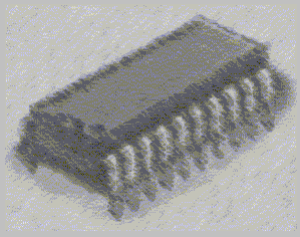
Tema 5. El Sistema de Memoria



Arquitectura de Computadores

I. T. Informática de Gestión

Curso 2009-2010

Índice

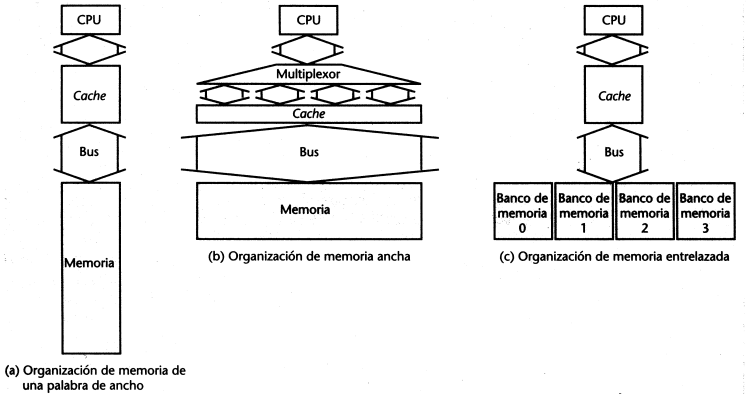
Características de la memoria

- El sistema de memoria es el lugar donde residen los programas y datos ya que según la arquitectura von Neumann un programa debe estar almacenado en memoria para poder ser ejecutado
- El sistema de memoria se organiza de manera jerárquica de forma que se tenga un sistema de alta capacidad, velocidad próxima a la de los dispositivos más rápidos y un coste cercano al de los dispositivos más lentos y baratos
- Factores a tener en cuenta en el diseño del sistema de memoria:
  - Capacidad de almacenamiento (tamaño máximo del sistema de memoria)
  - Velocidad (del conjunto)
  - Coste (del sistema de memoria)

Memoria principal (I)

- Para aumentar el rendimiento de la memoria principal se debe aumentar el ancho de banda (número de bits que se transmiten / segundo)
- Mecanismos para aumentar el ancho de banda de MP:
  - Reducción de su tiempo de acceso
  - Aumento del tamaño de palabra
  - Permitir el acceso concurrente a varios módulos de memoria, organización entrelazada

Memoria principal (II)



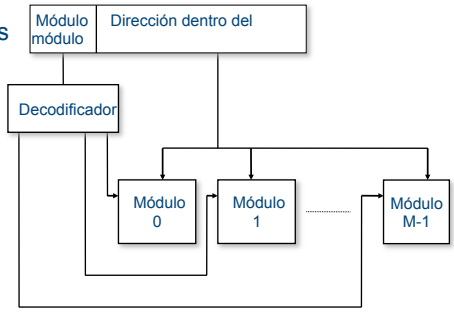
Memoria principal (III)
Entrelazado de memoria (I)

- Permitir el acceso concurrente a módulos de memoria de tecnología no demasiado rápida, y por tanto, de menor coste
- Permite multiplicar hasta por M el ancho de banda que se obtendría con un solo módulo, siendo M el número de módulos
- Conflictos de acceso la peticiones no permiten trabajar simultáneamente con todos los módulos
- Tipos de entrelazado:
  - Según la forma en que se distribuye el mapa de direcciones en memoria: de orden superior o de orden inferior
  - Según el modo en el que ser realiza el acceso a dichos módulos: entrelazado simple o entrelazado complejo

Memoria principal (IV)
Entrelazado de memoria (II)

Entrelazado de orden superior:

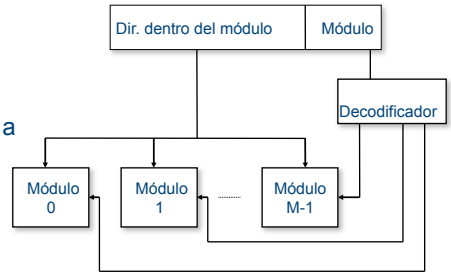
- Consiste en distribuir las direcciones de memoria entre los módulos de modo que cada uno contenga direcciones consecutivas
- **Ventajas:**
  - Facilita la expansión de la memoria
  - Fiabilidad ante el fallo de un módulo de memoria



