

Organización de Computadoras

SEMANA 10

UNIVERSIDAD NACIONAL DE QUILMES

¿ Qué vimos?

- Mascaras
- Repeticiones controladas
- Arreglos
- Modo indirecto
- Q5

Hoy!

- Memorias:
 - Características
 - Memorias ROM
 - Jerarquía de memorias
- Caché:
 - Motivación
 - ¿Qué?
 - Organización
 - Función de mapeo

Memorias

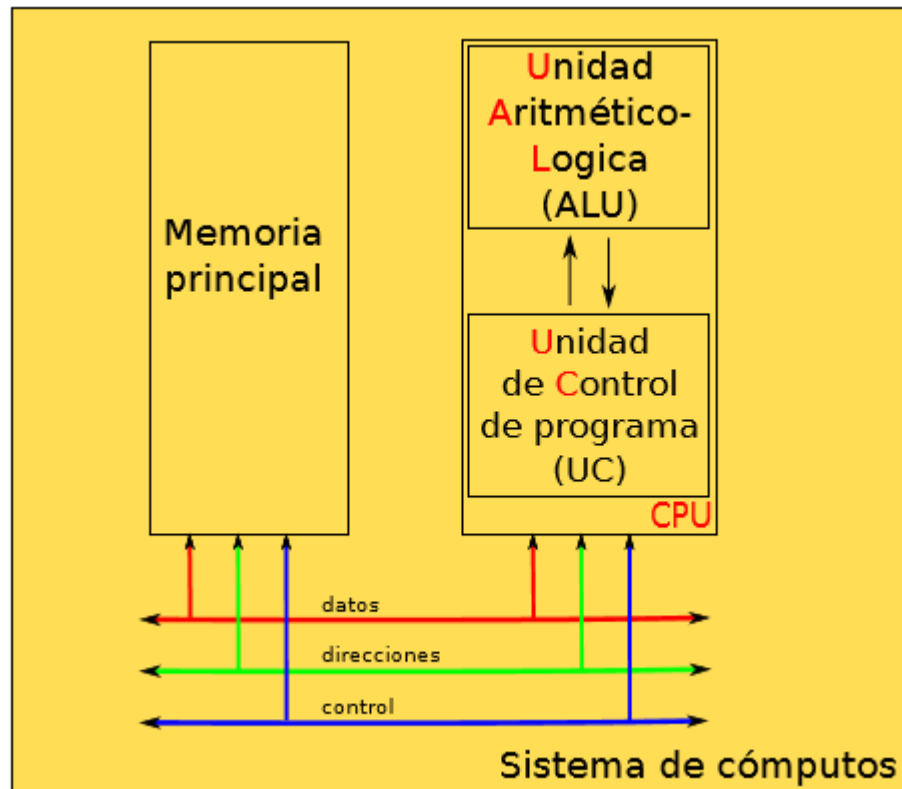
Características

- Cada componente de almacenamiento puede caracterizarse según las dimensiones:
 - Ubicación
 - Unidad de transferencia
 - Volatilidad
 - Capacidad de escritura
 - Método de acceso

Memorias

Características - Ubicación

- Interna o Externa al sistema de cómputos



Memorias

Características - Volatilidad

- Una memoria es volátil cuando necesita alimentación eléctrica para mantener la información.
- Una memoria no volátil almacena información sin necesidad de tener alimentación y esa información no se deteriora.

Memorias

Características - Capacidad de escritura

- Las memorias de solo lectura NO pueden ser alteradas luego de ser escritas. Necesariamente deben ser también NO volátiles.

Memorias solo lectura

Características - Capacidad de escritura

- ROM
 - Programas estáticos
 - Útiles también para dispositivos como microondas, calculadoras, etc

Memorias solo lectura

Características - Capacidad de escritura

- PROM
 - P es por *Programmable*
 - ROMs que pueden ser grabadas por el usuario
 - Se graban utilizando pulsos de altos voltajes

Memorias solo lectura

Características - Capacidad de escritura

- EPROM
 - E de *Erasable*
 - PROMs borrables
 - Pueden ser borradas con luz ultravioleta
 - EEPROM pueden ser borradas electrónicamente

Memorias solo lectura

Características – Métodos de acceso

- Secuencial
 - La dirección de cada dato esta almacenada junto con él. Debe recorrerse secuencialmente para buscar el dato.
 - Tiempo de acceso variable



