**Clasificación de Memorias**

* **Según el modo de acceso a la unidad de información**
  + **Aleatorio**: un componente de selección habilita una palabra e inhabilita los demás.
  + **Secuencial**: se establece una posición de referencia; el tiempo de acceso depende de la distancia de la posición inicial a la posición de referencia.
  + **Asociativo**: permiten acceder a un contenido determinado, que se denomina segmento. Son costosas. Se justifican para aplicaciones que requieran buscar conjuntos de elementos.
* **Según las operaciones aceptadas**
  + **Vivas**: aceptan operación de escritura y de lectura.
  + **Muertas**: solo aceptan operación de lectura.
* Según la duración de la información
  + **Volátiles**: la información se pierde al cortar la corriente.
  + **Permanentes o No Volátiles**: la memoria queda almacenada por mas que se pierda la corriente.

La capacidad o dimensión de la memoria representa la cantidad de información que puede almacenarse en ella.

**Memorias RAM (estáticas, dinámicas y con acceso directo)**

Las memorias RAM estáticas (SRAM) son memorias vivas, volátiles y estáticas en la que cada celda es un elemento biestable diseñado con compuertas. Posee un elemento WE (write enable) que establece si esta en modo escritura (1) o lectura (0) y otro elemento EN (enable) que establece el chip habilitado para su acceso. Posee 3 estados posibles: en reposo, lectura o escritura.

Las memorias RAM dinámicas (DRAM) son memorias vivas, volátiles y dinámicas que degradan su información con el paso del tiempo. Son mas lentas que las SRAM pero poseen mayor capacidad. Son de lectura y escritura, se utilizan para almacenar mayor cantidad de bytes en la memoria principal de las PC. Su costo es bajo y tiene una gran capacidad de almacenamiento en relación a las memorias estáticas.

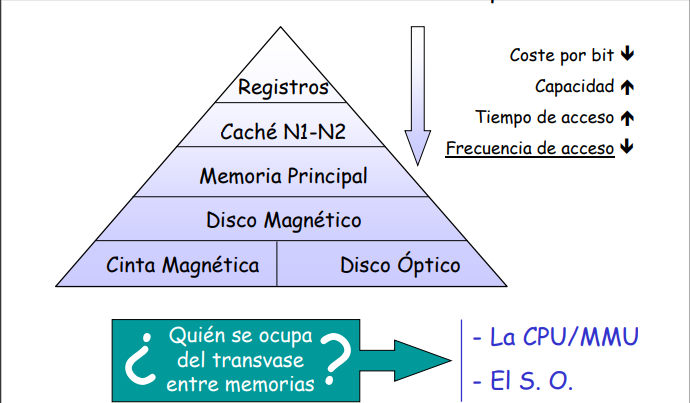
Las memorias RAM de acceso directo están establecidas en forma matricial (filas y columnas) y acceden aleatoriamente a la información. El numero que identifica a la información se denomina dirección física. Es una memoria de tipo asociativa.

**Big Endian y Little Endian**

En big endian el bit – significativo sigue siendo el – significativo y el + significativo sique siendo el + significativo.

En Little endian el bit – significativo pasa a ser el + significativo y viceversa.

**Jerarquía de Memorias**



**Memoria Caché**

La memoria caché es una memoria con muy poca capacidad de almacenamiento utilizada para retener temporalmente ciertos tipos de información.

**¿Por qué es tan útil**? Porque se basa en el principio de la localidad de referencia o vecindad, que dice que los accesos a memoria realizados por la CPU no se encuentran distribuidos por todo el espacio de direccionamiento, sino que se concentran temporalmente en ciertas áreas de memoria. La localidad temporal establece que es muy probable repetir las instrucciones que últimamente fueron utilizadas, y la localidad espacial establece que es muy probable utilizar instrucciones cercanas a las que utilizamos últimamente.

La unidad de transferencia entre la caché y la memoria principal se denomina **bloque**. Su tamaño se expresa en potencias de 2. Y la unidad de transferencia de la caché con la CPU se denomina **palabra**.

El espacio de memoria principal está dividido en **bloques** mientras que el de la caché en **líneas**. El tamaño del bloque es igual al tamaño de la línea, ya que dentro de cada una se almacena 1 bloque.

**¿Cómo es la operación de lectura y escritura?** Se recibe la dirección de la CPU y se analiza si el bloque que contiene a la palabra de esa dirección ya se encuentra en caché. Si el bloque ya se encuentra en caché se extrae la palabra y se entrega a la CPU; si el bloque no está en caché, hay que realizar un paso previo a esto. Primero va a haber que asignar espacio en caché para el bloque de memoria principal, luego, leer y trasladar el bloque a caché. Si el bloque ya se encuentra en la caché se denomina **acierto de caché**, si no lo está, se denomina **falta de caché**. Para colocar el bloque en caché es necesario responder a las **políticas de ubicación**; si queremos escribir en la caché es necesario responder a las **políticas de actualización** y si la caché esta llena debemos responder a las **políticas de sustitución**.

**¿Cómo calculamos la efectividad de la caché?** Calculando el tiempo medio de acceso a mejora.

T. acceso = (T. de acceso a caché \* Prob. de acierto) + (T. acceso a M.P. \* (1 – Prob. de Acierto))

Y luego, para ver el índice de memoria simplemente comparamos el tiempo sin caché con el tiempo con caché.

Índice de Mejora = T. sin caché / T. con caché

Si el tiempo con caché es mayor no hay índice de mejora, ya que el resultado sería peor y por lo tanto no sería efectivo.

