

Arquitectura y Tecnología de Computadores

Tema 3 : El sistema de Memoria

4º Curso de Ingeniería de Telecomunicación

Departamento de Ingeniería Informática y Telecomunicaciones



Escuela Politécnica Superior



Contenidos



6.1.- Sistema de jerarquía de memoria.

6.2.- La memoria cache.

Principios de operación. Organización.

Esquemas de funcionamiento.

Algoritmos de sustitución.

6.3 El Sistema de memoria Virtual

Direccionamiento virtual: Paginación y Segmentación

La unidad de Gestión de memoria (MMU): Funcionamiento del TLB

6.4.- Mecanismos para mejorar el sistema de memoria.

Cache multinivel

Cache virtual vs cache real

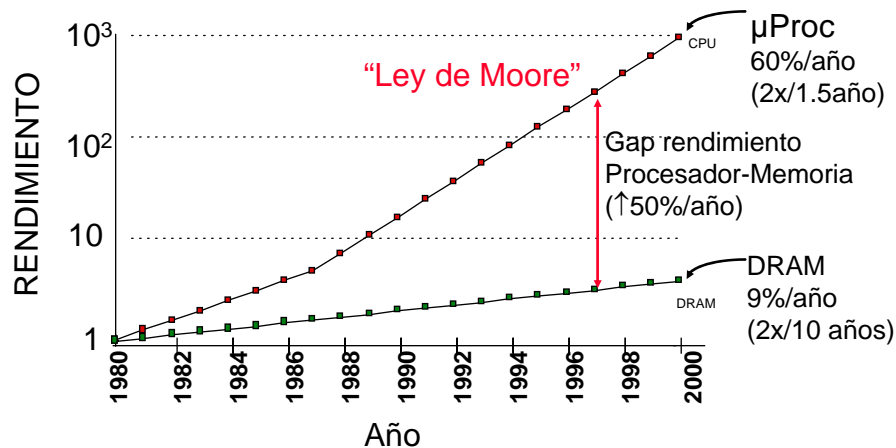
Ejemplos: La unidad cache de microprocesadores actuales



Tecnología de los dispositivos de memoria



¿Por qué se debe mejorar el sistema de memoria?



Universidad Autónoma de Madrid

El Sistema de Memoria Tema 3 / 2

Tecnología de los dispositivos de memoria



¿Cómo se puede mejorar un sistema de memoria?

Organizando una jerarquía de memoria

- ★ Memoria cache (nivel L1, L2,L3)

Aplicando el concepto de memoria virtual

- ★ MMU y TLB

Diseñando sistemas de conexionado CPU/MEM más

eficientes, con el objetivo de obtener transferencia de datos a la máxima velocidad



Universidad Autónoma de Madrid

El Sistema de Memoria Tema 3 / 3

Sistema de jerarquía de memoria



Capacidad
Tiempo de acceso
Coste

G.P.R.
100's Bytes
~ 1 ns

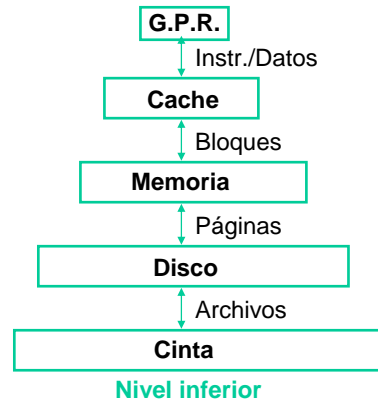
Cache
KBytes
~ 2-5 ns
 10^{-2} - 10^{-3} \$/bit

Memoria Principal
MBytes
~10-20 ns
 10^{-5} - 10^{-6} \$/bit

Disco
GBytes
~5-10 ms
 10^{-7} - 10^{-8} \$/bit

Cinta
TBytes
~sec-min
 10^{-10} \$/bit

Nivel superior



Gestor
Unidad de transfer.

Prog./Comp.
1-8 Bytes

Control cache
8-128 Bytes

Sist. operativo
512-4kBytes

Usuario
MBytes

Más rápida

Más grande



Universidad Autónoma de Madrid

El Sistema de Memoria Tema 3 / 4

Sistema de jerarquía de memoria



¿En qué se basa una jerarquía de memoria?

El principio de Localidad:

- ★ Los programas acceden a una porción relativamente pequeña del espacio de direcciones en cualquier instante de tiempo

Existen dos diferentes tipos de localidad:

- ❑ **Localidad Temporal:** Códigos y datos accedidos recientemente, tienen una gran probabilidad de volver a ser accedidos en un futuro cercano.
- ❑ **Localidad Espacial:** Si un código o dato es referenciado, códigos o datos cuyas direcciones de memoria estén cerca de la anterior, tenderán a ser referenciados pronto.



Universidad Autónoma de Madrid

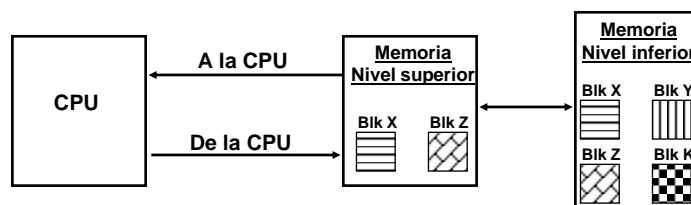
El Sistema de Memoria Tema 3 / 5

Sistema de jerarquía de memoria



¿Qué se debe tener en cuenta al diseñar una jerarquía de memoria?

- ¿Dónde colocar un bloque en el nivel superior?
- ¿Cómo encontrar un bloque si este está en el nivel superior?
- ¿Qué bloque debe ser sustituido en caso de fallo?
- ¿Qué estrategia se debe seguir cuando se modifica (escribe) un bloque en el nivel superior?



La memoria cache



Definiciones:

Cache: Memoria invisible, muy rápida y de pequeño tamaño entre el procesador y la memoria principal.

H, frecuencia de aciertos (Hit rate): % de referencias a memoria que se encuentran en la cache.

F, frecuencia de fallos (Miss rate): % de referencias a memoria que no se encuentran en la cache.
 $F = 1 - H$

t_c , Tiempo de acceso a la cache: tiempo necesario para acceder a un dato de la cache.

t_b , penalización por fallo: tiempo necesario para reemplazar un bloque desde el nivel inferior, incluye el tiempo para llevarlo al procesador.

El tiempo medio de acceso al sistema de memoria es: $\bar{t}_{acc} = t_c + (1 - H)t_b$

Un fallo en la cache **AUMENTA** el tiempo de ejecución

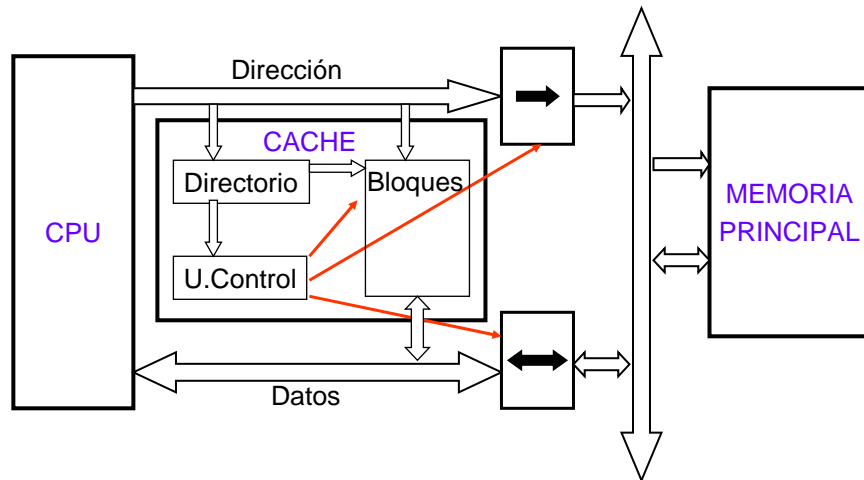
de la instrucción implicada $T_{CPU} = NI \times (CPI + CDM) \times T_{CICLO}$

$$CDM = \frac{\text{Accesos Memoria}}{NI} \times \text{Freq. fallos} \times \text{Penalización}$$



La memoria cache. Principios de operación

ATC



La memoria cache. Principios de operación

ATC

Directorio cache

- * Realizado en memoria asociativa o memoria CAM (*Content Addressable Memory*).
- * Señala si el dato solicitado por el procesador se encuentra en la cache. Si acierto (*match*) señala la dirección del bloque donde se encuentra el dato.

Bloques cache

- * Realizado en memoria SRAM
- * Cada bloque contiene un número determinado de bytes a los que se accede en el caso de un acierto cache.