Memoria principal (V)
Entrelazado de memoria (III)

Entrelazado de orden inferior:

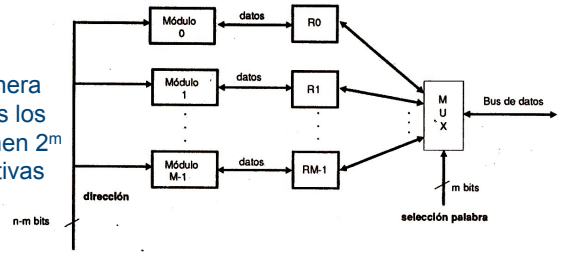
- Consiste en asignar a módulos consecutivos direcciones consecutivas del mapa de memoria
- **Ventaja**:
  - Si las referencias sucesivas a memoria son consecutivas, se reducen los conflictos de acceso



Memoria principal (VI)
Entrelazado de memoria (IV)

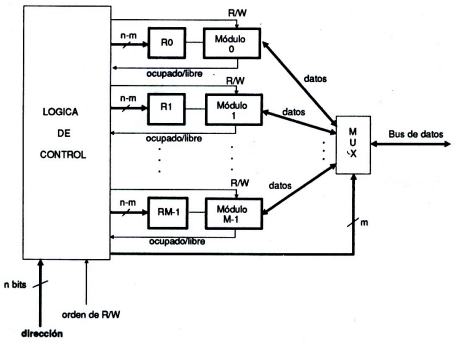
Entrelazado simple:

- Consiste en acceder a todos los módulos con la misma dirección
- Requiere entrelazado de orden inferior
- **Ventaja**:
  - Al acceder de manera simultánea a todos los módulos se obtienen 2^m palabras consecutivas

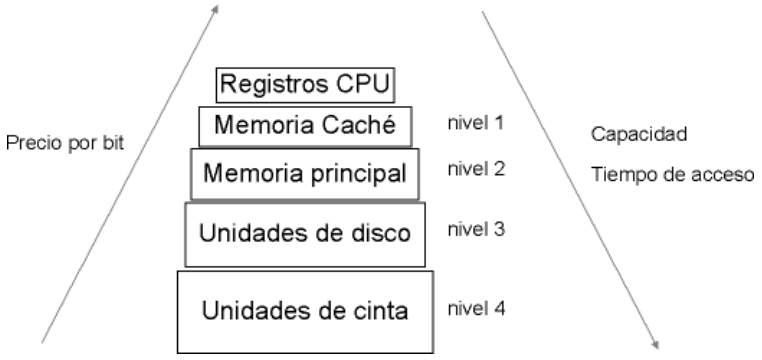


Memoria principal (y VII)
Entrelazado de memoria (y V)

- **Entrelazado complejo:**
- Consiste en acceder a todos los módulos con direcciones diferentes
- **Ventaja**:
  - Las direcciones a las que se accede no tienen que ser consecutivas



Jerarquía de memoria (I)



Jerarquía de memoria (II)

Modo de acceso a la información:

1º El procesador indica la dirección de la información en memoria principal

2º El acceso se intenta en el nivel más rápido (el de MCa)

- Si la información se encuentra, se accede (a la MCa)
- en caso contrario, se busca en el siguiente nivel (MP)
  - Si se encuentra (en MP), se transfiere al nivel anterior (MCa)
  - en caso contrario, se busca en el siguiente nivel,

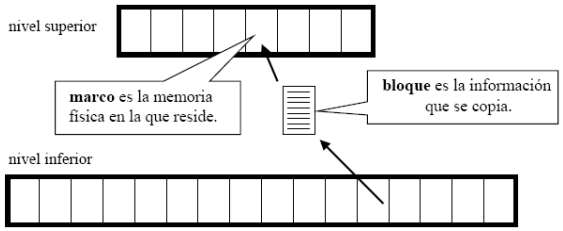
y así sucesivamente, ascendiendo la información hasta el primer nivel