Memorias solo lectura

Características – Métodos de acceso

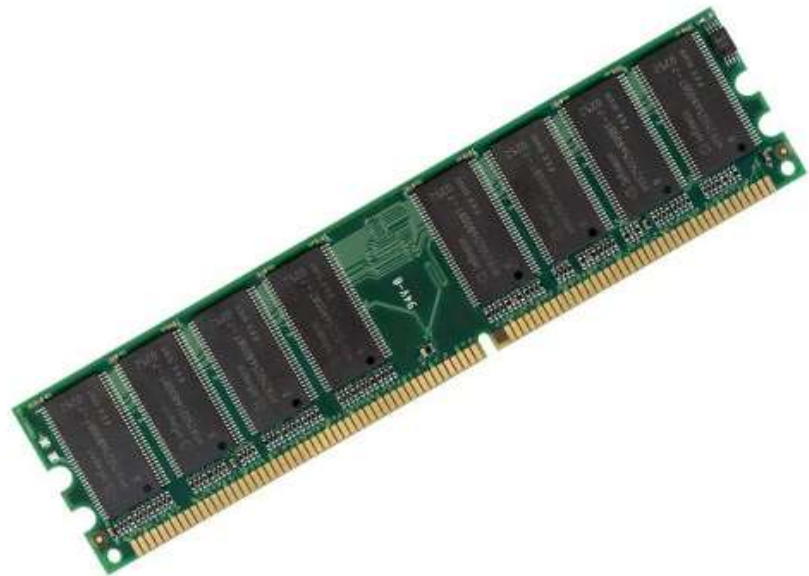
- Directo
 - La dirección de cada dato se basa en su ubicación física.
 - Etapas del acceso:
 - acceder directamente a la zona próxima del registro a buscar
 - buscar secuencialmente dentro de esa zona.



Memorias solo lectura

Características – Métodos de acceso

- Aleatorio
 - Cada dato tiene un mecanismo de acceso único y cableado físicamente tiempo de acceso constante e independiente de los accesos anteriores.



Memorias solo lectura

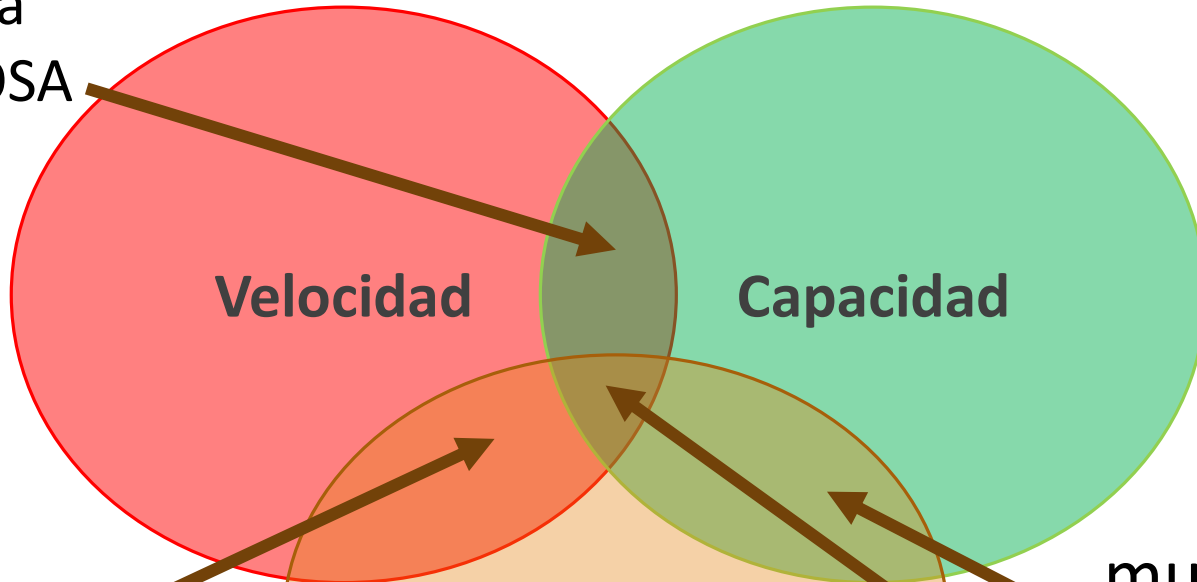
Características – Métodos de acceso

- Asociativo
 - Cada celda es recuperada en función de su contenido en lugar de su dirección:
 - Se busca un determinado patrón (clave) en cierto conjunto de bits de cada celda.
 - Las comparaciones son simultaneas
 - El tiempo de acceso es constante.



