CAPITULO 2

MEMORIA

2.1. - SUBSISTEMA DE MEMORIA

Este subsistema está constituido por:
i) Dispositivos de almacenamiento capaces de contener instrucciones y datos requeridos por ellas.
ii) Los algoritmos necesarios para el control y manejo de la información almacenada. Estos algoritmos pueden estar implementados por hardware y/o software.

Los distintos dispositivos o componentes de este subsistema pueden ser jerarquizados según el cuadro 2.1, donde se muestra la variación de velocidad/capacidad entre las distintas jerarquías. Las memorias pueden clasificarse o jerarquizarse en base a diferentes atributos.

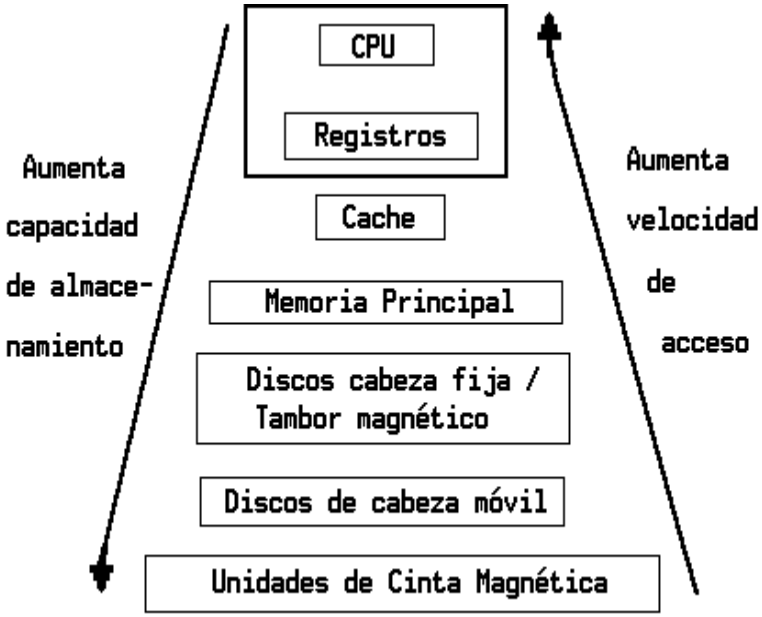

Fig. 2.1. - Jerarquías de memorias.

2.1.1. - Clasificación en base al método de acceso

Esta clasificación tiene en cuenta el orden o la secuencia en la cual la información puede ser accedida. Una memoria es aleatoria si una determinada posición de la misma puede accederse independientemente de la posición accedida con anterioridad; este es el caso de las memorias de semiconductores o de las memorias ferro-magnéticas (de núcleos).

Las memorias de acceso serial o secuencial son aquellas en donde una determinada posición puede ser accedida solamente en cierta secuencia predeterminada; este es el caso de cintas magnéticas o de burbujas magnéticas.

Las memorias semialeatorias o de acceso directo están constituidas por una serie de pistas que pueden accederse aleatoriamente, pero la posición deseada dentro de dichas pistas se accede en forma serial; este es el caso de discos magnéticos (con cabeza fija o móvil) y tambores magnéticos.

2.1.2. - Clasificación en base a velocidad o tiempo de acceso

La memoria interna del procesador comprende un conjunto reducido de registros de alta velocidad usados como registros de trabajo del procesador que almacenan temporariamente instrucciones y datos.

En la memoria principal cada posición puede ser accedida directamente por la CPU.

La memoria secundaria es de mayor tamaño y menor velocidad que la memoria principal, es usada para almacenar programas y archivos de datos que no son continuamente requeridos por la CPU.

La información en memoria secundaria es accedida a través de programas especiales que transfieren la información requerida a la memoria principal.

2.1.3. - Clasificación por la forma de ubicar la información

Las memorias de acceso por dirección operan de la siguiente manera:

La dirección de una posición requerida es transferida desde la CPU mediante el bus de dirección al registro de dirección de memoria, dicha dirección es luego procesada por el decodificador de direcciones, el cual seleccio-

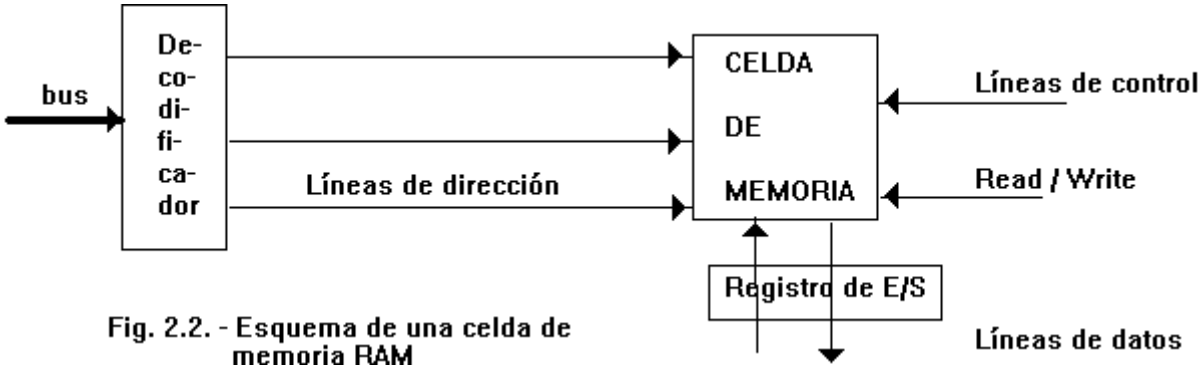

Fig. 2.2. - Esquema de una celda de memoria RAM

na la posición requerida en la memoria.

Las líneas de control de selección de Read o Write especifican el tipo de operación a realizarse. Si un Read es seleccionado, el contenido de la celda es transferido al registro de datos de salida. Si un Write es seleccionado, la palabra a ser escrita que ya se encuentra en el registro de datos de entrada de memoria es transferida a la celda seleccionada.

En las memorias asociativas o direccionables por contenido, el dato almacenado se identifica para su acceso por el contenido, en lugar de identificarse por su dirección. Cada celda de este tipo de memorias tiene capacidad de almacenar información y además circuitos lógicos para equiparar o comparar el contenido almacenado con un argumento externo.

2.1.4. - **Clasificación en base al espacio de direccionamiento**

El espacio de direccionamiento de los programas puede estar contenido en su totalidad en memoria real, o en su defecto puede estar parte en memoria real y parte almacenado en archivos de trabajo sobre algún periférico, que llamamos memoria virtual.
Estos aspectos serán tratados en profundidad en el capítulo de Administración de Memoria.

2.1.5. - **Clasificación en base a la capacidad de modificación de la información almacenada**

Las memorias RAM (Random Access Memory), también conocidas como memorias RWM (Read/Write Memory) son utilizadas para almacenar datos, variables y resultados intermedios que necesitan actualizarse.

Las memorias ROM (Read Only Memory) son utilizadas para almacenar programas y tablas de constantes que no cambian el valor una vez que han sido cargadas en la memoria del computador.

Las memorias PROM (Programmable Read Only Memory) son memorias que pueden ser programadas fuera de línea, es decir, son memorias ROM cuyo contenido puede ser programado fuera del sistema.