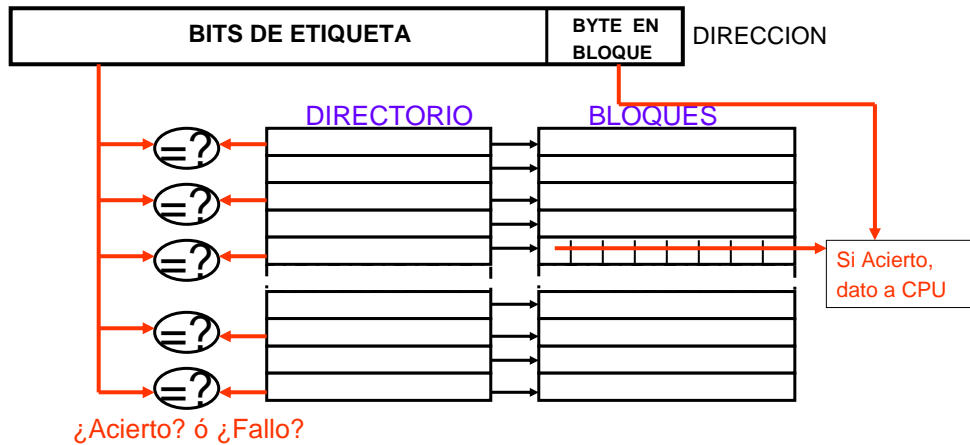
Unidad de control

- * Genera las señales de control para el funcionamiento de la unidad cache.

La memoria cache. Organización



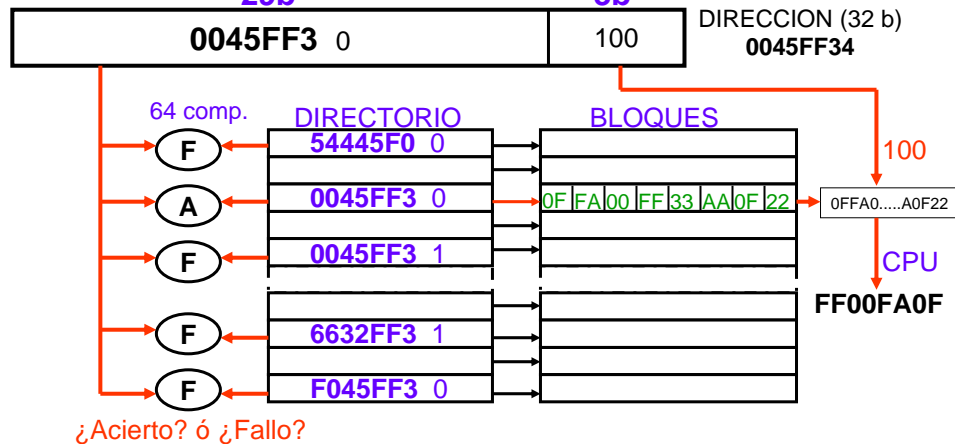
Cache Completamente asociativa



La memoria cache. Organización



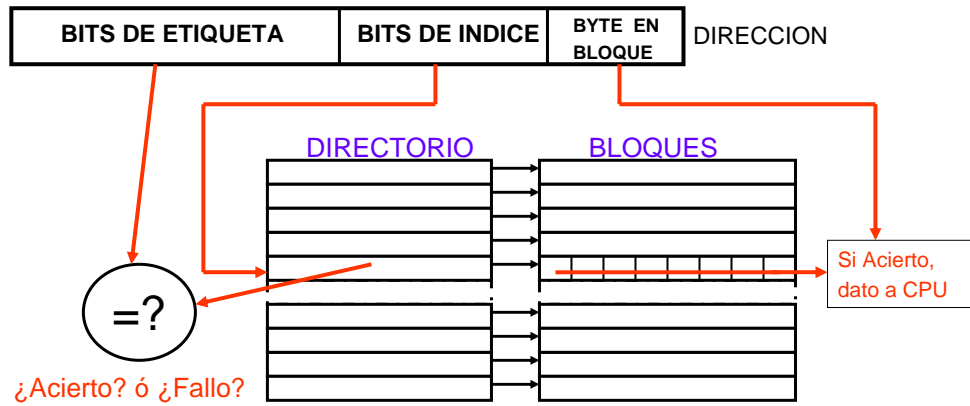
Ejemplo: Cache complet. asociativa de 512 Bytes, 8By/bloque



La memoria cache. Organización



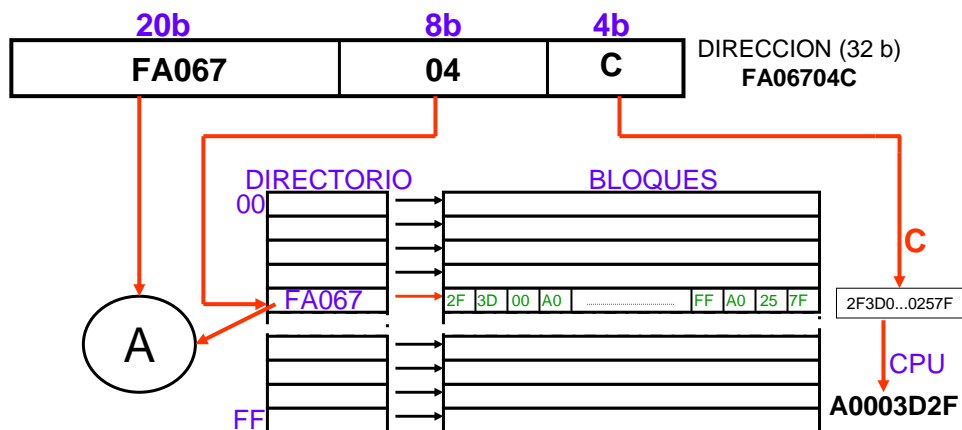
Cache de correspondencia directa (Asociativa de 1 vía)



La memoria cache. Organización



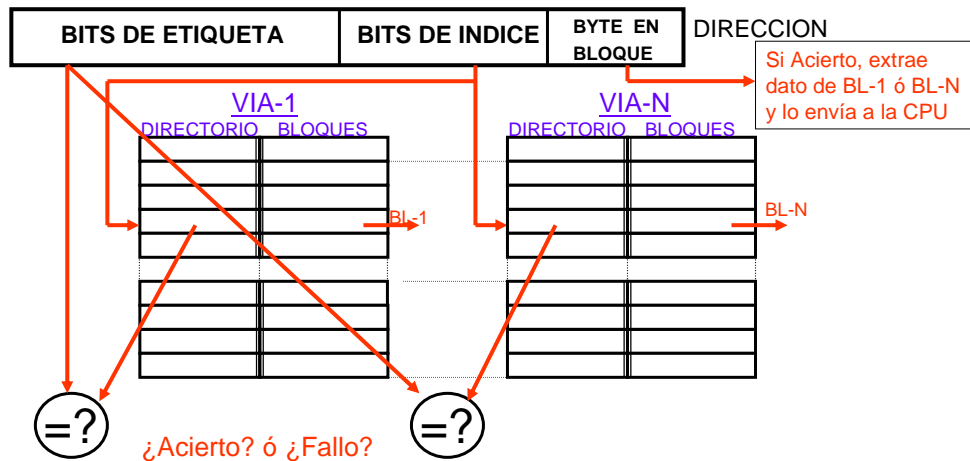
Ejemplo: Cache de correspondencia directa de 4kB y 16 By/Bloq



La memoria cache. Organización



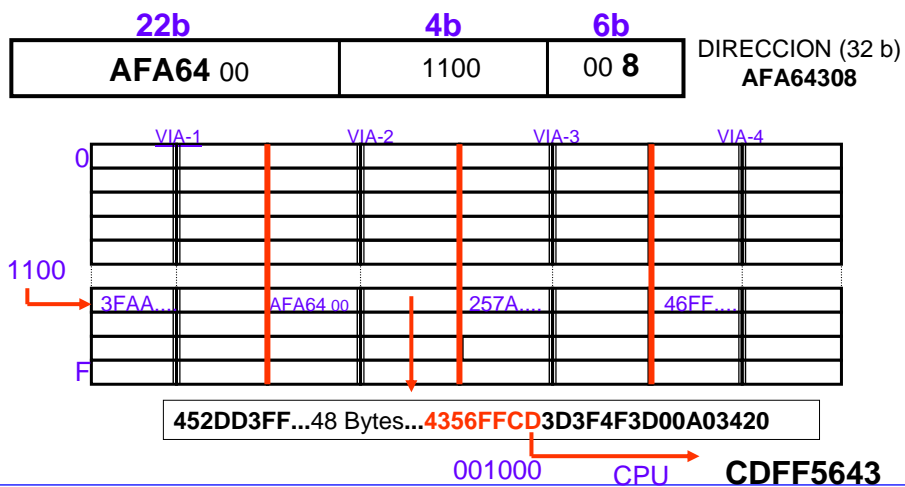
Cache asociativa por vías (N vías)



La memoria cache. Organización



Ejemplo: Cache de 4kB con 64 By/Bloq y asociativa de 4 vías



La memoria cache. Organización



Cache Correspondencia Directa VS Asociativa por N Vías

- * CD necesita 1 comparador AV necesita N comparadores
- * CD puede obtener los datos antes de producirse un acierto o fallo. AV dispone de los datos sólo tras la comparación, se necesita intercalar un multiplexor para seleccionar los datos de la vía en la que se produce el acierto.