Funcionamiento:

Se debe definir un mecanismo para realizar la traducción de direcciones ya que la memoria de un nivel contiene la información de la de nivel siguiente y no tiene por qué ocupar la misma dirección de memoria

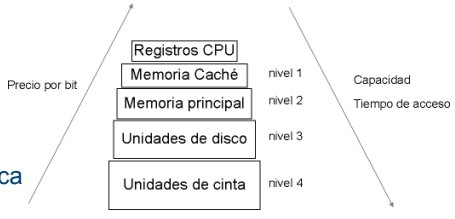
Jerarquía de memoria (III)

- **Bloque:** unidad lógica de intercambio de información entre niveles
  - En caché se suele llamar línea
  - En memoria virtual página o segmento
- **Marco:** espacio físico de memoria en el nivel en el que se almacena la información de un bloque



Jerarquía de memoria (IV)

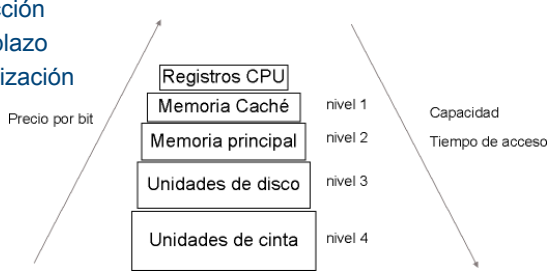
- **Acierto:** acceso en el que se encuentra el dato
- **Fallo:** acceso en el que no se encuentra el dato
- **Tasa de aciertos:** porcentaje de veces que se el dato que se busca
- **Tiempo de acierto:** tiempo para obtener el dato en un acierto
- **Penalización por fallo:** tiempo adicional que se tarda en obtener un dato cuando se produce un fallo
- El tamaño del bloque influye en la tasa de aciertos



$$T_m = t_{acc} + \nu_{fallos} * t_{pf}$$

Jerarquía de memoria (y V)

- **Se tiene que determinar en cada nivel:**
  - Política de ubicación
  - Política de extracción
  - Política de reemplazo
  - Política de actualización



Principios de localidad

- Las referencias a memoria por parte del procesador (instrucciones y datos), tienden a estar agrupadas. En periodos cortos de tiempo, el procesador trabaja con grupos fijos de referencias a memoria

Principio de	Comentarios
Localidad temporal	Si se referencia a un elemento, éste tendera a ser referenciado en un corto espacio de tiempo ▪ Los datos generados en operaciones siguientes ▪ Las mismas instrucciones dentro de un bucle
Localidad espacial	Si se referencia a un elemento, los elementos cercanos a él tenderán a ser referenciados también ▪ Los datos pertenecientes a un vector ▪ La estructura secuencial de un programa

Memoria cache (I)

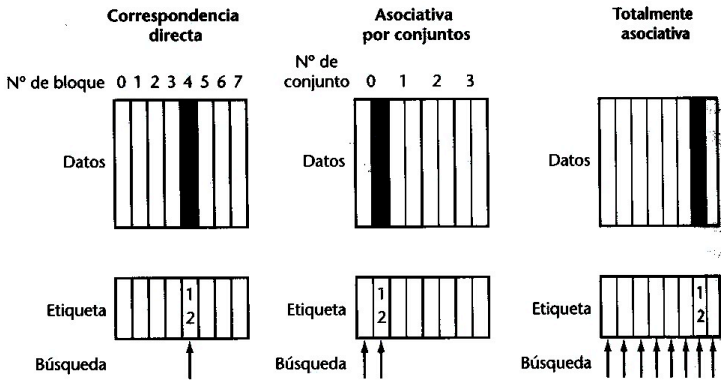
- **Objetivo:**
  - Dar la impresión de que las referencias memoria se sirven a una velocidad muy cercana a la del procesador
- **Diseño de la MCaché**: se debe tener en cuenta la optimización de los siguientes parámetros:
  - Probabilidad de acierto
  - Tiempo de acceso efectivo
  - Retardos debidos a fallos (actualizar de la MP en caso de escritura)
  - Establecer la correspondencia entre bloques de MP y MCa