Las memorias EPROM (Erasable Programmable Read Only Memory) se diferencian de las anteriores en que además pueden ser reprogramadas, alterándose la programación original insertada en la PROM.

Tanto las memorias RAM, ROM, PROM y EPROM figuran normalmente en la bibliografía como memorias de acceso aleatorio (RAM).

2.1.6.**- Clasificación en base a la perdurabilidad del dato almacenado**

La noción de perdurabilidad de dato almacenado está vinculada a los procesos físicos relacionados con el almacenamiento de la información que en ocasiones suele ser inestable, de ese modo la información puede perderse luego de transcurrido determinado período, a menos que se tomen las acciones apropiadas.

Una memoria cuyo contenido puede ser destruido por una falla de energía se denomina volátil. A esta categoría pertenecen las memorias de semiconductores. Las memorias magnéticas (discos, cintas, etc) no son volátiles.

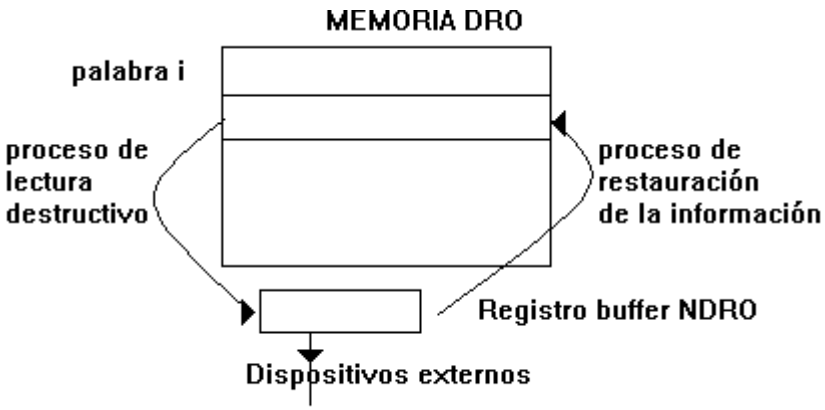

Fig. 2.3. - Ejemplo de lectura en una memoria DRO

Las memorias en las cuales el proceso de lectura altera el contenido de las mismas (por ejemplo, destruyendo la información) se denominan DRO (Destructive Read Out). Un ejemplo de éstas son las memorias con tecnología ferromagnética (núcleos).

Las memorias en las cuales el proceso de lectura no afecta el contenido se denominan NDRO (Non Destructive Read Out). En las memorias DRO cada operación de lectura debe ser seguida por una operación de regrabación que restaura el estado original de la memoria. Esta restauración es llevada a cabo automáticamente usando un registro buffer. La palabra de memoria deseada se transfiere a dicho registro buffer (memoria NDRO) y desde ese registro buffer se transfiere a los dispositivos externos, el contenido de ese registro buffer es automáticamente grabado dentro de la palabra de memoria original.
Ciertos tipos de memoria tienen la propiedad de tender a cambiar su estado con el transcurso del tiempo. Por ejemplo, las primitivas memorias con tecnología MOS (Metal Oxide Semiconductor) tendían a cambiar su estado por la pérdida de carga eléctrica en sus capacitores pasado cierto tiempo. Esto requería un proceso de restauración denominado refreshing. Las memorias que requieren este proceso periódico de restauración se denominan memorias dinámicas (Dynamic Storages). En oposición, las memorias que no requieren restauración se llaman estáticas.

El proceso de restauración en memorias dinámicas puede llevarse a cabo por un camino similar al de las memorias DRO, pero la diferencia estriba en que en estas últimas la restauración se efectúa frente a un proceso de lectura, mientras que en las memorias dinámicas el proceso de restauración es sistemático. Esto significa, en el caso de las memorias dinámicas que transcurrido un cierto período de tiempo (generalmente pequeño) se produce una interrupción al procesamiento que realiza la CPU para realizar este proceso de restauración. Tal proceso se denomina usualmente "Proceso de Refreshing".

2.1.7. - **Clasificación en base al tipo de tecnología**

En cuanto al tipo de tecnología incluiremos solamente un cuadro genérico (Ver Tabla 2.4). El mismo no cuenta con los últimos avances tecnológicos, pero hasta este punto es bastante ejemplificador en cuanto a las grandes diferencias de velocidades que se aprecian entre aquellas que corresponden a medios mecánicos y las que implican tecnología de semiconductores.

Tecnología	Tiempo de acceso	Modo de acceso	Capacidad de modific.	Perdurabilidad	Medio de almacen. físico
Semiconductor Bipolar	30-100 ns/palabra	Aleatorio	R/W	NDRO/ Volátil	Electrón.
Semiconductor Oxido Metálico	0.25-1 μs/2-32 pal	Aleatorio	R/W	DRO/NDRO volátil	Electrón.
Núcleo/Ferromagnética	5-10 μs/2-32 pal.	Aleatorio	R/W	DRO/ No volátil	Magnético
Discos y Tambores magnéticos	5-75 ms/4K	Semiale-atorio	R/W	NDRO/ No volátil	Magnético
Cintas magnéticas	1-5 s/1-16 K	Serial	R/W	NDRO/ No volátil	Magnético
Tarjetas perforadas, cintas de papel perforado	1 s/ tarjeta	Serial	Read	NDRO/ No volátil	Mécanico

TABLA 2.4.

2.2. - **RAM (RANDOM ACCESS MEMORY)**

Un diagrama en bloque de una memoria RAM puede verse en la figura 2.5.

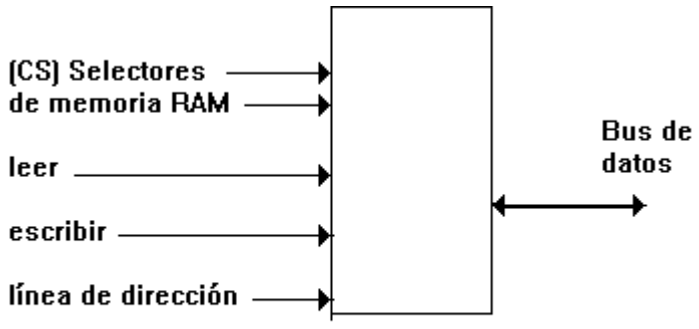

Fig. 2.5. - Una celda de memoria RAM.

Si la capacidad de memoria es de 128 palabras de 8 bits cada una se requiere una línea de dirección de 7 bits.

Los selectores (CS) se usan para seleccionar una de las memorias RAM. En realidad, una memoria RAM se compone de unidades RAM más pequeñas conectadas entre sí, y que se seleccionan mediante estas líneas de selección, es decir, de estar activas las líneas de selección de una de las memorias RAM entonces ésa es la elegida y se extrae o coloca información en dicha RAM específicamente en la posición indicada por la línea de dirección dentro de esa RAM.

Las líneas de leer/escribir pueden ser combinadas en una sola, y se usan para especificar la operación que se realizará sobre la memoria.