¿Qué misión tienen los bits de validez (BV)?

- * Si hay 1 BV por cada bloque \Rightarrow En caso de acierto si BV='0', el bloque no contiene datos válidos.
- * Si hay 1BV por cada palabra en el bloque \Rightarrow En caso de acierto, la palabra con BV a '0' no contiene datos válidos.

¿Qué misión tiene el bit de modificación o suciedad (M)?

- * Siempre asociado a un bloque. Si M='1' \Rightarrow ha habido al menos una escritura en la cache que no ha sido escrita en memoria.



La memoria cache. Esquemas de funcionamiento



Operaciones de captura

- * Captura a la demanda: Se carga un bloque (entero o parte) sólo cuando se referencia un dato en él contenido y se produce un fallo.
- * Captura anticipativa: Se carga un bloque (entero) antes de ser referenciado.
- * Captura selectiva: No siempre se carga un dato cuya referencia ha originado un fallo. Ejem: direcciones "no cacheables".

Operaciones de escritura y actualización de memoria

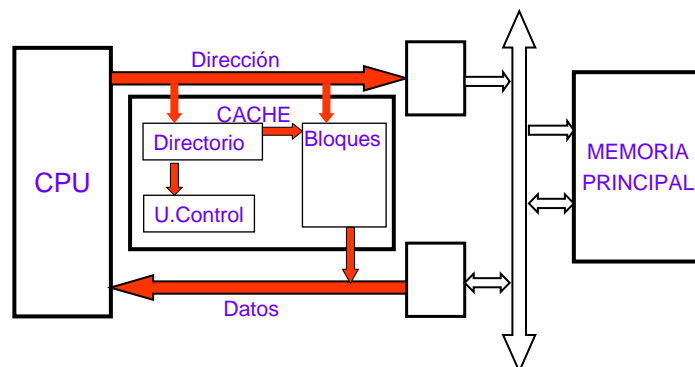
- * Escritura Directa (ED) (*Write Through*): Se escribe al tiempo en la cache y en la memoria principal.
 - Dos estrategias: EDAAE y EDSAAE.
- * Post-Escritura (PE) (*Write Back*): Se actualiza la memoria principal, sólo al reemplazar un bloque. Dos modalidades dependiendo del bit M.



