Universidad de Jaén

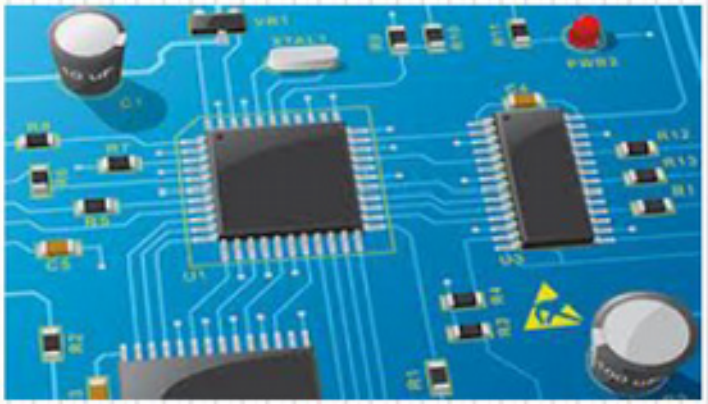
Compromiso con la sociedad

25 AÑOS DE UNIVERSALIDAD 1993-2018

T3. Sistema de Memoria:

T3.4 Aspectos avanzados Memoria Caché

FUNDAMENTOS DE ARQUITECTURA DE COMPUTADORES



Departamento de Informática

Contenido del capítulo

- Diseño de la memoria caché:
  - Política de ubicación.
  - Caché asociativa y asociativa por conjuntos
  - Política de actualización, extracción y reemplazo
- Rendimiento de la caché
- Avanzado: Niveles de memoria caché y Cachés separadas
- Coherencia de caché
- Bibliografía
- Actividades