Jerarquía de Memoria

Mucha memoria
y muy rápida
MUY COSTOSA



Velocidad

Capacidad

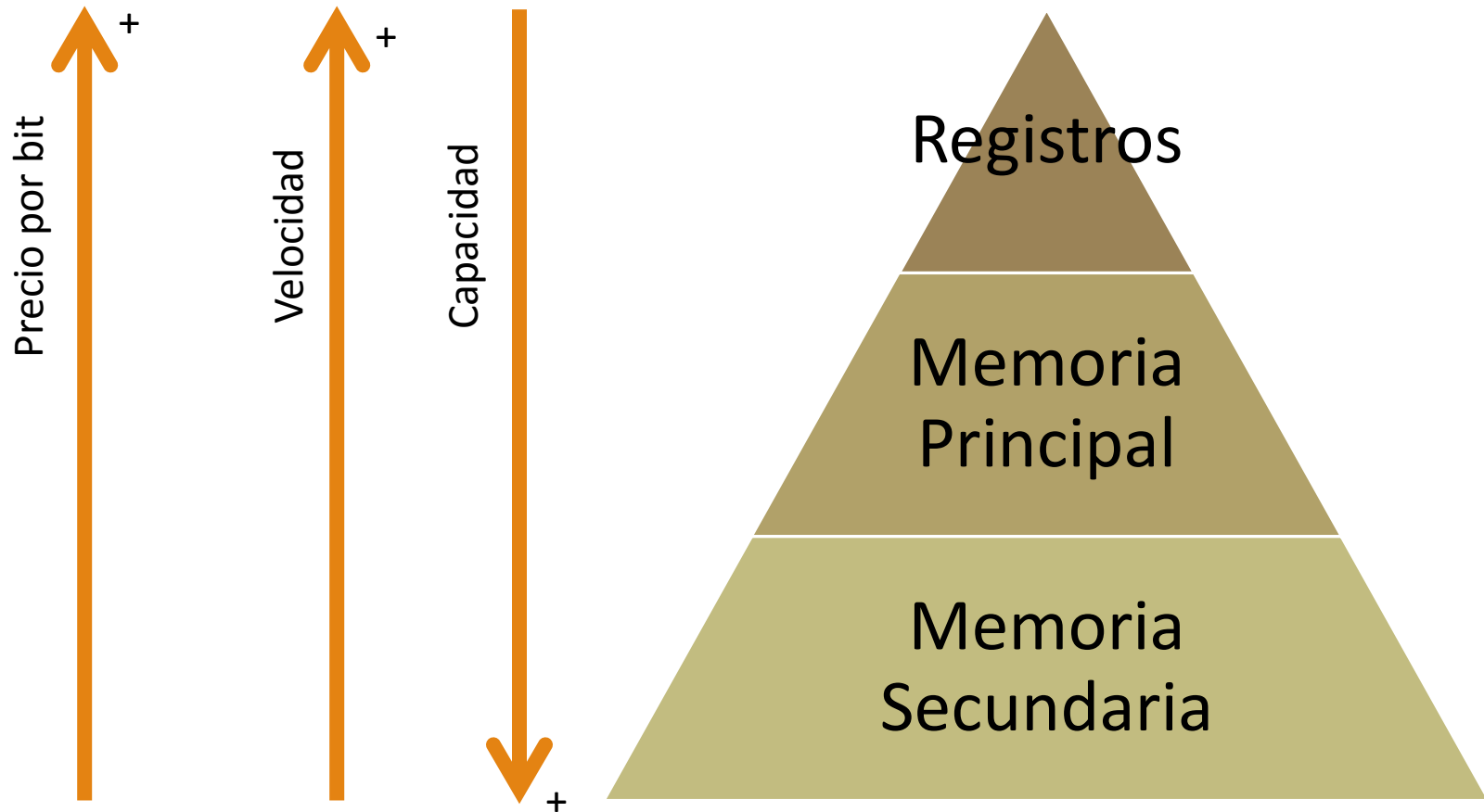
Precio

mucha y barata
LENTA

Rápida y barata
POCA CAPACIDAD

Imposible!!