La memoria cache. Esquemas de funcionamiento

ATC

1.- ED y PE: Operación de ACIERTO en lectura



Cache⇒CPU

$$t_{acc}^{AL} = (H)(1-w)t_C$$



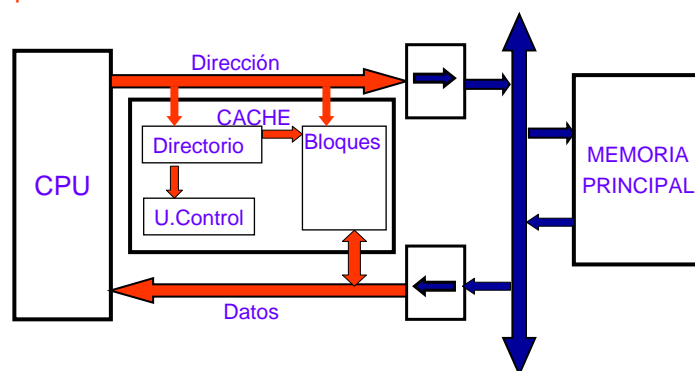
Universidad Autónoma de Madrid

El Sistema de Memoria Tema 3 /18

La memoria cache. Esquemas de funcionamiento

ATC

2.- ED: Operación de FALLO en lectura



Cache⇒CPU

MEM⇒Cache

$$t_{acc}^{FL} = (1-H)(1-w)t_C + (1-H)(1-w)t_B$$



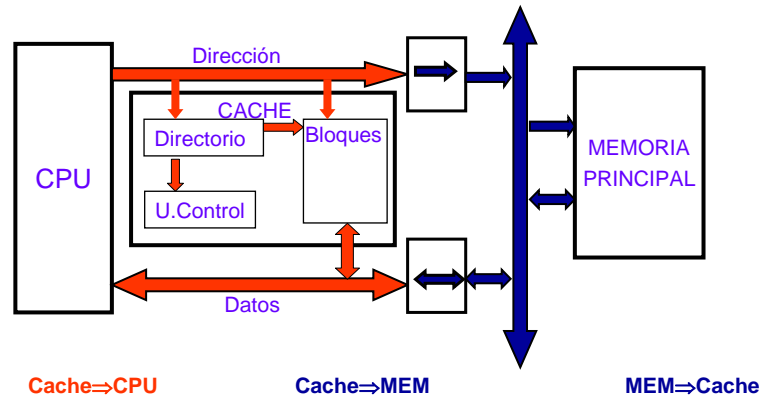
Universidad Autónoma de Madrid

El Sistema de Memoria Tema 3 /19

La memoria cache. Esquemas de funcionamiento

ATC

2.- PE: Operación de FALLO en lectura



$$t_{acc}^{FL} = (1-H)(1-w)t_C + (1-H)(1-w)w_M t_B + (1-H)(1-w)t_B$$



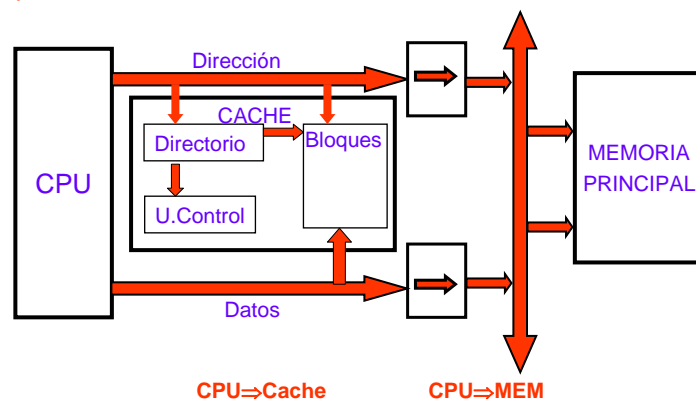
Universidad Autónoma de Madrid

El Sistema de Memoria Tema 3 /20

La memoria cache. Esquemas de funcionamiento

ATC

3.- ED: Operación de ACIERTO en escritura



$$t_{acc}^{AW} = (H)(w)t_m = (H)(w)t_C + (H)(w)(t_m - t_C)$$



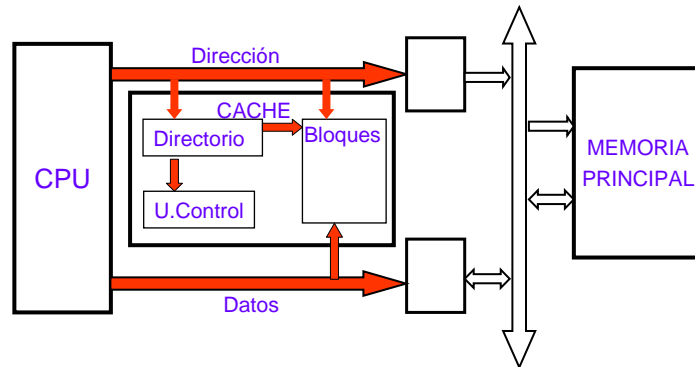
Universidad Autónoma de Madrid

El Sistema de Memoria Tema 3 /21

La memoria cache. Esquemas de funcionamiento

ATC

3.- PE: Operación de ACIERTO en escritura



CPU⇒Cache

$$t_{acc}^{AW} = (H)(w)t_C$$



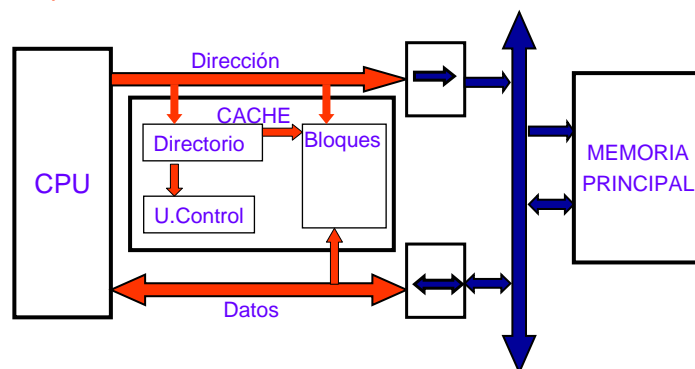
Universidad Autónoma de Madrid

El Sistema de Memoria Tema 3 /22

La memoria cache. Esquemas de funcionamiento

ATC

2.- EDAE: Operación de FALLO en escritura



CPU⇒Cache

CPU⇒MEM

MEM⇒Cache

$$t_{acc}^{FW} = (1-H)(w)t_C + (1-H)(w)(t_m - t_C) + (1-H)(w)t_B$$



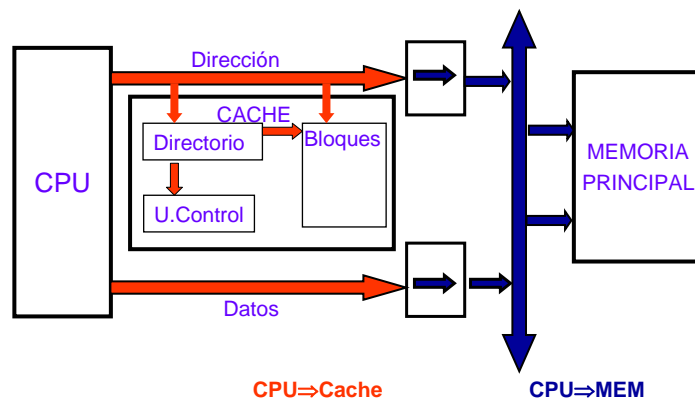
Universidad Autónoma de Madrid

El Sistema de Memoria Tema 3 /23

La memoria cache. Esquemas de funcionamiento

ATC

2.- EDSAE: Operación de FALLO en escritura



$$t_{acc}^{FW} = (1 - H)(w)t_m = (1 - H)(w)t_c + (1 - H)(w)(t_m - t_c)$$



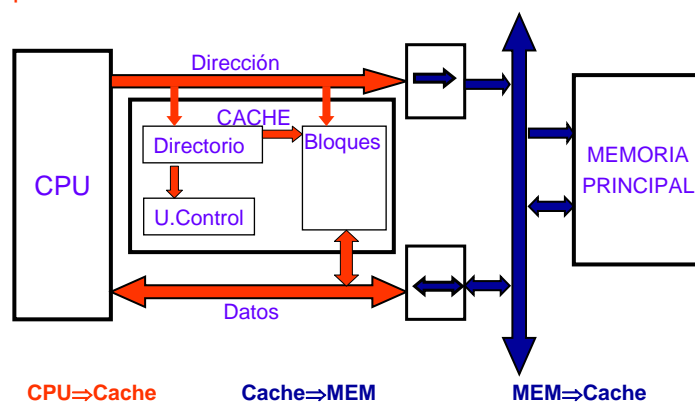
Universidad Autónoma de Madrid

El Sistema de Memoria Tema 3 /24

La memoria cache. Esquemas de funcionamiento

ATC

2.- PE: Operación de FALLO en escritura



$$t_{acc}^{FW} = (1 - H)(w)t_c + (1 - H)(w)w_M t_B + (1 - H)(w)t_B$$



Universidad Autónoma de Madrid

El Sistema de Memoria Tema 3 /25

La memoria cache. Esquemas de funcionamiento



Resumen de estrategias

Escritura Directa (EDAE):

$$\overline{t_{acc}} = t_C + w(t_m - t_C) + (1 - H)t_B$$

Escritura Directa (EDSAE):

$$\overline{t_{acc}} = t_C + w(t_m - t_C) + (1 - H)(1 - w)t_B$$

Post Escritura (PE):

$$\overline{t_{acc}} = t_C + (1 - H)(1 + w_M)t_B$$



La memoria cache. Esquemas de funcionamiento



Algoritmos de reemplazamiento

- Aleatorio
- FIFO (*First In First Out*)
- LRU (*Least Recently Used*)
 - Usando bits de control (N bloques \Rightarrow N! estados)
 - Usando contadores asociados a cada bloque
 - Usando una matriz de referencia(para un número N de bloques
necesita $(N \times (N-1))/2$ bits)

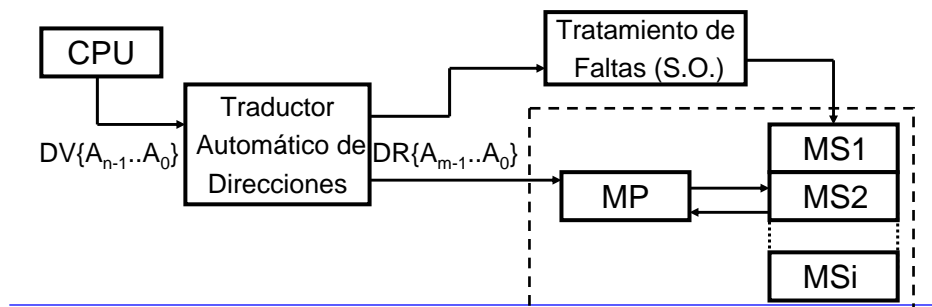


¿Qué se entiende por memoria virtual?

- El espacio de direcciones que ve una determinada tarea. Este espacio reside en el nivel inferior de la jerarquía de memoria.

¿Qué se entiende por memoria física o real ?

- El espacio de direcciones donde se ejecuta una determinada tarea. Este espacio reside en los niveles superiores de la jerarquía de memoria.



Conceptos: ¿Cuál es el tamaño de los bloques de información que deben ser transferidos entre la memoria principal (MP) y la memoria secundaria (MS)?

- **Organización por páginas:** La memoria virtual y física están siempre divididas en bloques de igual tamaño denominados páginas. Una tarea ocupa un determinado número de páginas.
- **Organización por segmentos:** La memoria virtual y física se dividen en bloques de diferentes tamaños denominados segmentos. El tamaño y número de los segmentos dependen de la tarea que se desea ejecutar en cada instante. La estructura de la memoria varía con cada tarea a ejecutar.
- **Organización por segmentos paginados:** Los segmentos se dividen en páginas, la unidad de transferencia entre MS y MP es la página de tamaño único.

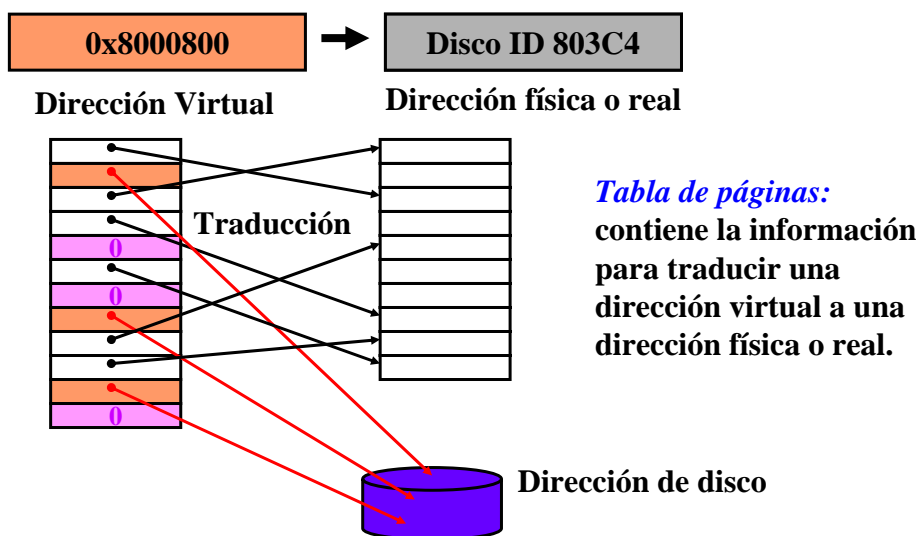
¿Porqué es mejor paginación que segmentación?

- ❑ La partición en páginas es transparente al programador.
- ❑ La dirección virtual está formada por una sola palabra, mientras que en segmentación está formada por dos, segmento y desplazamiento.
- ❑ Es fácil reemplazar un bloque ya que todos tienen el mismo tamaño.
- ❑ Elimina la fragmentación externa de memoria aunque no elimina completamente la interna.

¿Porqué es mejor segmentación que paginación?

- ❑ Facilita las operaciones con la memoria, ya que programas y tablas de datos tienen diferente tamaño.
- ❑ Permite compartir zonas de memoria entre procesos.
- ❑ Permite proteger zonas de memoria.
- ❑ Permite la existencia de varios espacios de direcciones, por lo que facilita la separación entre código y datos y la modificación independiente de programas.

Paginación: Mapeo de MV y MP



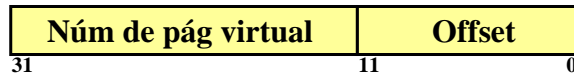
Paginación: Terminología del sistema de MV



Los bloques se denominan **Páginas**

* Una dirección virtual consiste en:

- Un número de página virtual
- Un campo de offset dentro de la página (LSB de la dirección)



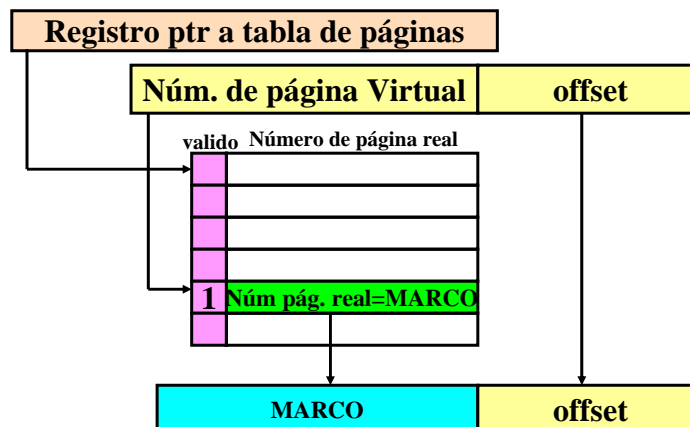
- * Los fallos se denominan **falta de página** y se gestionan como excepciones.
- * La **tabla de páginas** contiene la información para traducir una dirección virtual a una dirección física o real.