El funcionamiento de la RAM puede sintetizarse de la siguiente manera: si la entrada de selección de la RAM correspondiente no está activada, o si está activada pero las entradas de leer/escribir no están activadas, la memoria RAM está inhibida. Cuando la entrada de selección está activada, y la entrada de escribir está habilitada, la memoria RAM almacena un byte obtenido del bus de datos en la localización especificada por la línea de dirección. Si la entrada leer está habilitada, el contenido del byte indicado por la línea de dirección es ubicado en el bus de datos.

2.3. - **ROM (READ ONLY MEMORY)**

Una ROM está organizada internamente en una forma similar a la RAM, pero dado que la ROM solo puede leerse, el bus de datos está en modo salida. Para una ROM del mismo tamaño (en cuanto a dimensiones físicas) de una RAM es posible tener más bits, pues las celdas binarias internas de la ROM ocupan menos espacio que en la RAM ya que las dos líneas leer/escribir de la RAM no existen en la ROM (nótese que al no existir estas líneas se ahorra indudablemente el espacio dentro del circuito impreso que las mismas ocuparían y por ende la superficie libre aumenta), y pueden ser utilizadas para ampliar las líneas de dirección.

Así, una RAM que tenga una dirección de 7 bits es equivalente a una ROM de 512 bytes (9 bits). El funcionamiento de la ROM es similar al de la RAM, pero como en la ROM no hay necesidad de control de lectura/escritura, cuando una ROM es seleccionada por los selectores (CS), el byte indicado en la línea de dirección aparece en el bus de datos.

2.4. - **MEMORIA VIRTUAL**

Para entender un sistema con memoria virtual debemos distinguir los conceptos de espacio de direccionamiento lógico y espacio de direccionamiento físico. Un espacio de direccionamiento lógico es el conjunto de direcciones que aparece en un programa; el conjunto de las direcciones de memoria es el espacio de direccionamiento físico. El espacio lógico puede ser mayor que el espacio físico. Durante la ejecución de un programa, cada dirección lógica es traducida a una dirección física, este mecanismo es conocido como mecanismo de traducción (mapeo) de direcciones. La figura 2.6 muestra las componentes principales de un sistema con memoria virtual y sus conexiones lógicas.

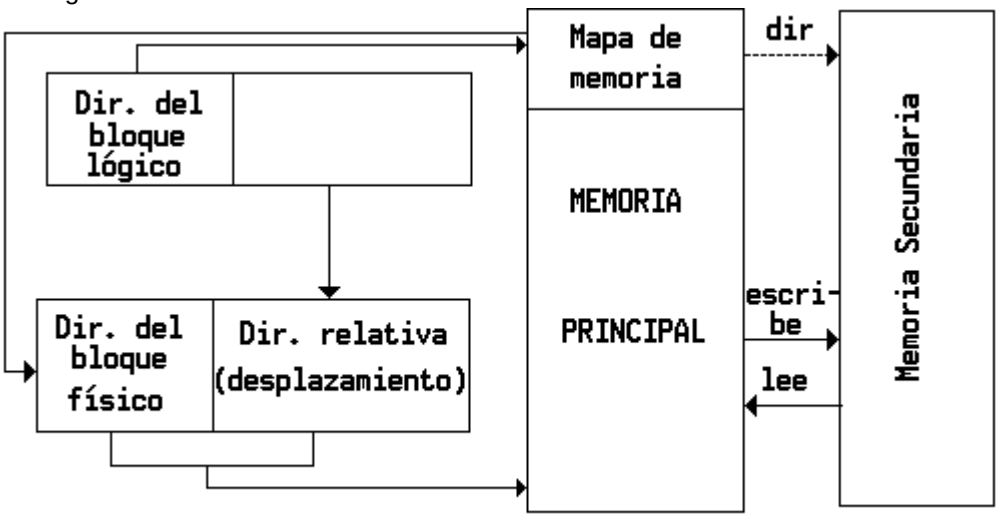

Fig. 2.6. - Componentes de un sistema con memoria virtual.

En sistemas con múltiples procesadores y memoria virtual, el mecanismo de traducción de direcciones es provisto para cada procesador.

Asumamos que el espacio de direcciones generado por el programa J en ejecución sobre un procesador es $V_j = \{0,...,n-1\}$ que consiste de n identificadores únicos. Asumamos que el espacio de memoria asignado al programa J tiene m posiciones, M = {0,...,m-1}, donde cada posición es la identificación de una única dirección de memoria. Como el espacio de memoria asignado puede variar con la ejecución del programa, m está en función del tiempo. Dicha función se define:

$f_j(x,t) =$ y si al instante t, x está en memoria.

$f_j(x,t) =$ $\varnothing$ si al instante t, x no está en memoria.

Cuando $f_j(x,t) =$ vacío se produce una interrupción por falta de direccionamiento que provoca que las rutinas que manejan esas interrupciones requieran el ítem desde el próximo nivel de memoria (si la falta ocurre en memoria principal, se requerirá en memoria secundaria; si la falta ocurre en memoria cache, se requerirá en memoria principal).

2.4.1. - **Implementación del mapa de direcciones**

La función de traducción f puede ser implementada de diversas formas. La implementación más simple es el mapeo directo, que consiste de una tabla de n entradas donde cada entrada contiene la dirección física de memoria y (si f(x) = y), o un carácter nulo. Esta implementación requiere de accesos a memoria adicionales para llegar a la tabla de mapeo, además, las tablas de mapeo pueden ser exageradamente grandes (n posiciones), y la mayoría de sus entradas contendrían caracteres nulos (n-m entradas nulas).

Otra implementación es usar memorias asociativas para efectuar una traducción/mapeo asociativo. Este tipo de memorias asociativas se las conoce como Translation Lookaside Buffer (TLB). La traducción de dirección virtual a dirección real se realiza a través de la búsqueda por contenido (en la TLB se encuentran los pares dirección virtual-dirección física que fueron referenciados en el último lapso).

2.4.2. - **Organización de direcciones en memoria virtual**

Hay 3 métodos de organizar las direcciones en un sistema de Memoria Virtual :

1) Paginado (por demanda): organiza el espacio de direcciones en bloques de tamaño fijo llamados páginas.

2) Segmentación: organiza el espacio de los nombres (identificaciones de posiciones lógicas del programa) en bloques de tamaño arbitrario llamados segmentos.

3) Segmentación con paginado: combina los dos métodos anteriores.

2.4.3. - **Traducción de una dirección virtual a real**

La traducción de una dirección virtual a real puede sintetizarse como se ve en la Fig. 2.7.

Existe un Registro base (PTBR) que apunta a la dirección de comienzo de la tabla de páginas que le corresponde a ese proceso. Con dicha información es que se ingresa originalmente a dicha tabla.

Con el número de página que se desea acceder se utiliza la tabla de distribución de páginas para mapear en qué bloque físico de la memoria real se encuentra dicha página cargada.

Luego reemplazando el número de página por el bloque físico correspondiente y utilizando el desplazamiento provisto en la dirección virtual original se obtiene la dirección exacta de la memoria real a la que se desea acceder.