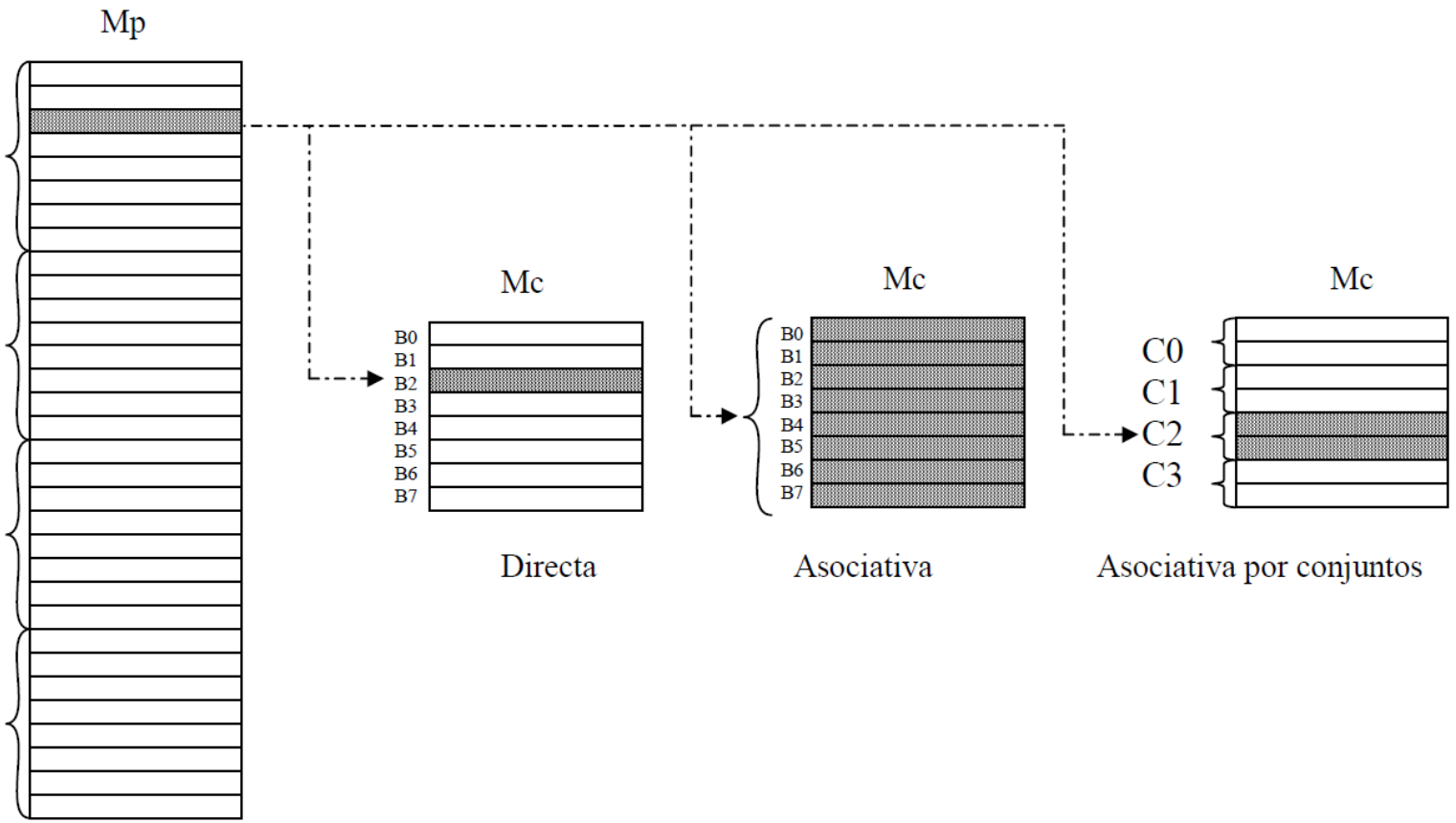
Contenido del capítulo

- Diseño de la memoria caché:
  - Política de ubicación
  - Caché asociativa y asociativa por conjuntos
  - Política de actualización, extracción y reemplazo
- Rendimiento de la caché
- Avanzado: Niveles de memoria caché y Cachés separadas
- Coherencia de caché
- Bibliografía
- Actividades

Política de Ubicación

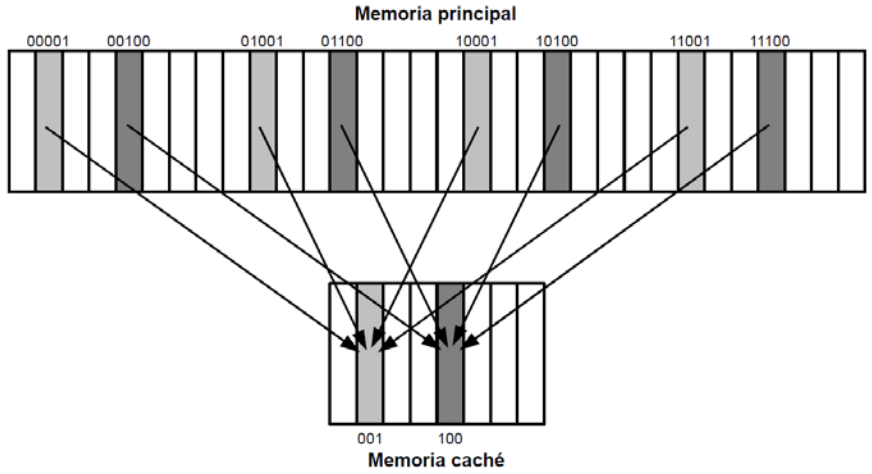
- Existen tres funciones de correspondencia para definir la posible ubicación de un bloque de MP en la memoria caché:
- **Directa:** un bloque de MP sólo puede ubicarse en una línea de la caché.
- **Asociativa:** un bloque puede ubicarse en cualquier línea
- **Asociativa por conjuntos:** es un compromiso entre las dos anteriores

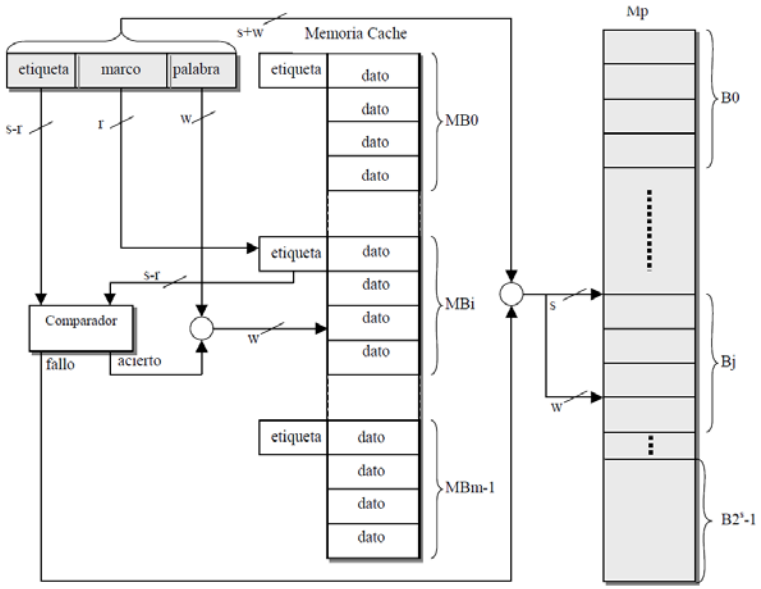
Política de Ubicación (2)