Jerarquía de Memoria



Performance

Relaciones

- En las últimas décadas, cada año y medio:
 - la velocidad de proceso (en cantidad de instrucciones ejecutadas por segundo) se duplica
 - el tamaño de la memoria se duplica
 - la velocidad de la memoria crece menos del 10%

Performance

Relaciones

- La CPU necesita sí o sí ir a memoria para obtener la próxima instrucción
- La velocidad de ejecución de instrucciones está limitada por el tiempo de acceso a memoria que es mucho mayor al de la CPU
- ¿Como mejorar la performance del procesador?

Performance

Camino a la solución

- Se incorpora memoria interna a la CPU (Registros)
- Se incorpora memoria externa al sistema (operada a través de un módulo de E/S)
- Empleo de memorias combinadas

Performance

Camino a la solución

- Los accesos en general no son al azar, sino que tienen cierto patrón:
 - El PC suele ejecutar la instrucción que está en la celda siguiente
 - Al recorrer un arreglo se recorren sus posiciones
 - El código de una subrutina se suele ejecutar mas de una vez

Performance

Principio de localidad temporal

- Cuando un programa hace referencia de una posición de memoria, se espera que vuelva a hacerlo en poco tiempo

0000	ADD R0, R1
0001	SUB R1, R3
0002	JMP -3

Performance

Principio de localidad espacial

- Las posiciones de memoria mas cercanas a alguna referenciada recientemente son mas probables de ser referenciada que las mas distantes.

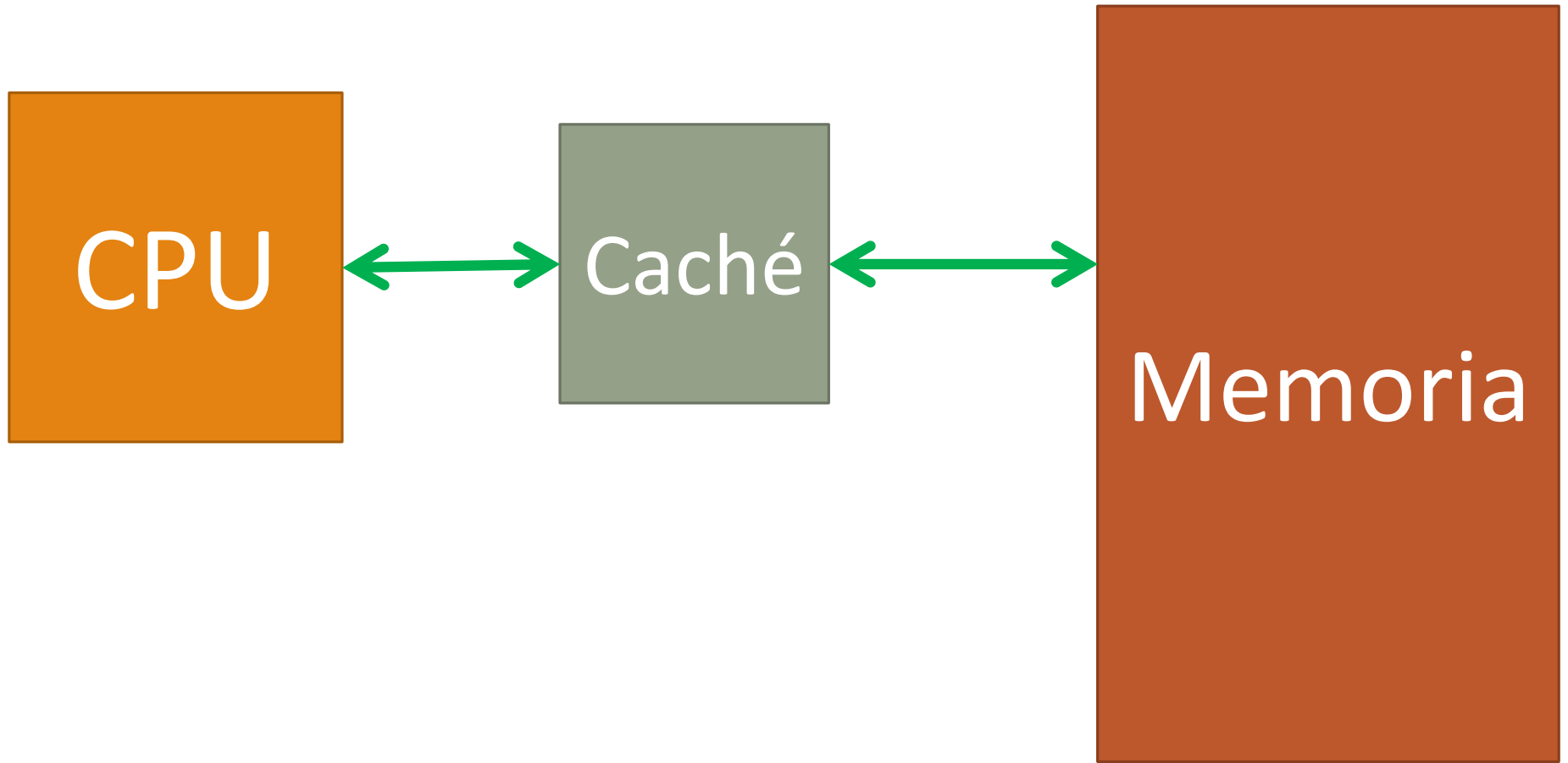
0000	ADD R0, R1
0001	SUB R1, R3
0002	MUL R0, R1
0003	...