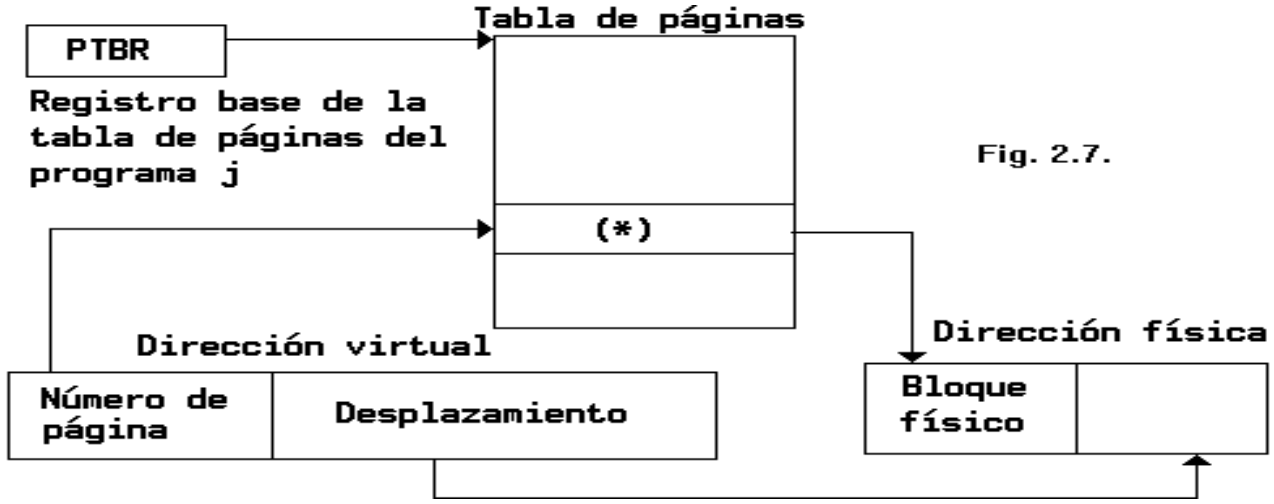

Fig. 2.7.

(*) Contiene información sobre:
- Código de acceso permitido (Read/Write/Execute).
- Bit de validez, indica si la página existe o es nula.
- Bit de página activa: indica si la página está en memoria principal o no.
- Dependiendo de si la página está activa o no:
  i)- Dirección de la página en memoria física.
  ii)- Dirección de la página en almacenamiento secundario.

2.5. - **MEMORIAS DE ALTA VELOCIDAD**

Como se ha visto anteriormente, el empleo de memorias de alta velocidad tiende a compensar la diferencia de bandwidth entre memoria y CPU. Entre los distintos métodos existen, a saber:

- Utilizar memorias de tecnologías rápidas (bipolar). Lamentablemente este método es muy costoso.
- Usar palabras de memoria de gran tamaño.
- Acceder a más de una palabra durante un ciclo de memoria (memorias interleaved).
- Insertar una memoria rápida (cache) entre la CPU y la memoria principal.

Clasificación por	Tipos
Método de acceso	Aleatorias Semialeatorias Secuenciales
Velocidad / Tiempo de acceso	Memoria interna del proc. (cache) Memoria Principal Memoria Secundaria
Según la forma de ubicar la información	Acceso por dirección - RAM Acceso por contenido - CAM
Según el espacio de direccionamiento	Memoria Real Memoria Virtual
Según la capacidad de modificación de la información almacenada	RAM ROM PROM EPROM
Según la perdurabilidad del dato almacenado	DRO (Destructive Read Out) / NDRO Dynamic Storage / Static Volátiles / No volátiles (discos)

TABLA 2.8. - Cuadro resumen de las clasificaciones.

2.5.1. - **MEMORIAS INTERLEAVED**

Este tipo de memorias es generalmente utilizada en entornos de multiprocesamiento, donde la memoria principal es compartida por varios procesadores. La memoria principal es particionada en módulos separados, o

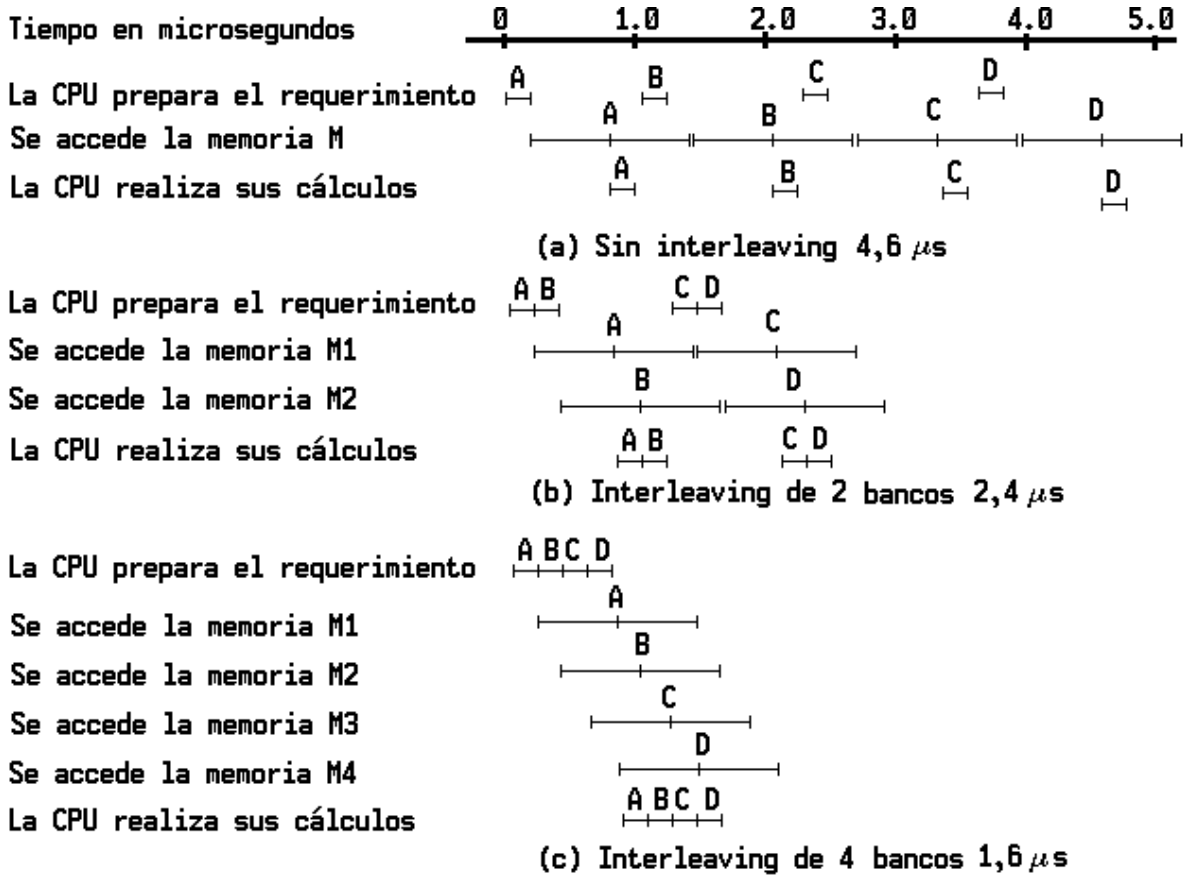

Fig. 2.9. Ejemplos de tiempos de memorias interleaved.