Memoria cache (II)

- Parámetros de un sistema con memoria cache:
  - **Política de ubicación:** correspondencia entre bloques de MP y MCa
  - **Política de extracción:** qué y cuándo se envía información de MP a MCa
  - **Política de reemplazo:** qué bloque abandona MCa para dejar espacio si está llena
  - **Política de actualización:** cuándo se escribe en la MP
  - Tamaño más adecuado de la Mca y de sus bloques
  - **Unicidad y homogeneidad de la Mca:** una o varias MCa
  - Minimización del tiempo de espera en caso de fallo en MCa

Memoria cache (III)
Política de ubicación (I)

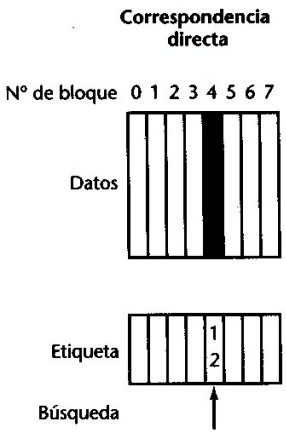
- Establece la correspondencia entre bloques de MP y MCa



Memoria cache (IV)
Política de ubicación (II)

Correspondencia directa

- Consiste en hacer corresponder a todo bloque **i** de MP el bloque (**i mod k**) de MCa, donde **k** es el número total de bloques de la MCa
- Una dirección en MCa consta de:
  - Etiqueta
  - Nº de bloque de MCa
  - Posición en el bloque (palabra)

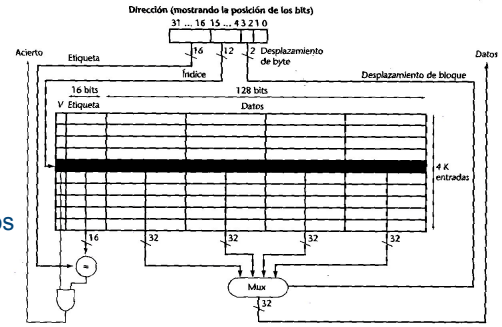


Memoria cache (V)
Política de ubicación (III)

Correspondencia directa

Ventaja:
La lectura permite el acceso simultáneo al directorio y a la palabra dentro del bloque de MCa

Inconveniente:
Incremento de la tasa de fallos de la MCa, si dos bloques de MP, que corresponden a un mismo bloque de MCa, se utilizan de forma alternativa



Memoria cache (VI)
Política de ubicación (IV)

Correspondencia totalmente asociativa

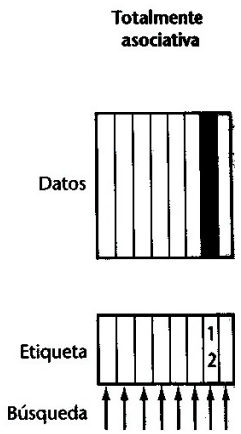
- Cualquier bloque de MP puede ubicarse en cualquiera de los bloques de la cache
- Una dirección en MCa consta de:
  - Etiqueta
  - Posición en el bloque (palabra)

Ventaja:

- Flexibilidad ya que permite la implantación de una gran variedad de algoritmos de reemplazo

Inconveniente:

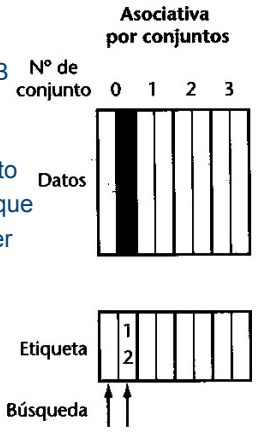
- Coste de las comparaciones



Memoria cache (VII)
Política de ubicación (V)

