S9-Memoria-del-Sistema

Martín Cusme, Celeste Gallardo, Josune Singaña, Richard Tipantiza

Algoritmo de Sustitución (E2,7,148)

Política de escritura

Política de escritura

Tamaño de Linea

Tamaño de Línea

Número de Cachés (E2, 7, 150)

Referencias

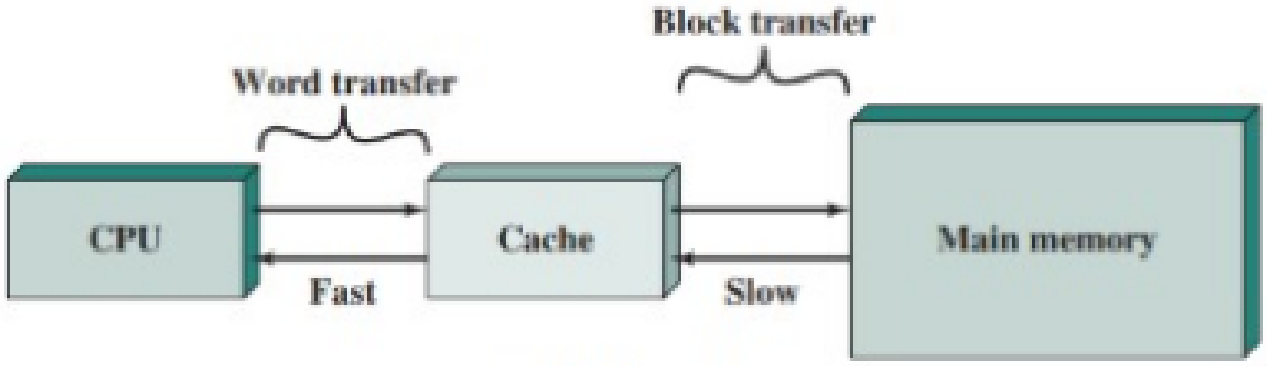
Bibliografía

- Este documento tiene la propuesta de temas a tratar y desarrollar por los estudiantes.
- Se ha de utilizar como base la bibliografía recomendada, pero puede consultar bibliografía adicional.

1. ¿Para que sirve?
   - El objetivo principal de la memoria caché es mejorar la velocidad de acceso a los datos almacenados, combinando el acceso rápido a datos de una memoria más cara y de alta velocidad (memoria caché) con el almacenamiento más lento pero de mayor capacidad de la memoria principal.
2. Funcionamiento

- La CPU transfiere palabras o bloques entre la caché y la memoria principal. La caché actúa como intermediaria rápida entre la CPU y la memoria principal, almacenando temporalmente datos que la CPU necesita frecuentemente.

- En el modelo simple de caché (como muestra la Figura 5.1a), la CPU realiza transferencias rápidas a la caché y transferencias más lentas a la memoria principal.



(a) Single cache

- Se organizan en varios niveles (L1, L2, L3). A medida que se avanza en los niveles, la velocidad disminuye, pero la capacidad aumenta.
  - Caché de Nivel 1 (L1): La más rápida y de menor capacidad.
  - Caché de Nivel 2 (L2): Un poco más lenta, pero con mayor capacidad.

- Caché de Nivel 3 (L3): Menos rápida que L1 y L2, pero aún más rápida que la memoria principal.

- "La memoria caché mejora la velocidad de acceso al reducir la distancia entre el procesador y la memoria principal."

- "Los fallos de caché generan tráfico en el bus del sistema."