Bancos de Memoria, para permitir accesos simultáneos independientemente en cada módulo. Esta organización permite que una o más palabras puedan ser accedidas en cada ciclo de memoria.

Debe tenerse bien presente que si bien dos módulos de memoria pueden estar siendo accedidos simultáneamente, si existe solo un bus de datos para acceder a memoria la transferencia en sí de lo accedido debe realizarse secuencialmente.

La diferencia entre ciclo de memoria y tiempo de acceso de una memoria estriba en que el tiempo de acceso es el tiempo que le lleva a la memoria enviar o recibir una palabra (o la unidad de transferencia que corresponda) a la CPU, en tanto que el ciclo de memoria es el tiempo que transcurre desde que se hace una referencia a memoria y la misma está nuevamente disponible para poder ser accedida.

Este último tiempo que usualmente es bastante mayor al anterior proviene del hecho de que las memorias (por ser circuitos capacitores) requieren del transcurso de un cierto período de tiempo antes de poder ser nuevamente referenciadas.

Veamos un ejemplo a este respecto.

Sea una memoria cuyo tiempo de acceso es del orden de 0,6 microsegundos y cuyo ciclo es de 1,2 microsegundos, y una CPU que requiere de 0,2 microsegundos para preparar el acceso a memoria y de 0,2 microsegundos más para poder procesar la información obtenida.

En estas condiciones (Ver Fig. 2.9) acceder 4 veces a la memoria si no se utiliza la técnica de interleaving consumirá 4,6 microsegundos. En tanto que si se utiliza una memoria interleaved de 2 bancos se tardará 2,4 microsegundos y si fuera de 4 bancos el tiempo total se reduce a 1,6 microsegundos.

Cada módulo de una memoria interleaved tiene su circuito de direccionamiento. Diferentes procesadores que referencian a distintos módulos pueden acceder simultáneamente a memoria. Para que esta organización funcione eficientemente, las referencias generadas por los distintos procesadores deben ser distribuidas entre los distintos módulos, dado que dos o más referencias sobre el mismo módulo no pueden ser llevadas a cabo simultáneamente.

2.5.1.1. - **INTERCALACIÓN DE DIRECCIONES**

Si conocemos que un procesador va a referenciar a K posibles palabras de memoria ($P_0,...,P_{k-1}$), que pueden ser instrucciones dentro de un programa, y que las mismas serán asignadas a K direcciones consecutivas del direccionamiento físico ($A_0,...,A_{k-1}$), se pueden aplicar las siguientes reglas de intercalación:

1) Asignar la dirección A_i al módulo M_j si j=i [módulo m], siendo m la cantidad de módulos en que fue dividida la memoria. Es conveniente definir el número de módulos m como una potencia de 2, es decir $m = 2^p$; luego los p bits menos significativos de la dirección identifican el módulo en el cual se encuentra esa dirección. (Ver figura 2.10.)

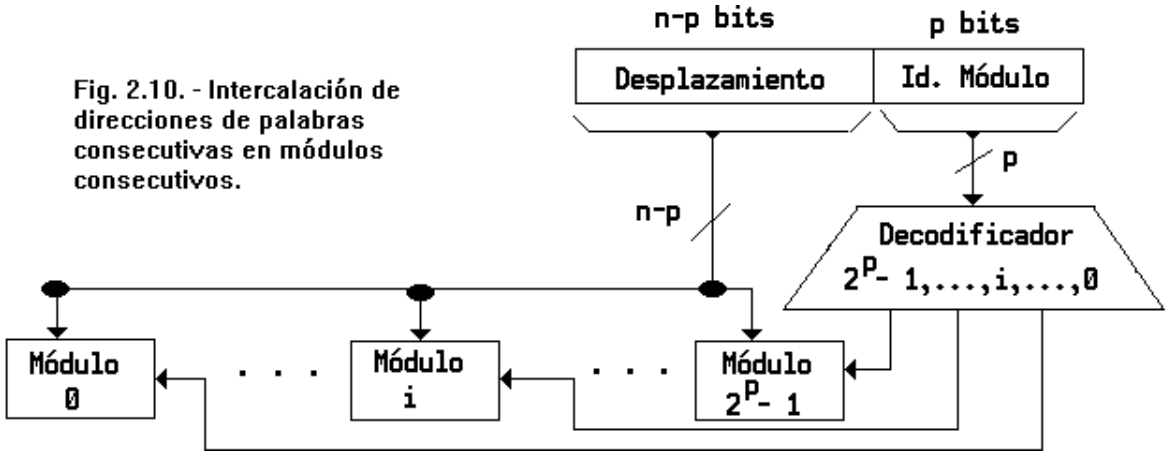

Fig. 2.10. - Intercalación de direcciones de palabras consecutivas en módulos consecutivos.

Este método de intercalación distribuye direcciones consecutivas en módulos consecutivos.

2) Asignar los bits más significativos para seleccionar el módulo, los restantes bits indican el desplazamiento dentro del módulo (Ver figura 2.11). Este esquema facilita la expansión de memoria por la adición de nuevos módulos, pero dado que direcciones contiguas están dentro del mismo módulo, puede causar conflictos en accesos simultáneos con procesadores pipeline, array processors, etc.

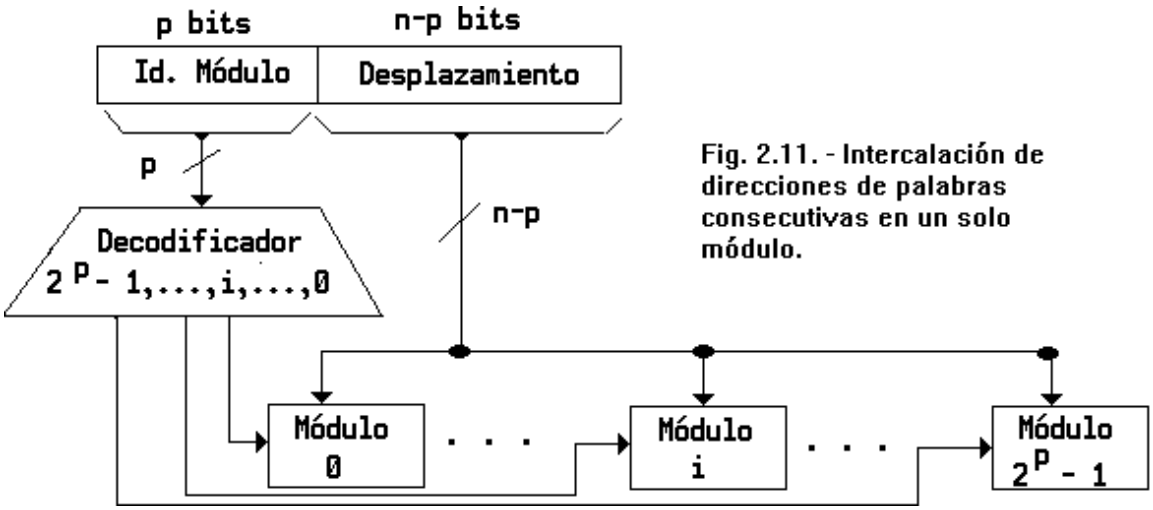

Fig. 2.11. - Intercalación de direcciones de palabras consecutivas en un solo módulo.