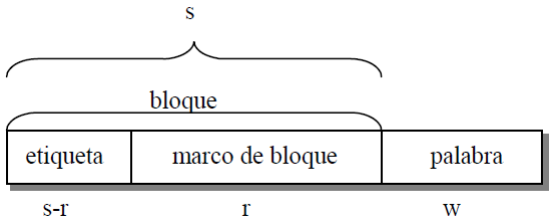


Ubicación: Correspondencia directa

- Bloque de MP SOLAMENTE puede estar en una línea de caché:
  - **Y = X módulo N**,
- El bloque guardado en la caché está determinado por los bits de "etiqueta".
- 2^w palabras/bloque
- 2^s bloques de MP
- 2^r líneas en MC ($2^r = N$)
- 2^{s-1}





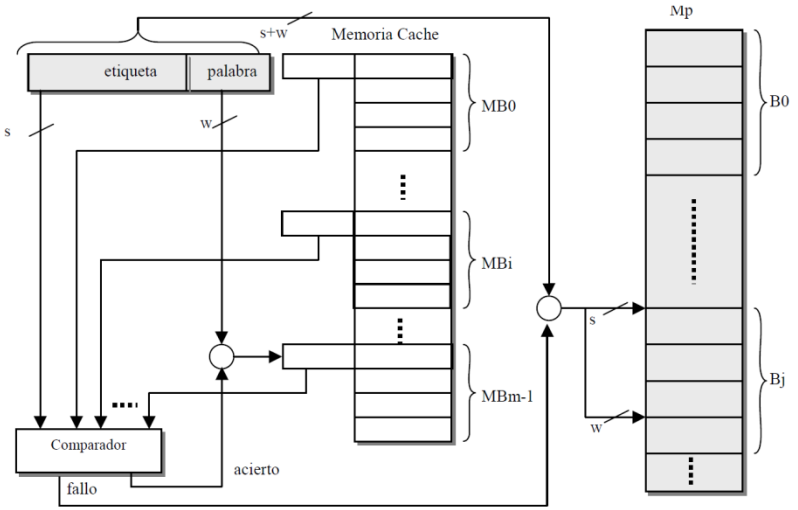


Contenido del capítulo

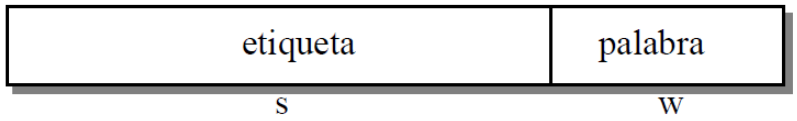
- Diseño de la memoria caché:
  - Política de ubicación.
  - Caché asociativa y asociativa por conjuntos
  - Política de actualización, extracción y reemplazo
- Rendimiento de la caché
- Avanzado: Niveles de memoria caché y Cachés separadas
- Coherencia de caché
- Bibliografía
- Actividades

Ubicación: Correspondencia Asociativa

- Un bloque de MP puede ir a **cualquier** línea de MC.
- Para saber que bloque de MP está guardado en MC se identifica con los bits de etiqueta.