Performance

Relaciones

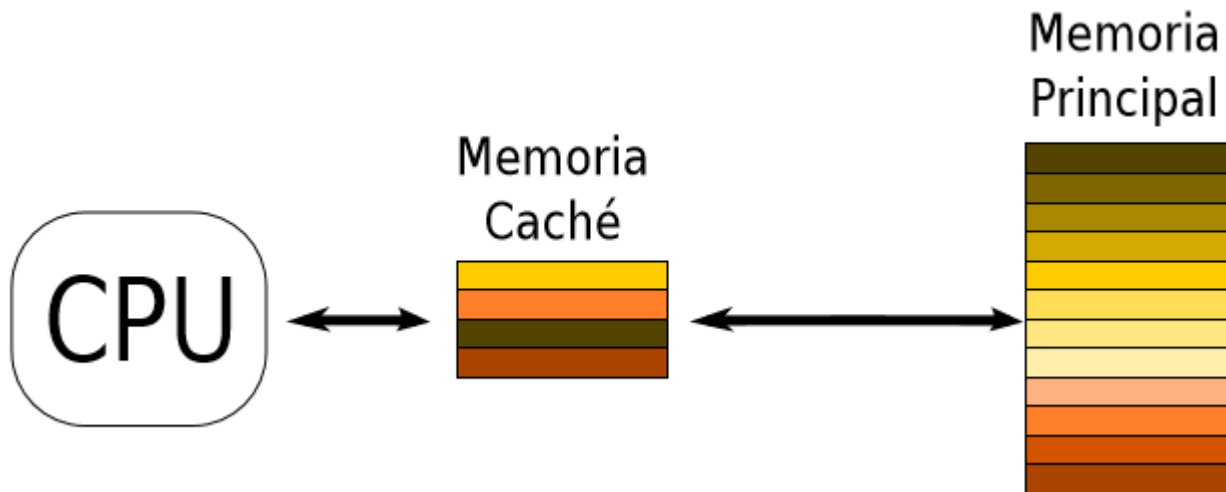
- En base a estos principios se ve que tiene sentido “tener a mano” a las posiciones recientemente usadas
- Se puede agregar una memoria rápida pero de poca capacidad para ir guardando estas posiciones

Memoria Cache



Memoria Cache

- La memoria caché tiene como objetivo proveer una velocidad de acceso cercana a la de los registros
- Para esto la caché contiene una copia de porciones de la memoria principal

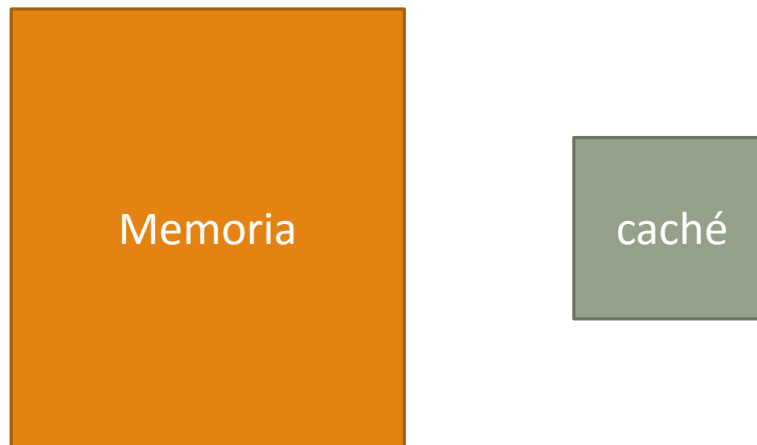


Memoria Cache

- ¿Cómo funciona?
 - La CPU pide una posición de memoria
 - La caché busca esta posición
 - Si está: la devuelve y no se accede a memoria
 - Si no: Se pide la posición a la memoria, se guarda en el cache y se la devuelve al CPU
- ¿Cómo saber si algo esta en la caché o no?