Esta técnica de interleaving es usada frecuentemente para incrementar la velocidad con la cual se levantan instrucciones de memoria.

La eficiencia con la cual es usado un sistema de interleaving de memoria es dependiente del orden en que fueron generadas las direcciones de memoria. Este orden es determinado por los programas que serán ejecutados. Si dos o más direcciones requieren accesos simultáneos al mismo módulo, se está en presencia de una **interferencia o contención de memoria**; los accesos a memoria en cuestión no pueden ser ejecutados simultáneamente.

2.5.2. - MEMORIAS MULTIPUERTAS Y CONEXIONES A TRAVÉS DE LINEAS (CROSSBAR SWITCH)

La figura 2.12 muestra una organización crossbar switch en la cual hay un camino (path) disponible para cada unidad de memoria:

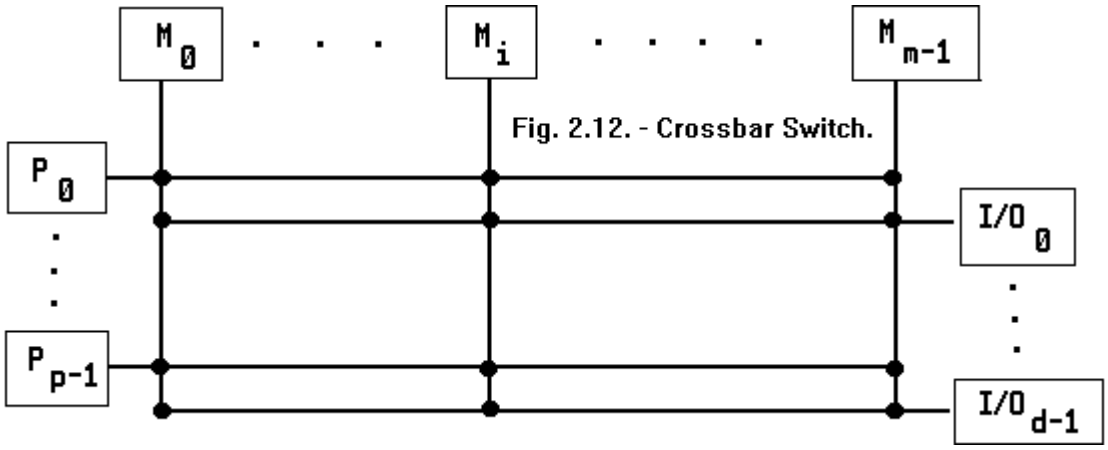

Fig. 2.12. - Crossbar Switch.

El crossbar switch brinda una conexión completa con los módulos de memoria porque existe un "bus" asociado con cada módulo de memoria. El máximo número de transferencias que pueden tomar lugar

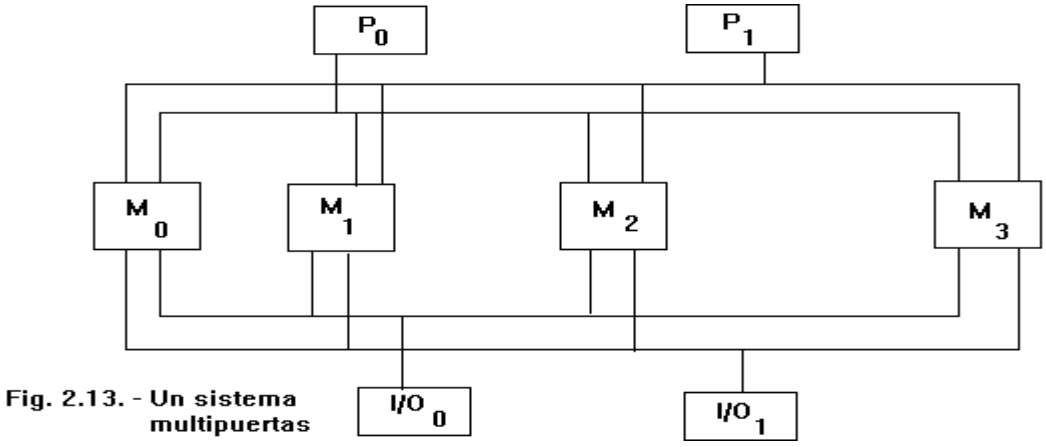

Fig. 2.13. - Un sistema multipuertas

simultáneamente está limitada por el número de módulos de memoria y la relación bandwidth-velocidad de los buses, y no por la cantidad de caminos (paths) disponibles.

Una característica importante de este sistema es la simplicidad de switchear (conmutar) unidades funcionales (es decir alterar en forma dinámica los caminos habilitados) y soportar transferencias simultáneas para todas las unidades de memoria.

Un problema que debe resolver esta organización es el arribo de múltiples requerimientos de acceso al mismo módulo de memoria en un ciclo de memoria. Este conflicto requiere del manejo de prioridades predeterminadas. Si el control y el manejo de la lógica de prioridades, que está distribuida a través de la matriz crossbar switch, se la incluye en las interfases de los módulos de memoria; el resultado es un sistema de memoria multipuerta.

La figura 2.13 muestra un sistema multipuertas. Nótese que a diferencia del gráfico del crossbar antes mencionado en este caso existen diferentes puertas de ingreso a cada elemento funcional y el método que se utiliza para resolver los conflictos de acceso a memoria es asignar prioridades a cada una de dichas puerta de acceso.

2.5.3. - **MEMORIAS ASOCIATIVAS (CAM)**

Las memorias CAM (Content Addressable Memory) son memorias direccionables por contenido, utilizadas en aplicaciones donde el procesamiento de datos requiere una búsqueda de ítems almacenados en memoria.

El dato almacenado se identifica para su acceso por el contenido, en lugar de su dirección. Debido a su organización, este tipo de memorias es accedida simultáneamente en paralelo. Las búsquedas pueden hacerse por la palabra entera o por un campo específico dentro de la palabra. Una memoria asociativa es mucho más cara que una memoria RAM. Cada celda tiene capacidad de almacenamiento y circuitos lógicos para equiparar o comparar el contenido con un argumento externo.

2.5.3.1. - **Organización del Hardware**

Una memoria de m palabras de n bits por palabra requiere un registro de equiparamiento de m bits, uno para cada una de las palabras de memoria. Cada palabra de memoria es comparada en paralelo con la clave de búsqueda que se encuentra en el registro de argumento.

2.5.3.2. **Operación de lectura**

Existe un registro argumento que contiene el dato que se desea ubicar. Además existe un registro clave cuyo fin es enmascarar parte de los bits del argumento a fin de proveer una búsqueda sobre una cantidad menor de bits de las palabras de la memoria asociativa.

Las palabras que equiparan todos sus bits con los de la clave de búsqueda activan el bit asociado a dicha palabra en el registro de equiparamiento. Después del proceso de equiparamiento, todos los bits del registro de equiparamiento que fueron activados indican el hecho de que sus palabras asociadas coinciden con el argumento (en los bits no enmascarados). La lectura se logra por un acceso secuencial a la memoria para todas aquellas palabras cuyos bits correspondientes en el registro de equiparamiento han sido activados.