- 2^s bloques de MP
- 2^r líneas (marcos de bloque) de MC



Ej. de Direccionamiento: Asociativo

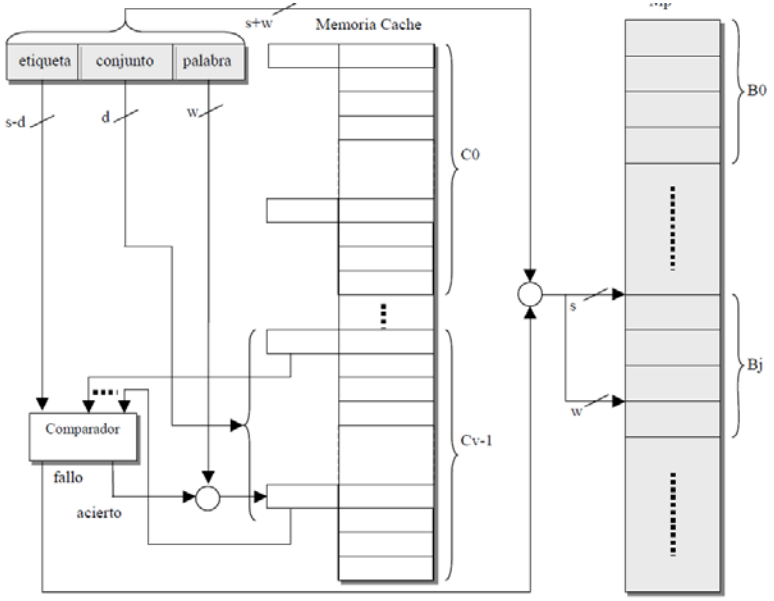
- Dirección de petición de datos →

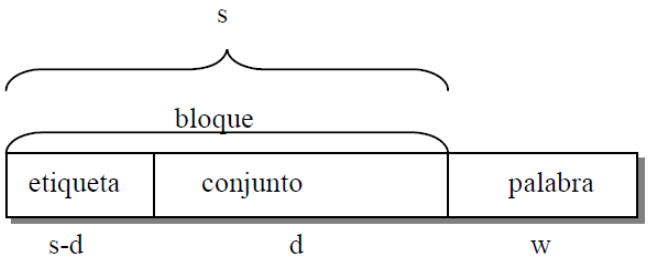
Marca	Palabra
11010001011111	1010

- Cualquier bloque de MP puede ir a cualquier línea de caché:
  - 2^{14} bloques MP → 14 bits para la etiqueta
- Todas las etiquetas se guardan en una memoria asociativa para comparación simultánea con la etiqueta de la dirección física:
  - Para cada una de las líneas se verifica el bit de validez.
  - Aquellas líneas que sean "válidas", se chequea el campo de etiqueta.
    - Si alguna etiqueta coincide, se extrae de ahí el dato.
    - En caso contrario, se debe traer el dato de MP a caché.
- En el ejemplo, debemos ir a la línea de caché con una etiqueta = 0x345F (si existe!) y buscar la posición 0x0A

Ubicación: Asociativa por Conjuntos

- Líneas de caché se agrupan en $v = 2^d$ conjuntos.
- Cada conjunto tendrá un número k de vías (líneas).
- Número total líneas $l = v \cdot k$
- Un bloque X de MP se asignará dentro del conjunto Y de MC:
  - $Y = X$ *módulo* v.
  - Dentro del conjunto Y-ésimo, podrá ubicarse en cualquier línea [0,…, k-1]
- Diseño de las cachés actuales.





Ej. Direccion.: Asociativo x conjuntos

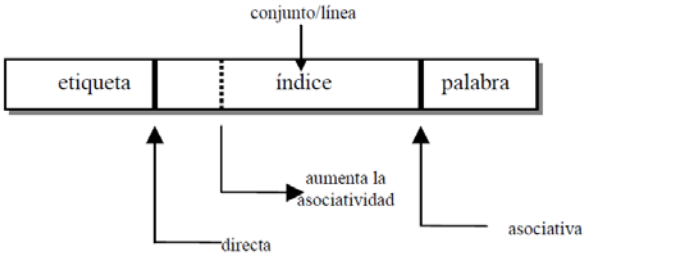
- Dirección de petición de datos →

Marca	Conjunto	Palabra
11010001	011111	1010

- Memoria caché asociativa de dos vías.
- Dado que hay 2^7 líneas, tendremos $2^7 / 2 = 2^6$ conjuntos
- Asignar una dirección a un conjunto → direccionamiento directo.
- Dentro del conjunto se utiliza una asignación de tipo asociativo.
- Cuando se recibe una dirección, se realizan los siguientes pasos:
  - (1) Buscar conjunto de caché correspondiente, en nuestro caso 0x1F
  - (2) Se verifica la etiqueta para las dos líneas que contiene: 0xD1
  - (3) Se chequean bits de validez y suciedad y se accede a palabra 0x0A

Direccionamiento: Resumen

- Las tres funciones de correspondencia se pueden ver en realidad como una sola: la asociativa por conjunto,



- Dimensión real de la caché:
  - #Total_bits = #posiciones x (#bits_ctrl + etiqueta + bloque)
    - Bloque = (#palabras_bloque) x (#bits_palabra)
    - Etiqueta = (dirección) – (#bits_índice) – (offset)
    - #bits_índice = log2 (#posiciones)
    - offset = log2 (#palabras_bloque) ó log2 (#bytes_palabra)