Memoria Cache

- ¿Puede estar todo el contenido de la memoria en caché?



- Hay que decidir cómo y dónde se guarda lo que entra en la caché

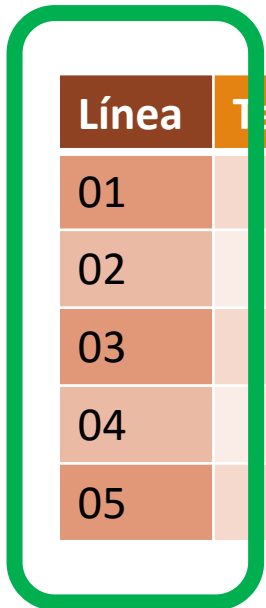
Memoria Cache

Organización

Línea	Tag	Contenido 1	Contenido 2	Contenido 3	Contenido 4
01					
02					
03					
04					
05					

Memoria Cache

Organización



Línea	Tag	Contenido 1	Contenido 2	Contenido 3	Contenido 4
01					
02					
03					
04					
05					

En vez de celdas la memoria caché tiene líneas
(esto no se guarda en la caché)

Memoria Cache

Organización

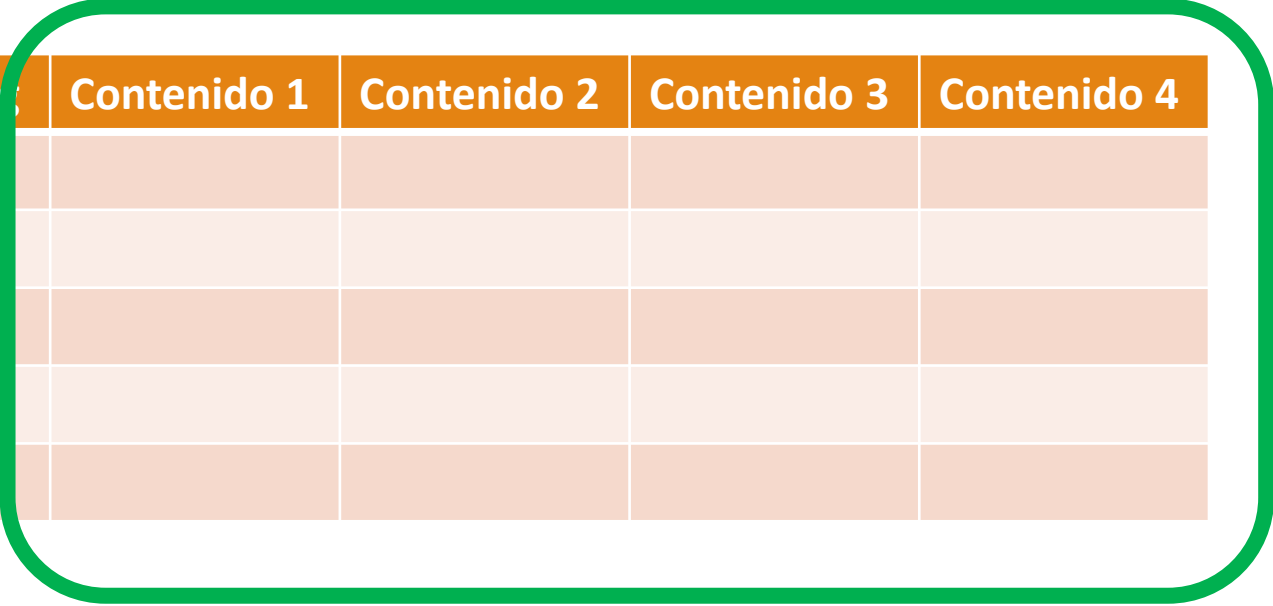


Línea	Tag	Contenido 1	Contenido 2	Contenido 3	Contenido 4
01					
02					
03					
04					
05					

Indica que posición de memoria está guardada en cada línea (Memoria asociativa)