Tabla de páginas

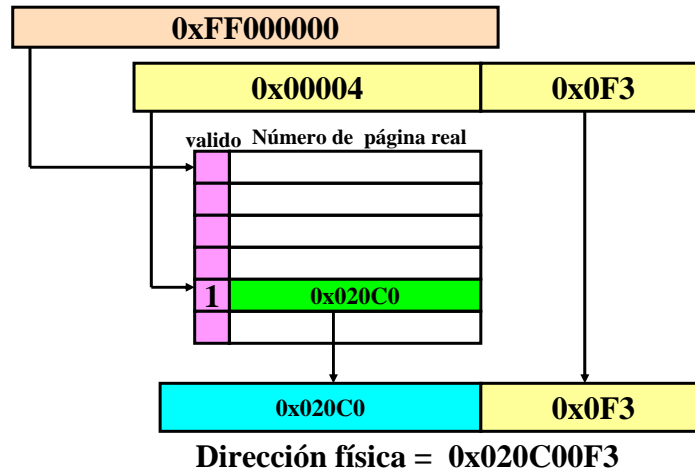


Conceptos: ¿Cuáles son los mecanismos automáticos de traducción de la dirección virtual (DV) a la dirección física o real (DR)? => **Tabla de páginas**



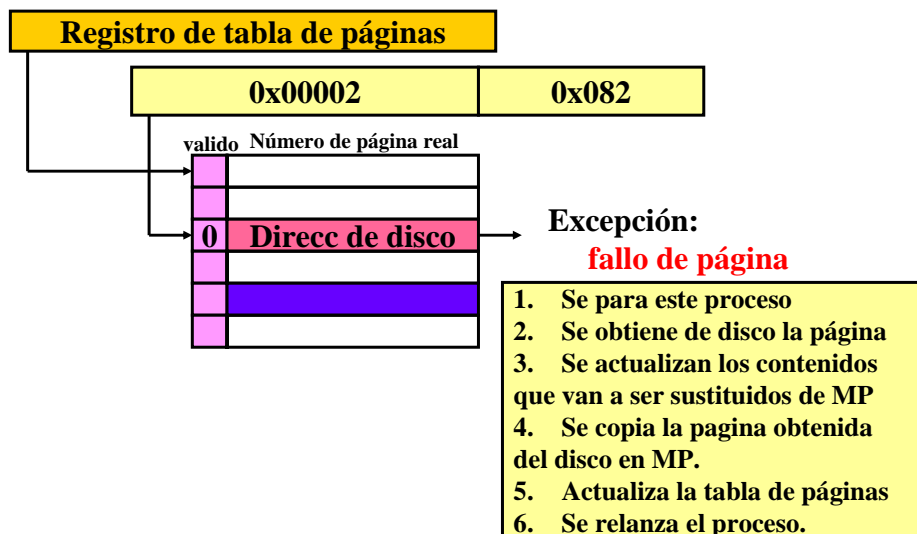
Ejemplo de traducción de direcciones

ATC



Ejemplo de traducción de direcciones

ATC

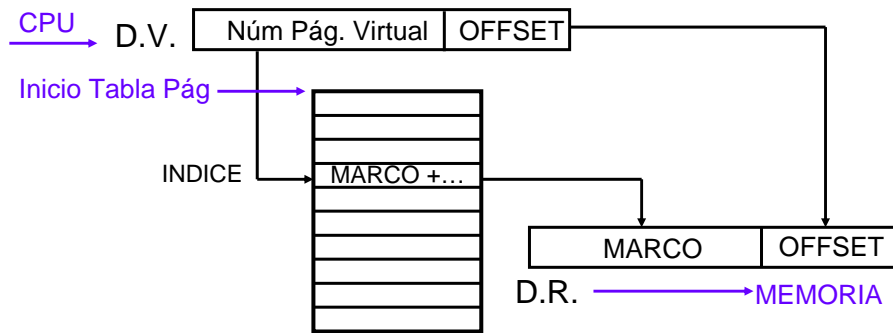


Paginación



Componentes y Recursos Necesarios

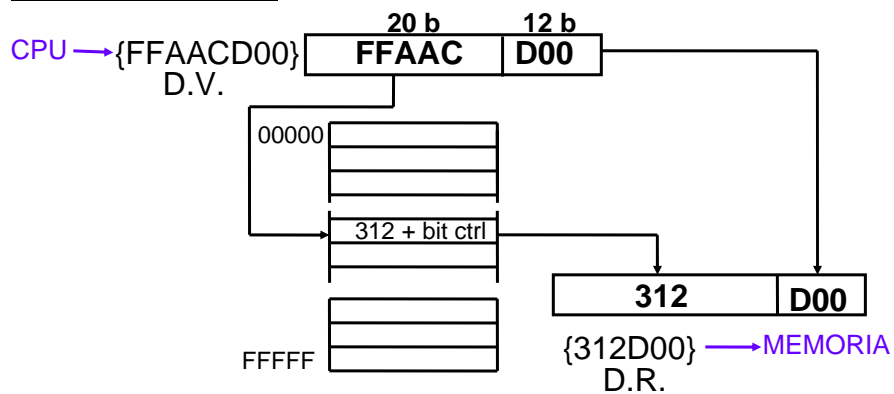
a) Tabla de páginas de sustitución directa (*Direct mapping*)



Paginación



Ejemplo: Memoria: Virtual: 4GB (2^{32}), Real: 16MB (2^{24}). Páginas: 4kB (2^{12})
Sustitución Directa



¡¡Como mínimo se necesita SRAM $2^{20} \times 12$ bits!!

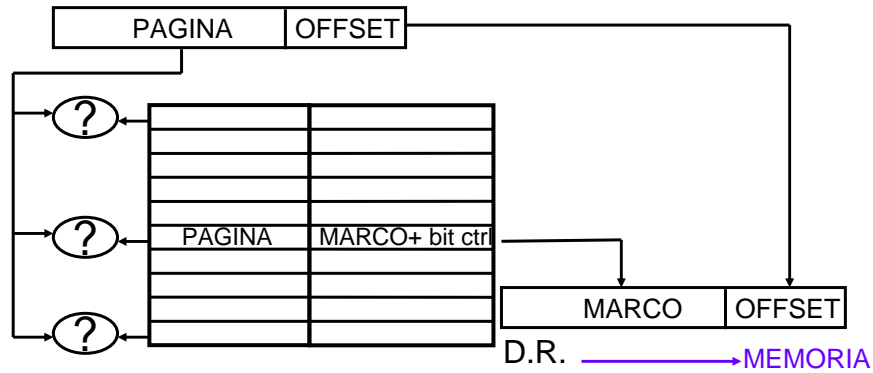


Paginación



b) Tabla de paginas por sustitución asociativa completa (*Fully associative mapping*)

CPU → D.V.



Universidad Autónoma de Madrid

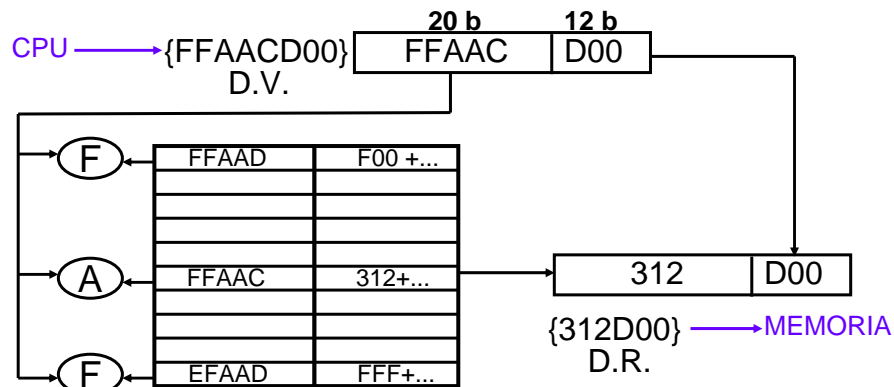
El Sistema de Memoria Tema 3 /38

Paginación



Ejemplo: Memoria: Virtual: 4GB (2^{32}), Real: 16MB (2^{24}). Páginas: 4kB (2^{12})

Sustitución asociativa completa



¡¡Necesita un mínimo de CAM $2^{20} \times 20$ bits y SRAM $2^{20} \times 12$ bits!!