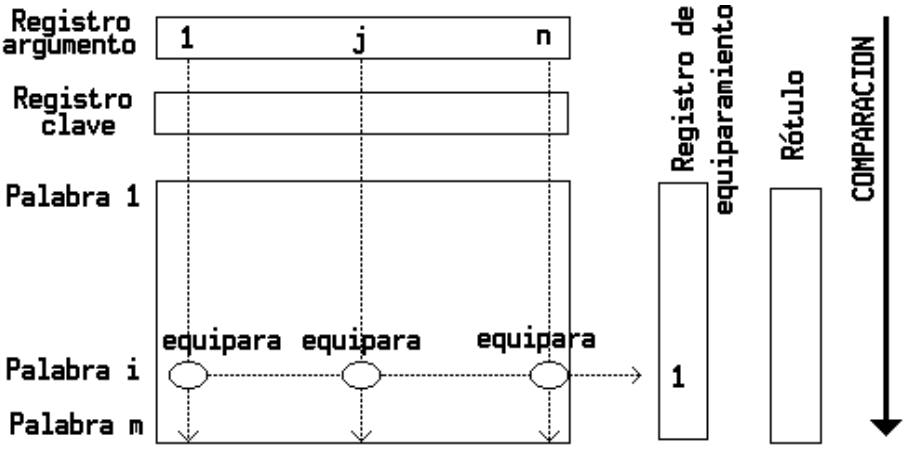

Fig. 2.14. - Lectura de una memoria CAM.

La organización interna de una celda cualquiera C_{ij} consta de un flip-flop de almacenamiento (celda binaria capaz de almacenar un bit de información) y los circuitos para leer, escribir y equiparar. (El circuito para equiparar esta constituido en base a una función booleana con inversores de bits, compuertas AND y OR).

2.5.3.3. - **Operación de escritura**

En una memoria de m palabras existe un registro especial "rótulo" de m bits que tiene por objeto llevar cuenta de las palabras activas e inactivas. Para cada palabra activa en memoria, el bit correspondiente en dicho registro se coloca en 1. Para poder almacenar una nueva palabra en memoria se recorre este registro hasta en-

contrar un bit en cero, esto da la posición de la primer palabra de memoria inactiva disponible. Después de que la nueva palabra fue almacenada se coloca su bit de rótulo en 1.

2.5.3.4 - **Tipos de memorias asociativas.**

Existen dos organizaciones diferentes de memorias asociativas:

- Organización paralela de bits: en esta organización el proceso de comparación es realizado en paralelo por palabras y en paralelo por bits. Es decir la arquitectura de estas memorias permite comparar simultáneamente por palabras completas al mismo tiempo que por franjas de bits-slices.
- Organización serial de bits: esta organización opera con un bit-slice a un tiempo a través de todas las palabras. Cada bit-slice es seleccionado por una unidad de control. Esta organización es usada en búsqueda y recuperación de información no numérica. Requiere menos hardware, pero es más lenta que la organización anterior.

2.5.4. - **MEMORIA CACHE**

Las memorias cache son buffers de alta velocidad insertados entre el procesador y la memoria principal para capturar una porción del contenido de la memoria principal que está actualmente en uso.

El éxito de las memorias cache puede ser atribuido a la propiedad de localidad de referencia. El análisis de un gran número de programas ha demostrado que las referencias a la memoria en un intervalo de tiempo tienden a estar circunscriptas dentro de un área localizada en la memoria. A este fenómeno se lo denomina localidad de referencia. Si las porciones activas del programa y los datos son colocados en una memoria rápida (más pequeña que la memoria principal), el tiempo de acceso promedio a memoria puede reducirse. El tiempo de acceso a memoria cache es menor que el tiempo de acceso a memoria principal en un factor de 5 a 10.

2.5.4.1. - **Cómo opera una Memoria Cache**

Es usual encontrar en las implementaciones que una memoria cache trabaja con una TLB asociada (Translation Lookaside Buffer), aunque puede operar sin ella.

Los guarismos para una procesador VAX-11/780 indican que la información que se necesita acceder se encuentra en la TLB más del 97 % de las veces, evitando el acceso a memoria principal.

Esto es, que el promedio de fracasos (miss) de la TLB es del orden del 3 %.

En la Fig. 2.15 puede verse un esquema genérico de funcionamiento de una cache con su TLB asociada.

El procesador utiliza los bits más significativos de la dirección virtual para obtener la información de la página referenciada desde la TLB. De encontrarse esa página en la TLB se produce una operación exitosa (TLB hit).

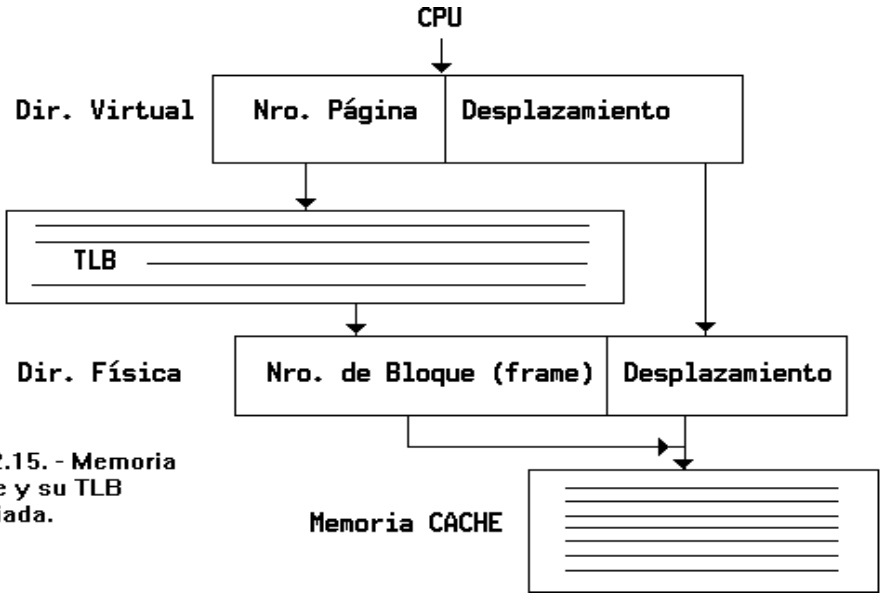

Fig. 2.15. - Memoria cache y su TLB asociada.

De la TLB se toma el número de bloque correspondiente (el bloque de memoria real que realmente ocupa dicha página), el cual se concatena con el desplazamiento original que figuraba en la dirección virtual.

Con esa información se accede entonces a la memoria cache en busca del dato requerido.

En suma la TLB contiene la información de conversión de la dirección de la página virtual al bloque de memoria real, en tanto que en la cache se hallan los pares de valores de direcciones reales y sus respectivos contenidos.

El Directorio de la Cache (CD) provee asimismo una forma más de acotar la búsqueda dentro de la misma cache.

El diagrama de flujo de la Fig. 2.16 indica cómo opera una memoria cache.

Cuando la CPU necesita acceder a la memoria, se examina la memoria cache. Si la palabra se encuentra (cache hit), la información es traída a la CPU desde la memoria cache.