Memoria Cache

Organización



Línea	Tag	Contenido 1	Contenido 2	Contenido 3	Contenido 4
01					
02					
03					
04					
05					

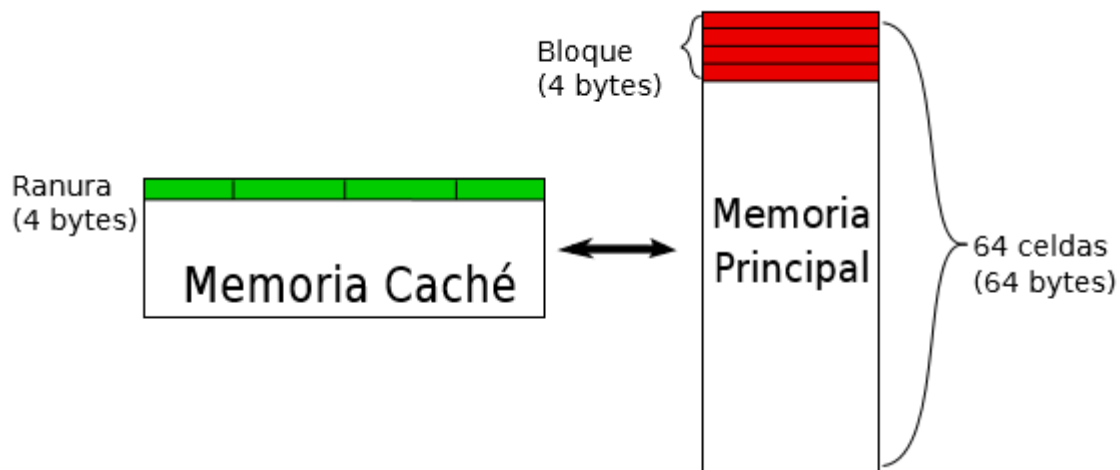
Contenido de la celda de memoria. A veces se guardan varias celdas consecutivas (bloque) por línea.

En ese caso se cargan todas juntas al cargarse alguna.

Memoria Cache

Funcionamiento

- ¿Cómo decidir en que línea guardar una celda de memoria?
- Función de correspondencia
 - (mapeo) asocia los bloques de memoria con ranuras de la cache.



Memoria Cache

Mapeo Directo

- La dirección se divide en 3 partes:
 - **Tag:** Indica el bloque de memoria que está guardada (hay que almacenarla en la caché)
 - **Línea:** Se usa para decidir en que línea se guarda el bloque (Esta parte no se guarda)
 - **Índice:** Indica dentro de la línea donde está una celda en particular

Memoria Cache

Mapeo Directo

Tag: 5F
 Línea: A
 Índice: 1

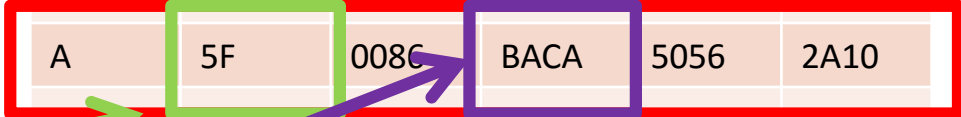
5FA1 → Mapeo Directo



Línea A

En 1 esta Hay que buscar esa celda

Línea	Tag	Cont 0	Cont 1	Cont 2	Cont 3
0	FE	0000	2234	2312	1234
1	DE	345F	AAA2	BE32	212A
2	5F	EA34	123F	DDDD	CCCC
3	6E	A689	0985	4359	1237
4	BC	1223	1123	2132	5753
5	D0	2334	2553	2132	213E
6	56	DDB5	DB65	A235	EEEE
7	23	4569	2386	1234	0532
8	2E	2543	4128	8654	003A
9	12	1250	2354	4467	0990
A	5F	008C	BACA	5056	2A10
B	05	5221	7652	4000	FE25
C	00	4652	1254	01A7	4567
D	AA	7904	0054	4322	8754
E	AA	56A3	0000	6789	4329
F	23	FC56	34AA	8754	8875



(Suponiendo 1 byte tag, 4 bits línea y 4 bits índice)

Memoria Cache

Mapeo Directo

Número de línea

Número de piso

Tag

¿Quién es el inquilino?