Contenido del capítulo

- Diseño de la memoria caché:
  - Política de ubicación
  - Caché asociativa y asociativa por conjuntos
  - Política de actualización, extracción y reemplazo
- Rendimiento de la caché
- Avanzado: Niveles de memoria caché y Cachés separadas
- Coherencia de caché
- Bibliografía
- Actividades

Política de Actualización de MP

- **Escritura directa** (Write-Through): Si se modifica un dato en caché se actualiza también en MP:
  - *Escritura directa con asignación en escritura*: Se cargan bloques en memoria caché tanto si hay fallos para lectura como para escritura.
  - *Escritura directa sin asignación en escritura*: Se cargan bloques en caché si hay fallos (faltas) por lectura pero no por faltas por escritura.
- **Postescritura** (Write-Back): Se actualiza la MP cuando se reemplace el bloque de memoria caché:
  - *Postescritura siempre*: Los bloques que abandonan la caché se reescriben en memoria principal → excesivo trasiego de información.
  - *Postescritura marcada*: Solo reescribir bloques modificados (dirty bit).

Política de búsqueda o extracción

- Por demanda
  - Se lleva un bloque a Mc cuando se referencia desde la CPU alguna palabra del bloque y éste no se encuentra en Mc
- Anticipativa (prebúsqueda) → Localidad espacial.
  - Prebúsqueda siempre: la primera vez que se referencia el bloque Bi se busca también Bi+1
  - Prebúsqueda por fallo: cuando se produce un fallo al acceder al bloque Bi se buscan los bloques Bi y Bi+1
  - Datos "inútiles" en caché…

Diseño: Políticas de reemplazo

- **Regla FIFO**: Se destruye la fracción que más tiempo lleva en la caché. Se implementa mediante una cola.
- **Regla LRU**: Porción con más tiempo sin haber sido usada o actualizada. Asociar contadores a las líneas de caché.
- **Regla LFU**: Porción que se ha pedido menos veces desde que se inició el proceso. Asociando un contador a cada marco de bloque.
- **Regla RAND**: Se elige un bloque al azar.

Contenido del capítulo

- Diseño de la memoria caché:
  - Política de ubicación
  - Caché asociativa y asociativa por conjuntos
  - Política de actualización, extracción y reemplazo
- Rendimiento de la caché
- Avanzado: Niveles de memoria caché y Cachés separadas
- Coherencia de caché
- Bibliografía
- Actividades

Rendimiento de la caché

- Si frente a un fallo, el bloque de MP se lleva a MC al tiempo que la palabra referenciada del bloque se lleva (en paralelo) a la CPU, el tiempo de acceso a memoria será:
  - Tacceso = Na * Tc + Nf * Tp
    - Na es el número de referencias con acierto
    - Nf es el número de referencias con fallo
    - Tc es el tiempo de acceso a una palabra de Mc
    - Tp es el tiempo de acceso a un bloque de Mp
- Tp es la componente principal del tiempo total de penalización por fallo.
- El tiempo de acceso medio durante la ejecución del programa valdrá:
  - Tacceso_medio = Tacceso/ Nr = Ta*Tc + Tf*Tp donde:
    - Ta = Na/Nr es la tasa de aciertos
    - Tf = Nf/Nr es la tasa de fallos; Tf = 1 - Ta
    - Nr es el número total de referencias a memoria; Nr = Na + Nf

Rendimiento de la caché (2)

- Para mejorar el rendimiento de una caché, podemos actuar sobre tres términos, dando lugar a tres tipos de optimizaciones:

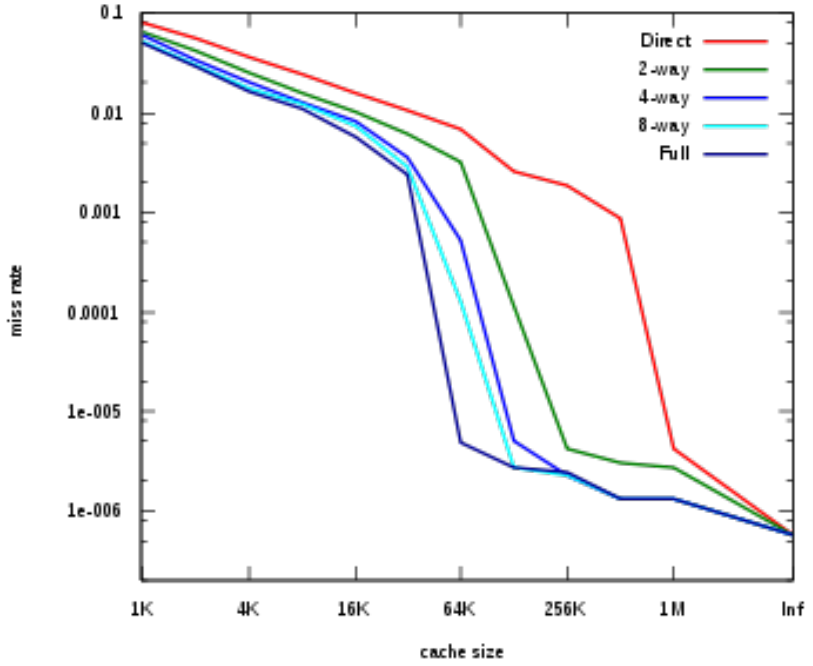
1. Reducción de la tasa de fallos, Tf
   - Tamaño de bloque
   - Grado de asociatividad
   - Niveles de caché
   - Optimizaciones del compilador
2. Reducción de la penalización de los fallos, Tp
   - Prioridad de fallos de lectura ante escritura
   - Uso de sub-bloques dentro de un bloque
   - Niveles de caché
3. Reducción del tiempo de acierto, Tacierto
   - Cachés pequeñas y simples

Contenido del capítulo

- Diseño de la memoria caché:
  - Política de ubicación
  - Caché asociativa y asociativa por conjuntos
  - Política de actualización, extracción y reemplazo
- Rendimiento de la caché
- Avanzado: Niveles de memoria caché y Cachés separadas
- Coherencia de caché
- Bibliografía
- Actividades

Niveles de memoria caché

- A mayor tamaño de caché, mayor probabilidad de encontrar un dato
- Existe un límite a partir del cual no interesa aumentar su tamaño.
- A mayor tamaño, mayor número de comparaciones respecto a la etiqueta.
- Caché Víctima (caché L2):
  - "Repositorio" de L1
  - Almacena bloques recién extraídos
  - Mantener la localidad temporal.



Niveles de memoria caché (2)

- Una solución alternativa al complicado diseño de una caché de gran tamaño: **cachés multinivel.**
- Una jerarquía multinivel consiste en "h" niveles de cachés, $c_h \ldots c_1$ que forman una estructura en árbol.
- La utilización de una caché de segundo (o tercer) nivel presenta las siguientes ventajas:
  1. Disminución del **tiempo de latencia**: Fallos en primer nivel se recuperan en segundo nivel: más rápido que la MP.
  2. Disminución de la **utilización del bus**: Disminuye el número de accesos a través del bus de MP.

Caché separada ~ datos/instrucciones

- Inicialmente los equipos constaban de una sola caché por simplicidad y aprovechamiento de ésta
- Particionar la memoria caché L1 permite:
  - Emisión de direcciones de instrucción y dato a la vez:
    - Doblar el ancho de banda entre la caché y la CPU,
    - Muy efectivo si se utilizan técnicas pipelines y superescalares.
  - Optimizar cada caché separadamente:
    - Diferentes capacidades, tamaños de bloque y asociatividades
    - De este modo se logra un menor tiempo de acceso.

Caché separada ~ datos/instrucciones

- Características de la caché de datos:
  - Es menos eficiente debido a la localidad de los mismos.
  - Es más compleja por la posibilidad de modificación de los datos
- Características de la caché de instrucciones:
  - Lazos de programas y rutinas en caché: Se evita el acceso a M.P. para código corto y frecuente: **inline**
  - Caché de "sólo lectura".
  - Caché de traza (Pentium IV)

Contenido del capítulo

- Diseño de la memoria caché:
  - Política de ubicación
  - Caché asociativa y asociativa por conjuntos
  - Política de actualización, extracción y reemplazo
- Rendimiento de la caché
- Avanzado: Niveles de memoria caché y Cachés separadas
- Coherencia de caché
- Bibliografía
- Actividades

Coherencia de caché

- Datos replicados a lo largo de jerarquía de memoria deben contener la misma información para la misma dirección física
- Replicar la modificación del nivel más alto a los inferiores.
- Esto puede cobrar más relevancia en el caso de los procesadores **multinúcleos**:
  - En general, se comparte un nivel de cache (por ej. L2), pero cada núcleo dispone de su propia L1 donde modifica los datos.
  - Incoherencia de datos que lee un núcleo y han sido modificados recientemente por otro: sin modificar en la cache compartida.