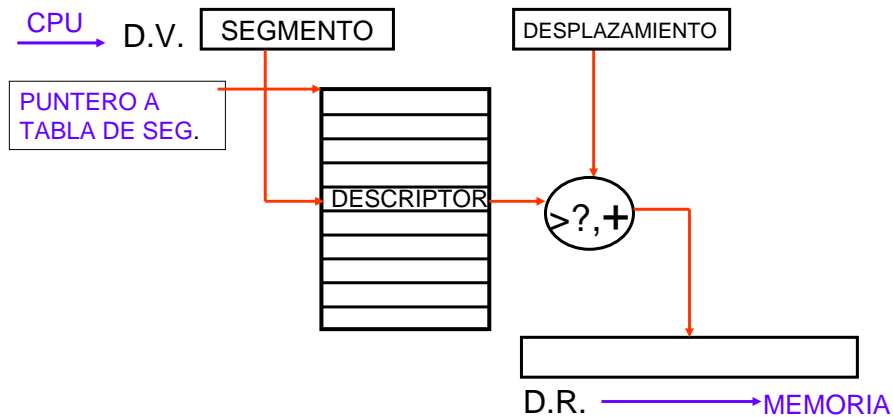
Universidad Autónoma de Madrid

El Sistema de Memoria Tema 3 /39

Segmentación

ATC

¿Cómo se traduce la dirección virtual en real?



Segmentación

ATC

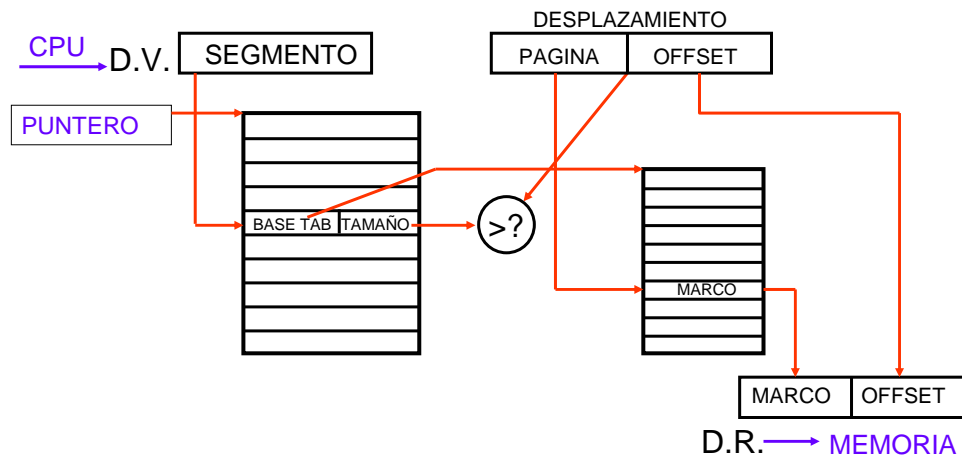
¿Qué contiene el descriptor de segmento?

- ★ La dirección base del segmento: **Sumada al desplazamiento indica la dirección de memoria real referenciada**
- ★ El tamaño del segmento: **Debe ser superior al desplazamiento**
- ★ Los bits de control de la MMU
 - **Bit de presencia en memoria real**
 - **Bit de protección contra escritura (segmento código)**
 - **Bit de exclusión o para limitar accesos (seg. sistema)**
- ★ Los bits para los algoritmos de reemplazamiento
 - **LRU o similares**

Segmentación-Paginada

ATC

¿Cómo se traduce la dirección virtual en real?



Universidad Autónoma de Madrid

El Sistema de Memoria Tema 3 /42

Mejorar la gestión de la memoria virtual.

ATC

¿Cómo reducir el coste del sistema de traducción automática de direcciones sin perder rendimiento?

- ❑ Optimizando el tamaño de página
 - ❑ Pag. Pequeñas.- Reducen la fragmentación, pero aumentan el trasiego (*thrashing*)
 - ❑ Pag. Grandes.- Aumenta el rendimiento en el cambio MP↔MS
- ❑ Utilizando traductores de menor tamaño, denominados TLB's (*Translation Look-Aside Buffers*)
 - ❑ Completamente asociativos
 - ❑ Asociativos por vías (*set associative mapping*)
- ❑ Utilizando sistemas de paginación multinivel
 - ❑ Se guardan en memoria rápida SRAM, sólo las páginas que con mayor probabilidad van a ser utilizadas
- ❑ Un sistema mixto con ambas opciones.



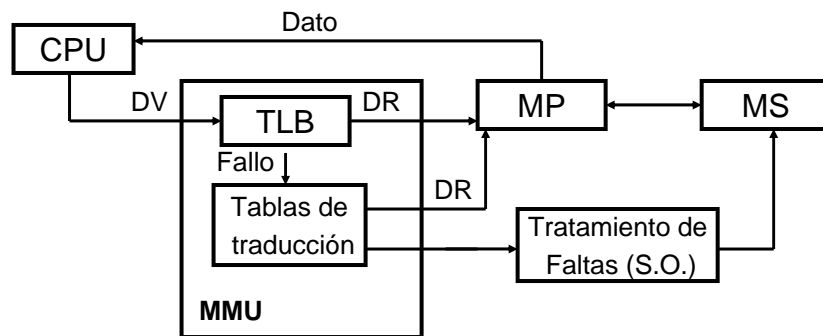
Universidad Autónoma de Madrid

El Sistema de Memoria Tema 3 /43

Mejorar la memoria virtual: TLB

ATC

- Intercalando una unidad de memoria (cache) que realice esta función denominada TLB (*Translation Look-Aside Buffers*)



Mejorar la gestión de la memoria virtual.

ATC

¿Cuál es la información contenida en un elemento del TLB o de la tabla de páginas?

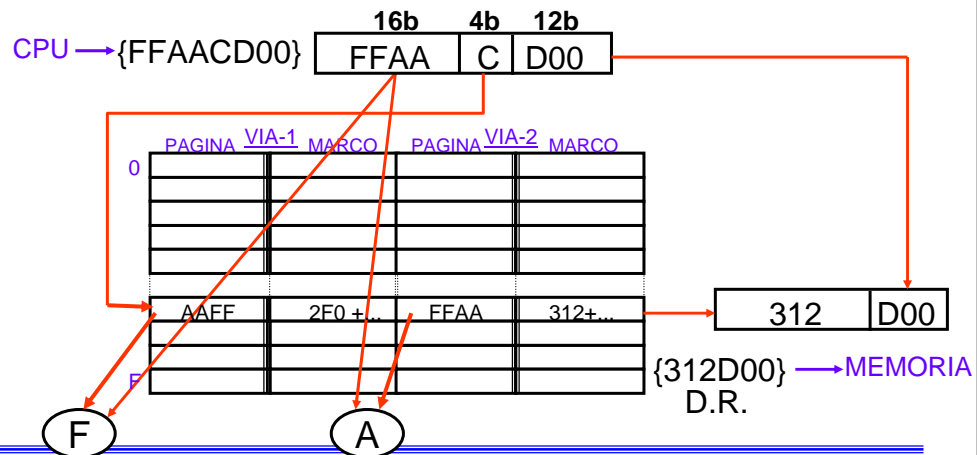
- * El marco de página: con el offset permite obtener la dirección real
- * Los bits para el control de la MMU
 - Bit de presencia: activo a '1' indica la presencia de la página referenciada en memoria principal
 - Bit de uso: activo a '1' indica que algún elemento de esa página ha sido referenciado
 - Bit de modificación: activo a '1' indica que una dirección de la página ha sido modificada (escrita)
 - Bit/s de protección o privilegios: supervisor, sólo lectura, no cacheable, otros...
 - Bits de reemplazamiento: necesarios para aplicar alguno de los algoritmos de reemplazamiento, Reloj, LRU, etc...