Correspondencia asociativa por conjuntos

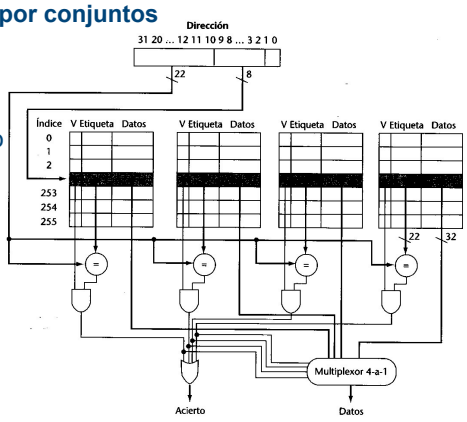
- Consiste en dividir la MCa en C conjuntos de B bloques cada uno
- Se aplica:
  - Correspondencia directa a nivel de conjunto
  - Correspondencia asociativa a nivel de bloque
- Un bloque i de MP puede ubicarse en cualquier bloque del conjunto (i mod C) de MCa
- Una dirección en MCa consta de:
  - Etiqueta
  - Conjunto
  - Posición en el bloque



Memoria cache (VIII)
Política de ubicación (VI)

Correspondencia asociativa por conjuntos

Ventaja:
Reduce el coste de la totalmente asociativa proporcionando un rendimiento cercano a esta última



Memoria cache (IX)
Política de ubicación (VII)

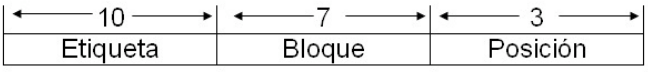
- **Ejemplo de correspondencia entre dirección de memoria principal y una memoria cache con correspondencia directa**

  Memoria principal de 1Mbyte y Memoria cache de 1Kbyte
  Tamaño del bloque de 8 bytes

$$\text{Número de bloques de memoria principal} = \frac{\text{Capacidad MP}}{\text{Tamaño bloque}} = \frac{2^{20}}{2^3} = 2^{17}\ \text{bloques}$$

$$\text{Número de bloques de memoria cache} = \frac{\text{Capacidad MCa}}{\text{Tamaño bloque}} = \frac{2^{10}}{2^3} = 2^7\ \text{bloques}$$

$$\frac{2^{17}}{2^7} = 2^{10}\ \text{bloques de MP por bloque de MCa}$$

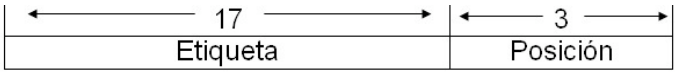


Memoria cache (X)
Política de ubicación (VIII)

- **Ejemplo de correspondencia entre dirección de memoria principal y una memoria cache totalmente asociativa**

  Memoria principal de 1Mbyte y Memoria cache de 1Kbyte
  Tamaño del bloque de 8 bytes

$$\text{Número de bloques de memoria principal} = \frac{\text{Capacidad MP}}{\text{Tamaño bloque}} = \frac{2^{20}}{2^3} = 2^{17}\ \text{bloques}$$



Memoria cache (XI)
Política de ubicación (y IX)

- **Ejemplo de correspondencia entre dirección de memoria principal y una memoria cache asociativa por conjuntos**

  Memoria principal de 1Mbyte y Memoria cache de 1Kbyte
  Tamaño del bloque de 8 bytes. Tamaño del conjunto 2 bloques

$$\text{Número de bloques de memoria principal} = \frac{\text{Capacidad MP}}{\text{Tamaño bloque}} = \frac{2^{20}}{2^3} = 2^{17} \text{ bloques}$$

$$\text{Número de bloques de memoria cache} = \frac{\text{Capacidad MCa}}{\text{Tamaño bloque}} = \frac{2^{10}}{2^3} = 2^7 \text{ bloques}$$

$$\text{Número de conjuntos de memoria cache} = \frac{\text{Bloques MCa}}{\text{Tamaño conjunto}} = \frac{2^7}{2^1} = 2^6 \text{ conjuntos}$$