Índice

Número de depto



Memoria Cache

Mapeo Directo

- Ejemplo:
 - AFF1 con 8 bits de tag, 4 de línea y 4 de índice
 - Tag = AF, línea = F, índice = 1
 - A cada bloque le toca una única línea
 - Hay varios bloques que “compiten” por la misma línea
 - Si hay que guardar un bloque y la línea esta ocupada, se saca a su ocupante actual

Memoria Cache

Mapeo Directo

- Ejercicio:
- Cuantos bits de una dirección se destinan a: tag, linea y palabra en el siguiente esquema:
 - Una memoria principal de 32 celdas de un byte
 - Una memoria cache con:
 - Bloques de 4 celdas
 - 4 lineas
 - correspondencia directa

Memoria Cache

Mapeo Directo

- Ventajas:
 - Sencillo de implementar
- Desventajas:
 - Poco flexible: Si varias celdas con el mismo número de línea pero distinto tag se acceden todo el tiempo la caché no funciona muy bien.
 - Ejemplo: 8 bits de tag, 4 de línea y 4 de índice, y tengo 2 arreglos en AA00 y BB00

Memoria Cache

Mapeo Asociativo

- Cada bloque puede ir a cualquier línea de caché
- Parte la dirección en tag e índice
- Ejemplo: F145 en una caché asociativa con 16 celdas por bloque:
 - Índice = 5
 - Tag = F14
- ¿Qué pasa cuando la caché se llena? Algoritmos de reemplazo, mas adelante

Memoria Cache

Mapeo Asociativo

- Ejercicio
 - Memoria principal de 16 celdas de 1 byte cada una
 - Memoria cache de 4 líneas
 - Bloques de 1 celda
 - Función de correspondencia asociativa
- ¿Que tamaño tiene el tag?
 - No hay patrón de asignación, cualquier bloque en cualquier línea 16 celdas = 16 bloques
 - Se debe distinguir cual de esos 16 bloques es el que esta almacenado en cada línea
 - Tag=4bits

Memoria Cache

Mapeo Asociativo

- Ejercicios
- Calcular como se divide una dirección de memoria de 16 bits si se tiene una cache asociativa de 4 celdas por bloque y 256 líneas
- ¿Cómo se decide si la dirección FA32 está en caché?

Memoria Cache

Mapeo Asociativo

- Ventajas:
 - Muy flexible. Solo se saca algo de la caché cuando no hay mas lugar
- Desventajas:
 - Muy compleja de implementar, hay que revisar toda la caché para saber si algo está o no.

Memoria Cache

Mapeo Asociativo por conjuntos

- Combina la flexibilidad del mapeo asociativo y la simplicidad del directo.
- Divide la caché en conjuntos de varias líneas.
 - Cada celda puede ir a un único conjunto
 - Dentro del conjunto puede ir a cualquier línea
 - Si se llena el conjunto hay que elegir a alguna para desalojar
- La dirección se parte en tag, conjunto e índice

Memoria Cache

Mapeo Asociativo por conjuntos

Línea	Tag	Cont 0	Cont 1	Cont 2	Cont 3
0					
1					
2					
3					
4					
5					
6					
7					

Memoria Cache

Mapeo Asociativo por conjuntos

Conjuntos de
2 vías

Línea	Tag	Cont 0	Cont 1	Cont 2	Cont 3
0					
1					
2					
3					
4					
5					
6					
7					

Memoria Cache

Mapeo Asociativo por conjuntos

Conjuntos de
4 vías

Línea	Tag	Cont 0	Cont 1	Cont 2	Cont 3
0					
1					
2					
3					
4					
5					
6					
7					

Memoria Cache

Mapeo Asociativo por conjuntos

- La dirección se divide en:
- tag(2b) conjunto (2b) palabra (2b)

- Para una dirección de memoria dada:
 1. Se identifica el conjunto
 2. Se comparan los primeros 2 bits con los tags de las ranuras del conjunto
 3. Si hay coincidencia se recupera la palabra dentro del bloque según los últimos 2 bits

Memoria Cache

Mapeo Asociativo por conjuntos

- Calcular como se divide una dirección de memoria de 16 bits si se tiene una cache con asociativa por conjuntos de 4 vías, de 4 celdas por bloque y 256 líneas.
- Tag: Conjunto: Índice:
- ¿Cómo se decide si la dirección FA32 está en caché?

¿Que pasó hoy?!

- Memorias:
 - Características
 - Memorias ROM
 - Jerarquía de memorias
- Caché:
 - Motivación
 - ¿Qué?
 - Organización
 - Función de mapeo