**Tipos de Caché**

**Correspondencia directa**: cada bloque de M.P. solo puede acomodarse en una única línea de caché. No posee política de sustitución. El problema de la correspondencia directa, son las colisiones (a cada línea de caché le corresponden muchas de M.P.). Esto genera que 2 bloques se encuentren expulsándose continuamente.

La dirección de memoria principal esta representada de la siguiente manera:

|  |  |  |
| --- | --- | --- |
| ETIQUETA | LINEA | DESPLAZAMIENTO |

* Tamaño total de bits de la dirección de memoria: dado por el tamaño de M.P. (Ejemplo: 64k = 2\*…\*16 = 16 bits)
* Linea = log(2)C / C = numero de bloques de la caché (tamaño caché / tamaño bloque).
* Desplazamiento = log(2)K / K = tamaño de bloque (4bytes = 2\*…\*2 = 2).
* Etiqueta = Bits de palabra de M.P. – (Linea+Desplazamiento).

**Correspondencia asociativa**: cualquier bloque de memoria puede estar en cualquier línea de caché. Cuando se llene hay que utilizar políticas de sustitución.

La dirección de memoria principal está representada de la siguiente manera:

|  |  |
| --- | --- |
| NUMERO DE BLOQUE O ETIQUETA | DESPLAZAMIENTO |

* Tamaño total de bits de la dirección de memoria: dado por el tamaño de M.P. (Ejemplo: 64k = 2\*…\*16 = 16 bits)
* Desplazamiento = log(2)K / K = tamaño del bloque (4bytes = 2\*…\*2 = 2).
* Etiqueta = Bits de palabra de M.P. – Desplazamiento

**Correspondencia asociativa por conjuntos**: la caché se divide en X conjuntos de Y líneas cada uno. El bloque se ubica en cualquier línea del conjunto. A muchos bloques les corresponderá el mismo conjunto. Combina la economía de correspondencia directa con la tasa de aciertos de la asociativa.

La dirección de memoria principal está representada de la siguiente manera:

|  |  |  |
| --- | --- | --- |
| ETIQUETA | CONJUNTO | DESPLAZAMIENTO |

* Tamaño total de bits de la dirección de memoria: dado por el tamaño de M.P. (Ejemplo: 64k = 2\*…\*16 = 16 bits)
* Desplazamiento = log(2)K / K = tamaño del bloque (4bytes = 2\*…\*2 = 2).
* Conjunto = log(2)Q / Q = cantidad de conjuntos en los que se divide la caché (cant. Líneas / bloques x conjunto)
* Etiqueta = Bits de M.P. – (Desplazamiento+Conjunto)

**Políticas de Sustitución**

Se utilizan cuando se llena la memoria caché para designar a quien reemplazar por la nueva información.

Pueden ser basadas en estadística o no basadas en estadística.

* **Basadas en estadística** (generalmente buenas, fallan con algunas matrices)
  + **LRU (Last Recently Used)**: elimina al bloque utilizado con mayor antigüedad.
  + **LFU (Last Frequently Used)**: elimina al bloque menos utilizado frecuentemente.
* **No basadas en estadística** (buena tasa de aciertos, no falla con matrices, es económico)
  + **Random**: elimina un bloque al azar.
  + **FIFO** **(First In First Out)**: elimina el primer bloque ingresado de todos los que se encuentran.

**Políticas de Escritura/Actualización**

Se puede dar por 2 motivos, la falta de coherencia de la caché o por la falta de caché.

Si se produce por un tema de coherencia tenemos 2 tipos:

* **Escritura Inmediata (no hay incoherencia, pero hay mucho tráfico)**: realiza todas las escrituras tanto en caché como en M.P. logrando que esta última siempre esté actualizada.
* **Escritura Diferida (es rápida, pero posee incoherencias)**: escribe únicamente en caché y actualizará en memoria principal únicamente cuando el bloque sea reemplazado.

Si se produce por un tema de falta de caché tenemos otros 2 tipos:

* **Escritura sin asignación**: escribe directamente en memoria (se relaciona con escritura inmediata).
* **Escritura con asignación**: primero trae el dato a caché y escribe en esta memoria (se relaciona con escritura diferida).

**Tamaños de caché y bloques**

* **Caché**: cuanto mas grande es se vuelve mas lenta, mejora la tasa de aciertos hasta un punto determinado (más o menos 512kb).
* **Bloque**: cuanto mas grande es aumenta la tasa de aciertos hasta un punto determinado, tarda más en trasladarse a M.P., hay menos cantidad de bloques (implica muchas sustituciones) y las palabras están mas distantes (menos probabilidad de ser referenciadas).

**Modo Desprotegido o Real (procesadores 8086)**

Calculo de dirección física:

FFFF:AAAA [Dirección segmentada]

Segmento:Desplazamiento

Segmento x 10 (agrego un 0 al final) + desplazamiento = Dirección Física o Absoluta

**Memoria Segmentada Pura**

* Para acceder a la memoria física se puede utilizar una técnica denominada segmentación.
* Es de tamaño variable y el mismo esta administrado por el Sistema Operativo.
* Los segmentos contienen un solo tipo de objeto.
* Denominaremos dirección lógica a la entidad binaria que identifica una locación de memoria a nivel software.

**Memoria Virtual**

* Es una forma de administrar la memoria para asignar más de ella a cada tarea, utilizando almacenamiento en disco, que suele ser el dispositivo de acceso directo más rápido.
* Una dirección virtual debe traducirse a una dirección física; esto se intercala durante la ejecución del programa.
* Cuando no hay suficiente RAM, mueve los datos de esta a un espacio llamado **archivo de paginación**.