$$\frac{2^{17}}{2^6} = 2^{11} \text{ bloques de MP por conjunto de MCa}$$

← 11 →	← 6 →	← 3 →
Etiqueta	Conjunto	Posición

Memoria cache (XII)
Política de extracción

- Indican cuándo y qué información se envía de MP a MCa
- Las más utilizadas son:
  - **Extracción por demanda:** traer a MCa el bloque en el momento en que se necesita, es decir, cuando se produce un fallo
  - **Extracción con anticipación:**
    - **Anticiparse siempre:** traer a MCa el bloque i+1 al referenciar el i
    - **Anticiparse si se produce fallo:** traer a MCa el bloque i+1 de MP solamente cuando se produce un fallo al refernciar el bloque i
    - **Anticipación marcada:** traer el bloque i+1 de MP al producirse el fallo al referenciar el i, marcarlo como traído y al referenciar el i+1 traer el i+2
    - **Extracción selectiva:** marca algún tipo de información para que nunca pueda ser enviada a MCa

Memoria cache (XIII)
Política de reemplazo

- Determina cuándo y qué bloque se sustituye en MCa porque al traer un bloque a cache cuando todos los marcos se encuentran ocupados (solamente se da en las caches asociativas y asociativas por conjuntos)
- Las más utilizadas son:
  - **Aleatoria:** consiste en elegir el bloque a reemplazar de forma aleatoria
  - **LRU Least-Recently Used:** consiste en reemplazar aquel bloque de MCa que no ha sido utilizado durante el mayor periodo de tiempo
  - **FIFO - First In First Out:** consiste en reemplazar el bloque que ha permanecido en MCa el mayor periodo de tiempo
- La política aleatoria es la más sencilla y menos costosa pero no aprovecha la localidad temporal
- En las políticas LRU y FIFO la complejidad y el coste aumentan a medida que aumenta el número de bloques entre los que se debe elegir

Memoria cache (XIV)
Política de actualización (I)

- Determina cuándo se actualiza la información en MP al haberse producido una escritura en MCa para evitar los problemas de falta de coherencia

Escritura inmediata: consiste en escribir en MCa y MP a la vez

Ventajas: la realización es muy sencilla y asegura la coherencia

Inconvenientes: produce mucho tráfico entre memoria y el procesador debe esperar a que se complete la escritura lo que lleva al empleo de buffer de escritura

Escritura aplazada: consiste en escribir en MCa y únicamente se escribe en MP si el bloque a reemplazar ha sido modificado
Ventajas: produce menos tráfico entre la memoria y el procesador y las escrituras se hacen a la velocidad de la cache
Inconvenientes: es diseño es más complejo ya que hace falta implementar el control del dirty bit

Memoria cache (XV) Política de actualización (y II)

- Las maneras más empleadas de actuar ante un fallo de escritura son:

- **Con ubicación:** consiste en llevar el bloque que produce el fallo de MP a MCa y a continuación realizar la escritura en MCa
- Suele ir unida a la escritura aplazada

- **Sin ubicación:** consiste en escribir únicamente en MP.
- Suele ir asociada a la escritura inmediata

Memoria cache (XVI) Minimización del tiempo de espera ante un fallo

- Las maneras más empleadas de actuar ante un **fallo de lectura** son:

- **Early start:** consiste en enviar al procesador la palabra que produjo el fallo tan pronto como llegue a cache y sin esperar a que se complete la transferencia del bloque
- **Out of order fetch:** consiste en enviar la palabra que produjo el fallo a MP y a MCa y luego terminar de transferir el bloque

- La manera más empleada de actuar ante un **fallo de escritura** es el empleo de un buffer de escritura, de tal manera que en el caso de **escritura sin ubicación** se escribe en el buffer y éste luego escribirá la información en MP

Memoria cache (XVII)
Otras características a tener en cuenta

Tamaño de la memoria cache y sus bloques

- Aumentar el tamaño del bloque hace que aumenta la tasa de aciertos, por la localidad espacial y que disminuya el número de bloques en MCa por la localidad temporal
- Disminuir el tamaño del bloque hace que disminuya la cantidad de tráfico de información entre MCa y MP

Unicidad y homogeneidad de la memoria cache

- Las instrucciones presentan localidad temporal y los datos espacial
- Acceso simultaneo a instrucciones y datos (aumenta la velocidad)

Cache de uno y dos niveles

- Mejora de las prestaciones con MCa de 2 niveles
- **MCa integrada:** elimina el acceso al bus y aumentan los tiempos de ejecución