La gestión de la memoria virtual: TLB

ATC

Ejemplo: Memoria: Virtual: 4GB (2^{32}), Real: 16MB (2^{24}). Páginas: 4kB (2^{12})

TLB asociativo por conjuntos de 2 vías y 16 entradas/vía



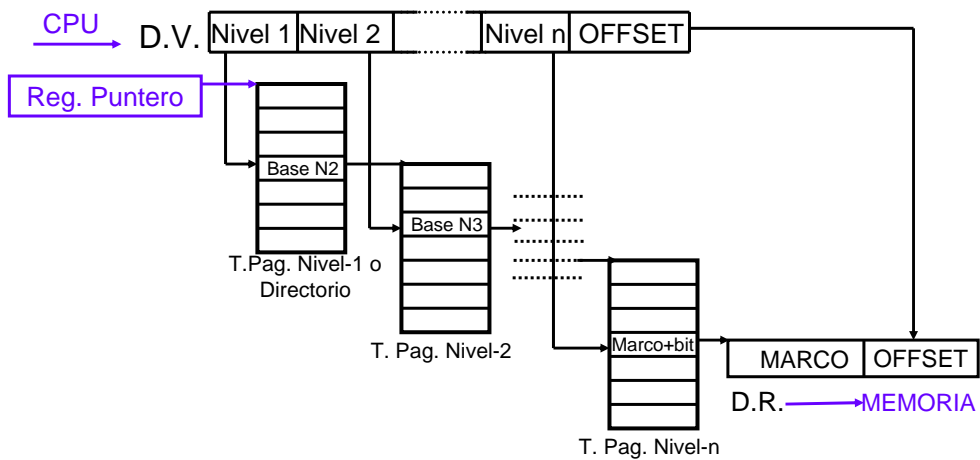
Universidad Autónoma de Madrid

El Sistema de Memoria Tema 3 /46

Mejorar la gestión de la memoria virtual: Paginación multinivel

ATC

□ Sistema de paginación multinivel



Universidad Autónoma de Madrid

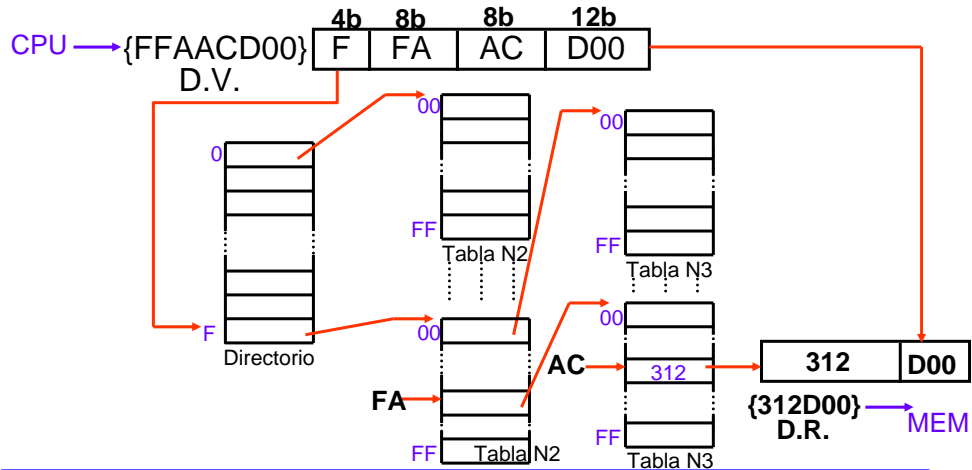
El Sistema de Memoria Tema 3 /47

Paginación multinivel



Ejemplo: Memoria Virtual: 4GB (2^{32}), Real: 16MB (2^{24}). Páginas: 4kB (2^{12})

Sistema multipaginado de 3 niveles: (N1: 16 entradas y N2 y N3 con el mismo número de entradas. Tamaño del descriptor 4 bytes en todos los niveles)



Mejoras del sistema de memoria



Mecanismos para mejorar el rendimiento Cache

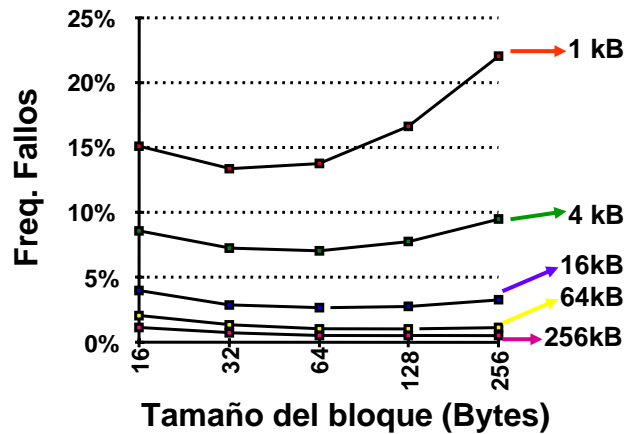
- ❑ Reducir la tasa de fallos
 - ❑ Aumentando el tamaño del bloque
 - ❑ Aumentando el número de vías
 - ❑ Caches independientes vs unificada
- ❑ Reducir la penalización por fallos
 - Intercalar *buffer* de escritura
 - Sistema cache de dos niveles
- ❑ Reducir el tiempo de acceso a la cache



Mejoras sistema de memoria

ATC

Reducir la tasa de fallos: Aumentar el tamaño del bloque



Universidad Autónoma de Madrid

El Sistema de Memoria Tema 3 /50

Mejoras sistema de memoria

ATC

Reducir la penalización por fallos: Caches de dos niveles

- Compromiso entre realizar caches mas rápidas o caches mas grandes

Definiciones

- Frecuencia local de fallos (F_{L1}).- Fallos en el nivel i / Accesos al nivel i
- Frecuencia global de fallos (F_G).- Total Fallos / Accesos solicitados por la CPU.

$$F_G = F_{L1} \times F_{L2}$$

$$t_{acc} = t_{L1} + F_{L1}(t_{L2} + F_{L2}t_{B2})$$



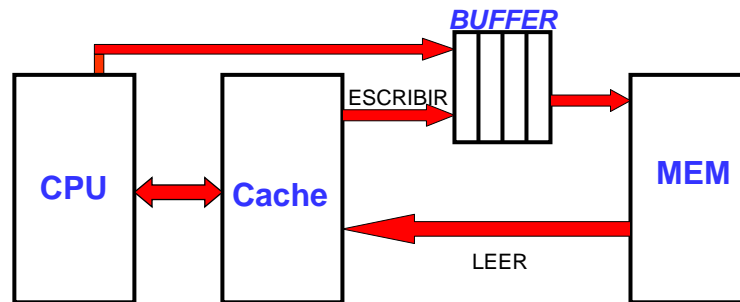
Universidad Autónoma de Madrid

El Sistema de Memoria Tema 3 /51

Mejoras sistema de memoria

ATC

Reducir la penalización por fallos: "Buffer" de escritura



Escritura Directa (EDAE): $\overline{t_{acc}} = t_C + (1 - H)t_B$

Escritura Directa (EDSAE): $\overline{t_{acc}} = t_C + (1 - H)(1 - w)t_B$

Post Escritura (PE): $\overline{t_{acc}} = t_C + (1 - H)t_B$



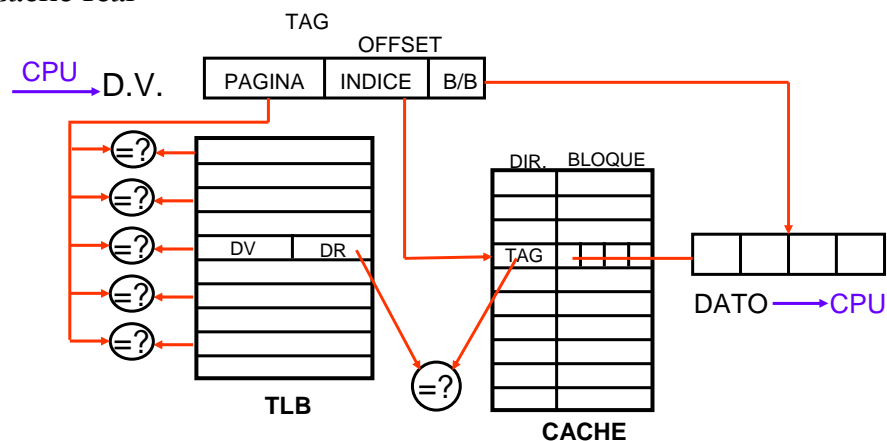
Universidad Autónoma de Madrid

El Sistema de Memoria Tema 3 /52

Mejoras sistema de memoria Caches virtuales y reales

ATC

Cache real



Universidad Autónoma de Madrid

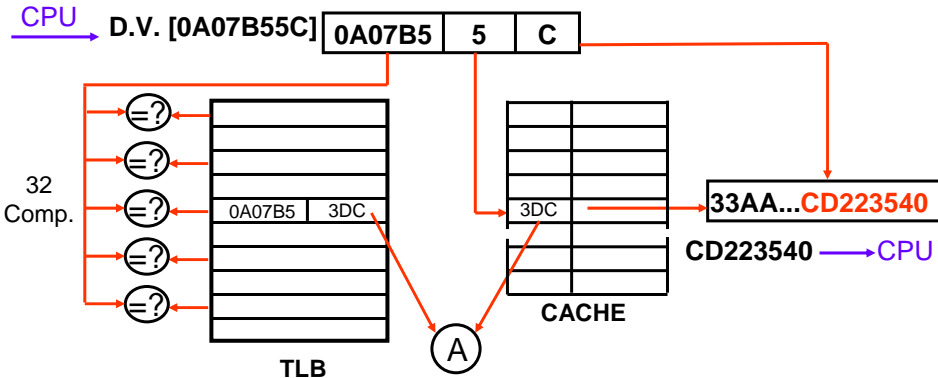
El Sistema de Memoria Tema 3 /53

Caches virtuales y reales

ATC

Ejemplo.- DV: 32b; DR: 20b; Pag: 256 Bytes; TLB CA de 32 Entradas; Cache CD 256

Bytes, con 16 B/B.