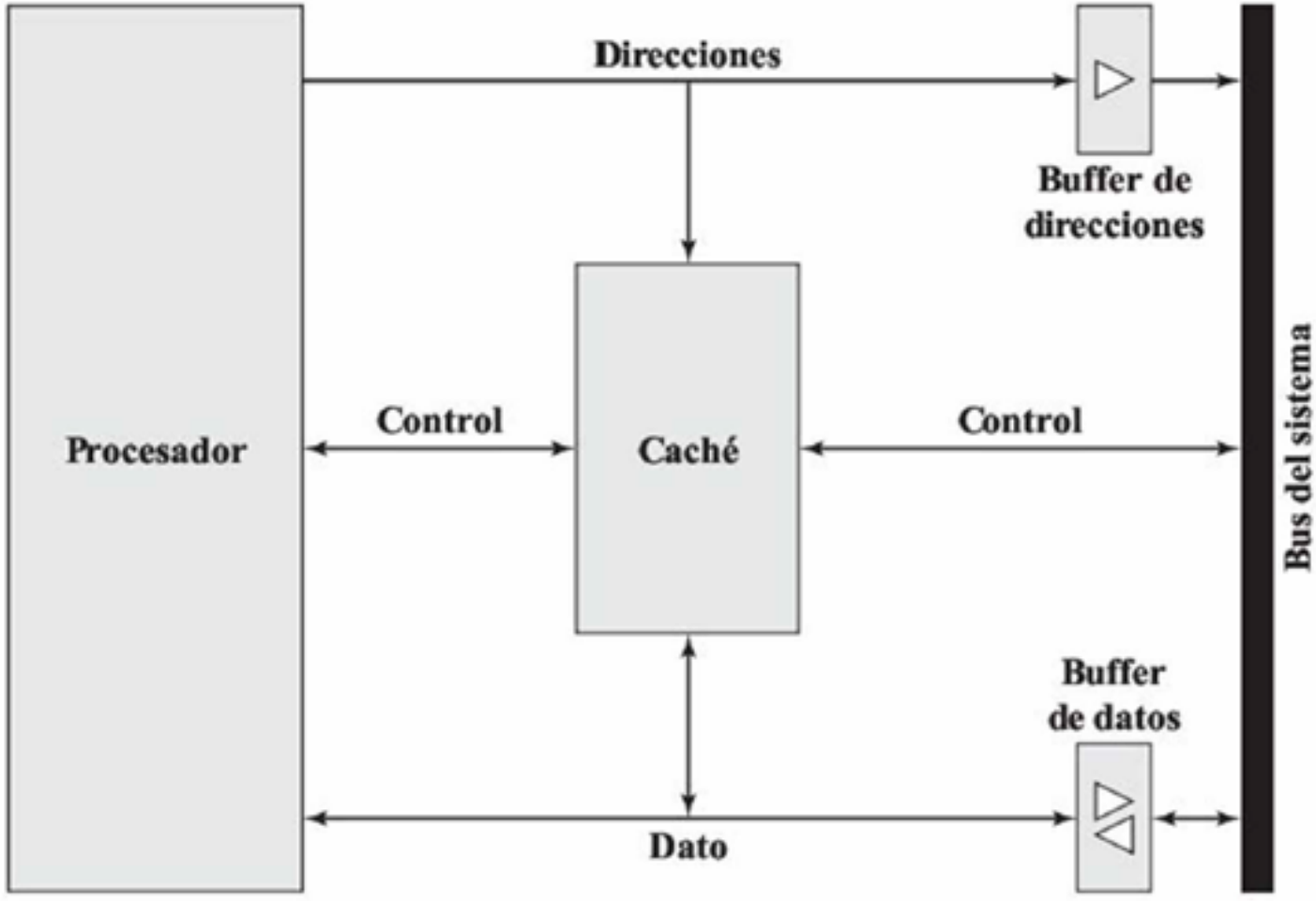


Figura 4.6. Organización típica de caché.

1. Parámetros Principales
   - "La función de correspondencia, el tamaño de línea y el algoritmo de sustitución son clave para el diseño de una caché eficiente."

- "La jerarquía de cachés puede mejorar el rendimiento en aplicaciones bien optimizadas."

Table 5.1 Elements of Cache Design

Cache Addresses	**Write Policy**
Logical	Write through
Physical	Write back
Cache Size	**Line Size**
Mapping Function	**Number of Caches**
Direct	Single or two level
Associative	Unified or split
Set associative	
Replacement Algorithm	
Least recently used (LRU)	
First in first out (FIFO)	
Least frequently used (LFU)	
Random	

1. Consideraciones de Tamaño
   - ."El tamaño de la caché impacta directamente en su velocidad y costo."
   - "No existe un tamaño 'óptimo' único, ya que depende de la naturaleza de las tareas."