Memoria cache (XVIII)
Cache en multiprocesadores (I)

- **Latencia de memoria:** tiempo transcurrido desde que un procesador realiza la petición de acceso a memoria hasta que se completa

Depende de:

- Los conflictos de acceso producidos al tratar de acceder varios procesadores simultáneamente al mismo módulo de memoria principal
- Los retardos y conflictos introducidos por la red de interconexión de los procesadores
- El exceso de tráfico entre la red de interconexión y la memoria

Caché en multiprocesadores:

- **Cache compartida:** una única cache compartida por todos los procesadores con lo que se evita el problema de la coherencia de la información (**desventaja**: puede convertirse en el cuello de botella)
- **Cache privada**

Memoria cache (XIX)
Cache en multiprocesadores (II)

- **Cache privada**: memoria cache asociada a cada procesador
- **Ventaja**: disminuye el problema de la latencia y el tráfico de información entre la memoria principal y los procesadores
- **Inconveniente**: problemas de falta de coherencia de la información entre las diferentes caches

Tipos de cache privadas:

- **Caches locales:** cada cache contiene únicamente información de sólo lectura y datos de escritura locales al procesador asociado. Las referencias a datos compartidos directamente sobre MP
- **Directorio compartido:** se emplea una zona de memoria compartida para almacenar de forma centralizada el estado de cada uno de los bloques de información presentes en las caches
- **Protocolo de escucha:** dotar a las caches de la capacidad para permanecer a la escucha de las peticiones de acceso a memoria que aparecen en el bus

Memoria cache (y XX)
Cache en multiprocesadores (y III)

Política de actualización en caches privadas:

- **Inmediata:** al escribir en una posición de memoria cache se escribe también en MP. Las demás caches “escuchan” esta escritura e invalidan su copia. Si alguna de ellas requiere acceder a la copia invalidada se produce un fallo y traerá de MP esa información

- **Aplazada (escribir una única vez):** la primera vez que un procesador escribe en una posición de su cache. Se actualiza también esa información en la MP. De esta forma el resto de caches invalidan el bloque. A partir de entonces, dicho bloque “pertenecerá” al procesador que realizó la escritura con lo que puede continuar escribiendo en él sin ningún problema. La cache que contiene la copia válida debe “escuchar” si alguna otra cache pide el bloque a MP. En ese caso, actualizará el valor en MP antes de que se envíe ese bloque a la otra cache que lo ha solicitado

Sistemas de memoria
Pentium Pro y PowerPC 604

Característica	Pentium Pro	PowerPC 604
Direccion virtual	32 bits	52 bits
Dirección física	32 bits	32 bits
Tamaño de página	4KB, 4MB	4KB, 256MB
Organización TLB	TLB para datos TLB para instrucciones Asociativas de 4 vías Reemplazo pseudo LRU TLB-i 32 entradas TLB-d 64 entradas Fallos TLB tratados por hardware	TLB para datos TLB para instrucciones Asociativas de 2 vías Reemplazo LRU TLB-i 128 entradas TLB-d 128 entradas Fallos TLB tratados por hardware
Organización cache	Caches de datos e instrucciones separadas	Caches de datos e instrucciones separadas
Tamaño cache	8KB cada una	16 KB cada una
Asociatividad	Asociativas de 4 vías	Asociativas de 4 vías
Reemplazo	LRU aproximado	LRU
Tamaño bloque	32 byte	32 byte
Actualización	Aplazada	Aplazada o inmediata

Bibliografía

- Estructura y diseño de computadores
  David A. Patterson y John L. Hennessy. Reverté, 2000
  Capítulo 7
- Arquitectura de computadores. Un enfoque cuantitativo
  John L. Hennessy y David A. Patterson. Mc Graw Hill, 3ª ed, 2002
  Capítulo 8
- Arquitectura de computadores
  José A. de Frutos y Rafael Rico. Servicio de Publicaciones de la Universidad de Alcalá, 1995
  Capítulo 5
- Fundamentos de computadores
  Pedro de Miguel Anasagasti. Paraninfo, 1999
  Capítulo 11