Funcionamiento óptimo en cache real de correspondencia directa se consigue con
 “tamaño de la unidad cache igual que el tamaño de la página”

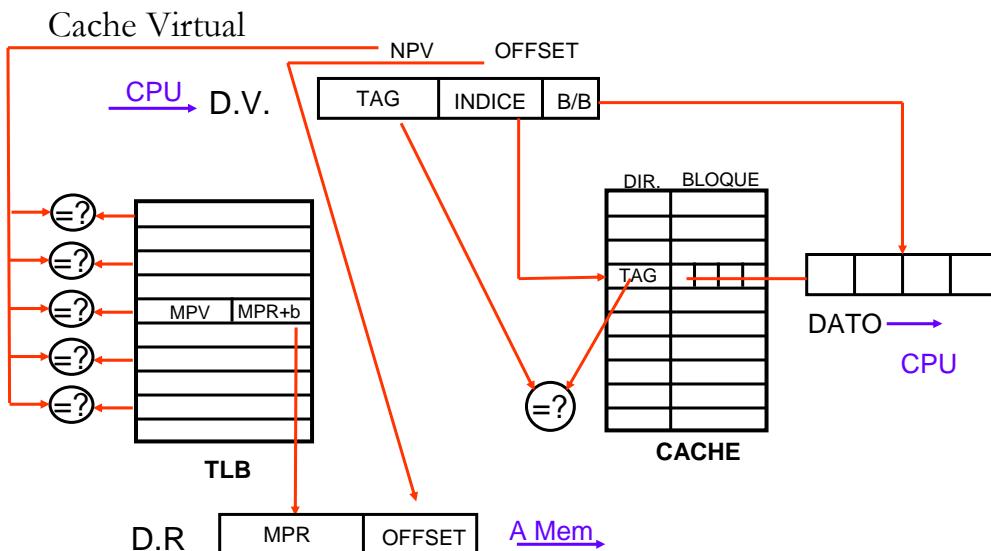


Universidad Autónoma de Madrid

El Sistema de Memoria Tema 3 /54

Caches virtuales y reales

ATC



Universidad Autónoma de Madrid

El Sistema de Memoria Tema 3 /55

Ejemplos: La unidad cache del Pentium



Pentium (1993)

Sistema cache interno (L1), con dos unidades independientes para instrucciones y datos de 8 kBytes cada una.

Cache de asociativa de 2 vías con 32 Bytes/Bloque

Estrategia de post-escritura. Posible configuración a escritura directa.

Bus de enlace entre CPU y cache-instrucciones de 256 bits. Bus de enlace entre CPU y cache-datos de 32+32 (2 ALU-INT) o 64 bits (ALU-FP)

Cache externa L2 256-512 kBytes asociativa de 2 vías, con 32,64, ó 128 Bytes/Bloque.

Protocolo MESI para el control de coherencia de 2 bits.



Ejemplos: La unidad cache del Pentium



Pentium P-II (1997)

Sistema cache interno (L1), no unificado 16+16 kB. LRU

Cache de asociativa de 4 vías con 32 Bytes/Bloque.

Cache de datos *Non-Blocking*. Protocolo MESI. EDAE.

Cache externa L2 0kB(Celeron)-512kB(PIII)-2MB(Xeon).
4 vías, con 32 Bytes/Bloque.

Pentium III (1999)

Sistema cache interno (L1) igual que PII

Cache L2 externa 512kB(PIII)-2MB(Xeon) 4 vías, con 32 Bytes/Bloque.

Cache L2 interna 128kB*(Celeron)-256kB(PIII)-2MB(XeonA)
8 vías, con 32 Bytes/Bloque.



Ejemplos: La unidad cache del Pentium



PowerPC 601 (1995)

Sistema cache con una cache unificada de 32 kBytes asociativa de 8 vías y 32 Bytes/Bloque

PowerPC 620 (1996)

Sistema cache interno con dos unidades independientes para datos e instrucciones de 32 kBytes cada una. Asociativas de 8 vías y 64 Bytes/Bloque.

Bus de enlace entre CPU y cache-instrucciones de 128 bits y de 64 bits entre CPU y cache-datos.

Protocolo MESI para el control de coherencia de 3 bits



Ejemplo: La MMU del Pentium



Cuatro formas de gestionar la memoria

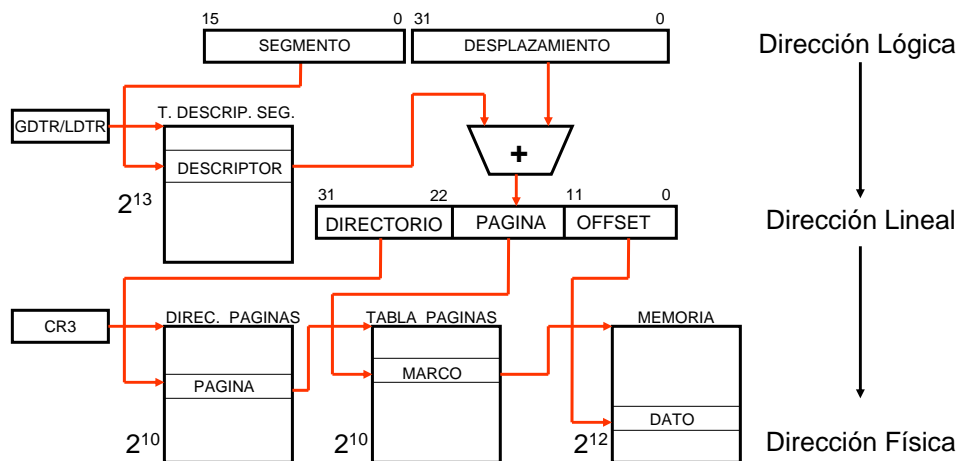
- * Sin segmentos y sin páginas: **Memoria lógica (virtual) = Memoria Física**
- * Memoria segmentada pura: **16b segmento y 32b desplazamiento. Espacio virtual de 64TB (16 kTablas de 4GB, 8 globales y 8 locales). Espacio físico 4GB**
- * Memoria paginada pura: **Memoria lineal de 4 Gb, en páginas de 4kB ó 4MB**
- * Segmentos paginados: **DV {Seg(16b)+Despl(32b)}; D.Lineal (32b); DR (32b)**

Dispone de una unidad **TLB** completamente asociativa de 32 entradas

Cada descriptor de la tabla de segmentos ocupa 64 bits. Si se utiliza paginación, el directorio de páginas del proceso en curso, siempre se encuentra en memoria principal, cada elemento del directorio o de la tabla de páginas ocupa 32 bits



Ejemplo la MMU del Pentium



Ejemplo: la MMU del Pentium



DESCRIPTOR DE SEGMENTO

63..56	55	54	53	52	51..48	47	46-45	44	43..40	39..16	15..00
Base[31:24]	G	D/B	0	AVL	Limite[19:16]	P	DPL	S	Tipo	Base[23:00]	Limite[15:00]

Limite: Tamaño máximo del segmento (16 b)
Tipo: Tipos y atributos de acceso (4 b)
DPL: Nivel de privilegio del descriptor (2 b)
AVL: Reservado para el programador (1 b)
G: Granularidad (tamaño x1B/x4kB) (1 b)

Base: Dirección base del segmento (32 b)
S: DTipo de descriptor (sistema o código/datos) (1 b)
P: Bit de presencia (1 b)
D/B: Tamaño de operación por defecto (16/32) (1 b)

DESCRIPTOR DE PAGINA

31.....12	11...9	8	7	6	5	4	3	2	1	0
Dirección marco de página	AVL		P	S	D	A	P	C	D	P
							P	W	T	
							U	S	R	W
										P

P: Bit de presencia (1 b)
US: Bit Usuario/Supervisor (1 b)
PCD: Bit de inhabilitación cache (1 b)
D: Bit de modificación (1 b) (en tabla de 2 nivel)
AVL: Reservado para el programador (3 b)

RW: Bit de lectura/Escritura (1 b)
PWT: Bit para la estrategia de escritura (ED/PE) (1 b)
A: Bit de acceso o uso (1 b)
PS: Tamaño de página (4kB/4Mb) (1 b) (en tabla directorio)
Dirección del marco de página real (20 b)



Bibliografía



- * **COMPUTER ARCHITECTURE: Design and performance, 2ª Edition"**
Barry Wilkinson
Prentice Hall International. 2ª Edition. 1996.
- * **"COMPUTER ARCHITECTURE: A Quantitative Approach".** John L. Hennessy y David A. Patterson. 3rd Edition, Ed. Morgan Kaufmann 2001.
- * **"ESTRUCTURA Y DISEÑO DE COMPUTADORES interficie circuitería/programación".**
David A. Patterson y John L. Hennessy. Ed. Reverte 2000. Volumen. 1.
- * **"ORGANIZACIÓN Y ARQUITECTURA DE COMPUTADORES".** W. Stallings. 5ª Edición. Prentice Hall 2000.

Referencias WEB

- * Memoria cache: <http://www.slcentral.com/articles/00/10/cache/>