Table 5.2 Cache Sizes of Some Processors

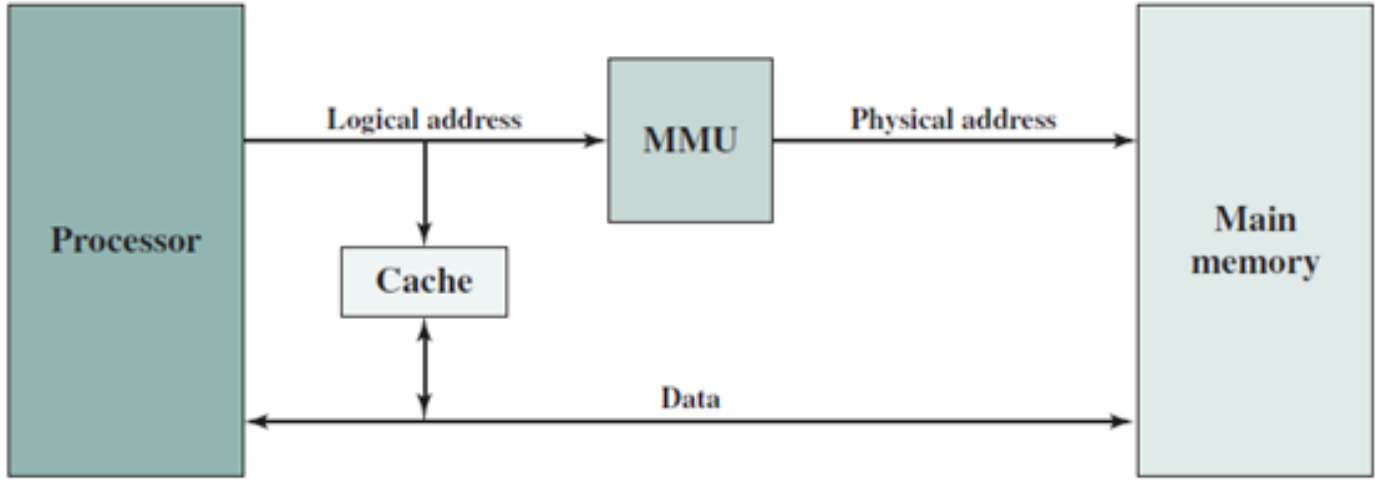
Processor	Type	Year of Introduction	L1 Cache[a]	L2 cache	L3 Cache
IBM 360/85	Mainframe	1968	16 to 32 kB	—	—
PDP-11/70	Minicomputer	1975	1 kB	—	—
IBM 3033	Mainframe	1978	64 kB	—	—
IBM 3090	Mainframe	1985	128 to 256 kB	—	—
Intel 80486	PC	1989	8 kB	—	—
Pentium	PC	1993	8 kB/8 kB	256 to 512 kB	—
PowerPC 620	PC	1996	32 kB/32 kB	—	—
IBM S/390 G6	Mainframe	1999	256 kB	8 MB	—
Pentium 4	PC/server	2000	8 kB/8 kB	256 kB	—
Itanium	PC/server	2001	16 kB/16 kB	96 kB	4 MB
Itanium 2	PC/server	2002	32 kB	256 kB	6 MB
IBM POWER5	High-end server	2003	64 kB	1.9 MB	36 MB
CRAY XD-1	Supercomputer	2004	64 kB/64 kB	1MB	—
IBM POWER6	PC/server	2007	64 kB/64 kB	4 MB	32 MB
IBM z10	Mainframe	2008	64 kB/128 kB	3 MB	24-48 MB
Intel Core i7 EE 990	Workstaton/ Server	2011	6 × 32 kB/32 kB	6 × 1.5 MB	12 MB
IBM zEnterprise 196	Mainframe/ Server	2011	24 × 64 kB/128 kB	24 × 1.5 MB	24 MB L3 192 MB L4
IBM z13	Mainframe/ server	2015	24 × 96 kB/128 kB	24 × 2 MB/2 MB	64 MB L3 480 MB L4
Intel Core i0-7900X	Workstation/ server	2017	8 × 32 kB/32 kB	8 × 1 MB	14 MB

[a]Two values separated by a slash refer to instruction and data caches.

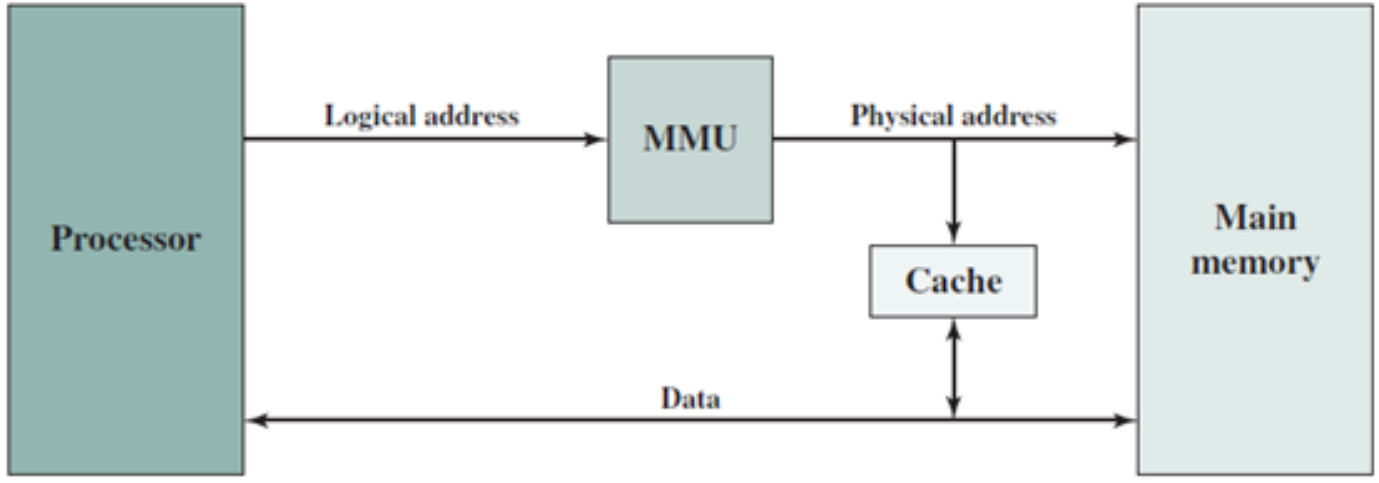
1. Clasificación
   - "La caché lógica utiliza direcciones virtuales; la física,

direcciones físicas."