Coherencia de caché (2)

- Protocolo usado para establecer la coherencia de caché: **MESI**.
- Dos bits de estado por cada etiqueta, de manera que cada línea puede estar en uno de estos cuatro estados:
  - **M**odificada: La línea de caché ha sido modificada (diferente de memoria principal) y está disponible sólo en caché.
  - **E**xclusiva: El contenido de la línea de caché coincide con lo que se tiene en memoria principal y no ésta presente en ninguna otra caché.
  - Compartida (**S**hared): La línea en la caché coincide con memoria principal y puede estar presente en otra caché.
  - No válida (**I**nvalid): La línea de la caché no contiene datos válidos.
- En función de estos 4 estados, cuando se accede a memoria se comprueba su valor y se actúa en consecuencia.

Contenido del capítulo

- Diseño de la memoria caché:
  - Política de ubicación
  - Caché asociativa y asociativa por conjuntos
  - Política de actualización, extracción y reemplazo
- Rendimiento de la caché
- Avanzado: Niveles de memoria caché y Caché separadas
- Coherencia de caché
- Bibliografía
- Actividades

Bibliografía

- Patterson y Hennessy: Estructura y Diseño de Computadores. Capítulo 5.

- Murdocca y Heuring: Principios de Arquitectura de Computadoras: Capítulo 7.

- Memoria del computador. Disponible en http://www.slideshare.net/Sofylutqm/memoria-del-computador

Contenido del capítulo

- Diseño de la memoria caché:
  - Política de ubicación
  - Caché asociativa y asociativa por conjuntos
  - Política de actualización, extracción y reemplazo
- Rendimiento de la caché
- Avanzado: Niveles de memoria caché y Cachés separadas
- Coherencia de caché
- Bibliografía
- Actividades

Actividades

- Realiza uno de los siguientes dos ejercicios:
  - (1) Un computador tiene una memoria principal de 1MB y una memoria caché con 16 marcos de 64 bytes cada uno, siendo la palabra de 8 bits.
    - Determinar el formato de la dirección física, (número de bits que ocupa cada uno de los campos) para caché directa, asociativa y asociativa por conjuntos de 4 vías.
    - Calcular el número de bits necesarios para implementar la caché en cada uno de estos casos incluyendo datos y etiquetas. (obviar los bits de validez, etc.).
    - Tiempo medio de acceso a memoria dados los siguientes datos de benchmarking:
      - Un programa realiza 275 acceso a memoria, donde un total de 264 son accesos directamente hacia/desde la caché.
      - 5 ciclos para acceder a la caché, 4 para búsqueda y 1 más para trasferencia.
      - En caso de fallo, el núm. de ciclos para la trasferencia desde MP será 25 ciclos.
      - La frecuencia de reloj del sistema son 800MHz.
    - ¿Qué cambios habría en función del tamaño de la caché, tasa de aciertos y tiempo de búsqueda al aumentar la asociatividad?¿Y si duplicamos el número de palabras por bloque? Razone su respuesta; no es necesario el uso de datos numéricos exactos.

Actividades (2)

- (2) Un computador de 16 bits tiene una memoria máxima de 8 KiB, junto con una memoria caché de 4 Kib. La memoria caché es asociativa por conjuntos de 2 vías, con un total **16 conjuntos**. La política de reemplazo utilizada es FIFO.
  - ¿Cuál es el tamaño del bloque o línea?¿Cuál es el tamaño y formato de las direcciones de memoria?
  - Suponiendo que la caché está inicialmente vacía ¿cuántos fallos se producirían en la caché si se leyeran consecutivamente las siguientes direcciones de la memoria principal? 0x00F, 0x04F, 0x10F, 0x10C, 0x28F, 0x00D.
  - El tiempo medio de acceso en este sistema de memoria suponiendo que una estadística realizada muestra que, de cada 178 accesos a memoria, 160 se realizan desde la caché; y que asimismo el tiempo de comprobación de caché es de 2ns., el tiempo de acceso a caché otros 2ns. y el tiempo en traer el dato directamente desde la memoria 20ns.