Si la palabra requerida no se encuentra en memoria cache (cache miss), se accede a la memoria principal. El bloque (de 1 a 16 palabras) de la memoria principal que contiene la palabra referenciada es transferido a la memoria cache.

De esta manera, posiblemente, los datos requeridos en referencias futuras se encuentran en memoria cache.

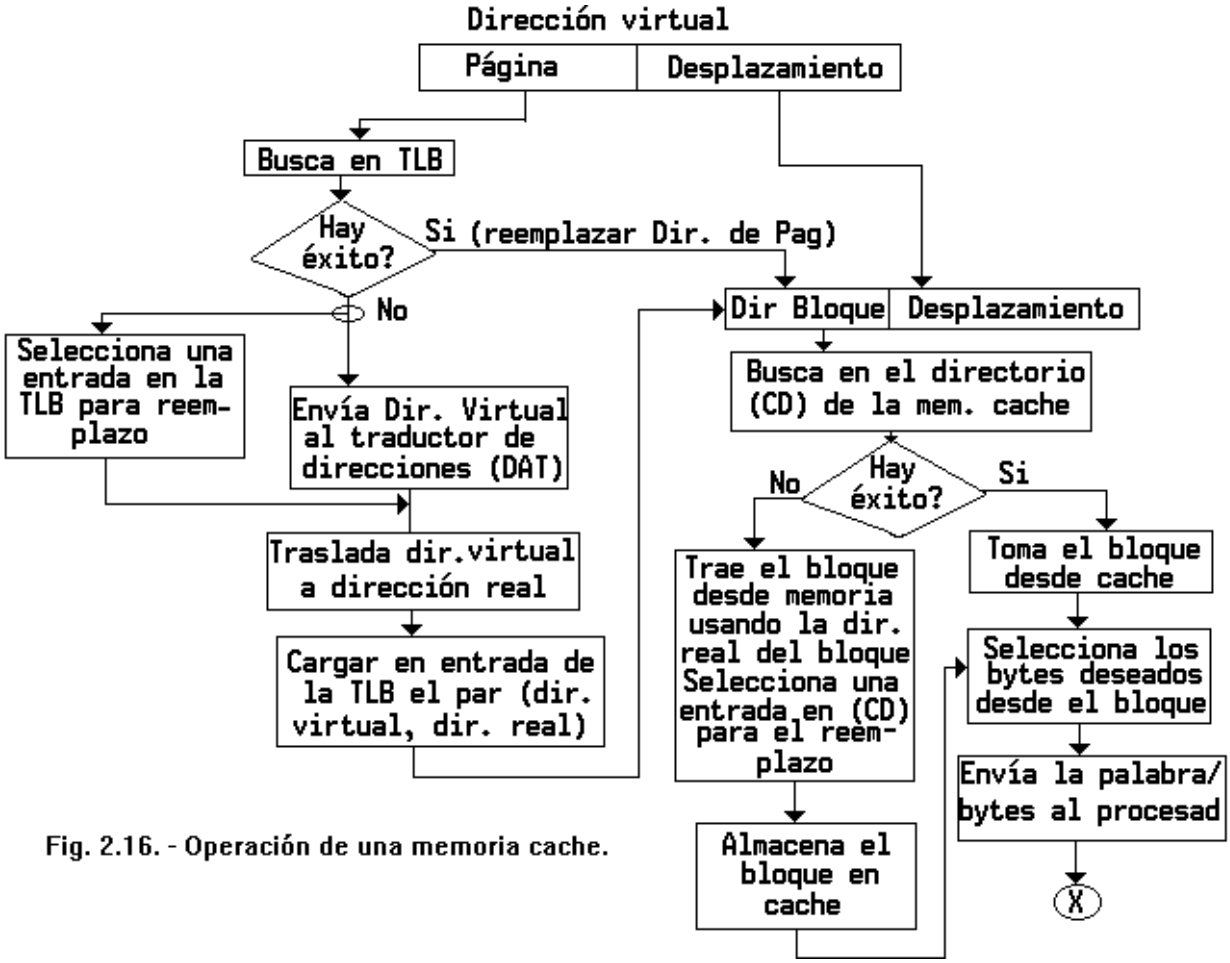

Fig. 2.16. - Operación de una memoria cache.

Las memorias cache generalmente consisten de dos partes:

- El directorio de la memoria cache (CD).
- La memoria de acceso aleatorio (RAM).

El directorio (CD) está comúnmente implementado como una memoria asociativa consistente de direcciones de bloques y bits de información adicional (bit de validez de dato, bit de protección, etc.). La ubicación de datos de la memoria principal a la memoria cache se la conoce como mapeo de datos. Existen distintos tipos de mapeo (o políticas de ubicación), a saber: directo, asociativo y asociativo de conjunto.

2.5.4.2. - **Actualización de la Memoria Cache**

Cuando la CPU encuentra una palabra en la memoria cache y la operación es de lectura, la memoria principal no está involucrada en la transferencia.

Pero si la operación es de grabación, el sistema puede actuar de dos maneras:

- El procesamiento más simple utilizado es actualizar la memoria principal cada vez que se efectúa una operación de grabación en memoria cache. Esto se denomina método de escritura directa. La ventaja del mismo es que la memoria principal siempre contiene los mismos datos que la memoria cache. Este método es importante en sistemas que poseen DMA (Direct Memory Address) puesto que asegura que los datos de la memoria principal son válidos en el instante en que los dispositivos se comunican a través del DMA.
- El segundo procedimiento es denominado método de Reescritura. En este método solamente la palabra de memoria cache es actualizada durante la operación de escritura, pero esa palabra es marcada para indicar que fue modificada y en el momento que se retire de la memoria cache (desalojo), será copiada en memoria principal.

EJERCICIOS

1) Defina qué es :

- memoria serial
- memoria semialeatoria
- memoria interna
- memoria principal
- memoria secundaria
- memoria asociativa
- memoria real
- memoria virtual
- memoria RAM
- memoria ROM
- memoria PROM
- memoria EPROM
- memoria DRO
- memoria NDRO
- memorias dinámicas
- memorias estáticas
- memoria volátil

2) Dibuje esquemáticamente cómo es una celda de una memoria RAM. En qué se diferencia con una memoria ROM ? Indíquelo expresamente en su dibujo.

3) Qué es memoria interleaved y cómo funciona ?

4) En qué esquema de memoria se puede producir lo que se denomina **contención de memoria** ?

5) Explique en qué consiste el mecanismo de refreshing y en qué tipos de memoria es necesario.

6) Explique cómo funciona el direccionamiento en una memoria de tipo asociativa (CAM).

7) Cuál es la diferencia entre ciclo de memoria y tiempo de acceso ?

8) Justifique la utilidad de las memorias interleaved.

9) Dado un programa secuencial con direccionamiento de instrucciones y datos correlativos qué tipo de Intercalación de direcciones conviene ? Justifique.

10) Explique con palabras qué es una memoria cache y cómo opera.

11) Explique en qué situación se produce un "TLB miss" ?

12) Cuáles son los dos métodos que se utilizan normalmente para asegurar la integridad de la información de memoria principal con respecto a la de memoria cache y en qué consisten ?