- "La caché lógica puede ser más rápida pero requiere mayor gestión en cambios de contexto."



(a) Logical cache



(b) Physical cache

Figure 5.5 Logical and Physical Caches

1. Técnicas de Organización Se requiere un algoritmo que permita asociar los bloques de memoria principal con las líneas de caché, ya que hay menos líneas de caché que bloques de memoria. Además, es necesario un método para identificar qué bloque de memoria está ocupando una línea específica. Para organizar la caché, se utilizan tres técnicas principales: correspondencia directa, asociativa y asociativa por conjuntos, las cuales serán explicadas junto con ejemplos concretos. El texto explica tres técnicas para organizar la caché:
   - **Correspondencia directa:** Cada bloque de memoria principal se asigna a una línea de caché específica. Por ejemplo, el bloque 10 se asigna a la línea $10 \bmod 8 = 2$.
   - **Correspondencia asociativa:** Cualquier bloque puede ocupar cualquier línea de caché, lo que ofrece más flexibilidad, pero es más lento de buscar.
   - **Correspondencia asociativa por conjuntos:** La caché se divide en conjuntos, y cada bloque puede ocupar cualquier línea dentro de un conjunto específico. Por ejemplo, el bloque 10 se asigna al conjunto $10 \bmod 4 = 2$.

   Estas técnicas optimizan la asignación de memoria y el uso eficiente de la caché.

1. Tipos de Algoritmos Una vez llena la caché, se debe reemplazar un bloque existente para introducir uno nuevo. En correspondencia directa, no hay elección, ya que cada bloque tiene una línea específica. En técnicas asociativas, se requieren algoritmos de sustitución implementados en hardware para alta velocidad.(Stallings, 2006)
   1.1 LRU (Least Recently Used)
   1.2 FIFO (First-In-First-Out
   1.3 LFU (Least Frequently Used
   1.4 Aleatoria

- Casos de reemplazo en caché
  1. Casos de reemplazo en caché
  2. Casos de reemplazo en caché
- Problemas al reemplazar bloques
  1. Acceso múltiple a la memoria principal
  2. Sistemas multiprocesado
- Sistemas multiprocesado
  1. Escritura inmediata
  2. Postescritura
- Estadísticas de escritura

- Vigilancia del bus con escritura inmediata
- Transparencia hardware
- Memoria excluida de caché
- Tamaño de línea de caché:
- Efectos al aumentar el tamaño del bloque:
  1. Reducción de bloques en caché
  2. Mayor distancia de las palabras adicionales:
- Relación compleja entre tamaño y tasa de aciertos

Inicialmente, los sistemas contaban con una sola caché, pero con el tiempo se ha vuelto común utilizar múltiples cachés. Este diseño incluye consideraciones como el número de niveles de caché y el uso de cachés unificadas o separadas. Las cachés separadas evitan la competencia entre instrucciones y datos, mejorando el rendimiento en sistemas avanzados.

Cachés Multinivel Las cachés on-chip, integradas en el procesador, reducen el uso del bus externo y mejoran el rendimiento. Normalmente, se complementan con una caché externa (L2). Los diseños más recientes incluyen múltiples niveles: L1, L2 y, en algunos casos, L3. Estas cachés adicionales, ahora frecuentemente

on-chip, mejoran significativamente el rendimiento al reducir los tiempos de acceso a memoria, aunque complican aspectos como tamaño, políticas de escritura y algoritmos de reemplazo.

Caché Unificada Las cachés unificadas almacenan tanto instrucciones como datos en un único espacio, maximizando la tasa de aciertos al adaptarse dinámicamente a las necesidades de ejecución. Además, solo requieren un diseño único, simplificando la implementación.

Cachés Separadas Por otro lado, las cachés separadas para instrucciones y datos son preferidas en sistemas super-escalares y con segmentación de cauce. Este diseño elimina la competencia por recursos entre la ejecución de instrucciones y la unidad de datos, mejorando el rendimiento y optimizando la ejecución paralela de instrucciones.


Stallings, W. (2006). *Organización y arquitectura de computadores*. Pearson Educación. https://books.google.com.ec/books?id=C3HTAAAACAAJ


Stallings, W. (2022). *Computer Organization and arquitecture*. Pearson Global Editions.