un error.

El c *ecting)*. Más comunmente res únicos y detección de os SEC-DED requieren de

La f e 4 bits. La

Bit position	12	11	10	9	8	7	6	5	4	3	2	1
Position number	1100	1011	1010	1001	1000	0111	0110	0101	0100	0011	0010	0001
Data bit	D8	D7	D6	D5		D4	D3	D2		D1		
Check bit					C8				C4		C2	C1
Word stored as	0	0	1	1	0	1	0	0	1	1	1	1
Word fetched as	0	0	1	1	0	1	1	0	1	1	1	1
Position number	1100	1011	1010	1001	1000	0111	0110	0101	0100	0011	0010	0001
Check bit					0				0		0	1

secuencia muestra que si ocurren dos errores (c), el procedimiento de revisión sale mal (d) y empeora el problema creando un tercer error (e). Para superar el problema, un octavo bit se adhiere y se define de manera que el número total de 1's sea par. Este bit extra de paridad detecta el error (f).



Un código de corrección de errores mejora la confiabilidad de la memoria con el costo de mayor complejidad. Algunos ejemplos son, las implementaciones 30xx de IBM utilizaban un código SEC-DED de 8-bits por cada 64 bits de datos. Así, el tamaño de la memoria principal es en realidad 12% más grande que lo aparente para el usuario. Las computadoras VAX utilizaban un SEC-DED de 7 bits por cada 32 bits de memoria para un 22% de tamaño adicional. Un gran número de DRAM's contemporáneas utilizan 9 bits de código por cada 128 bits de datos para un 7% de tamaño adicional.

4.3 Organización avanzada de DRAM