Arquitectura i Configuracions Informàtiques

Tema 3. La memoria

Davide Careglio

Introducción

- Tema 1. Introducción
- Tema 2. El microprocesador
- Tema 3. Memoria
- Tema 4. Dispositivos de E/S y buses
- Tema 5. DataCenters y modelos de comunicación

Tema 3. La memoria

- Tema 1. Introducción
- Tema 2. El microprocesador
- Tema 3. Memoria
  - Tipos de memoria
  - Métricas
  - Breve historia y evolución hacia la jerarquía de memorias
  - DRAM y SDRAM
  - SRAM
  - Jerarquía de memorias y funcionamiento
  - Rendimiento
  - Memoria virtual
- Tema 4. Dispositivos de E/S y buses
- Tema 5. DataCenters y modelos de comunicación

Tema 3. Tipos

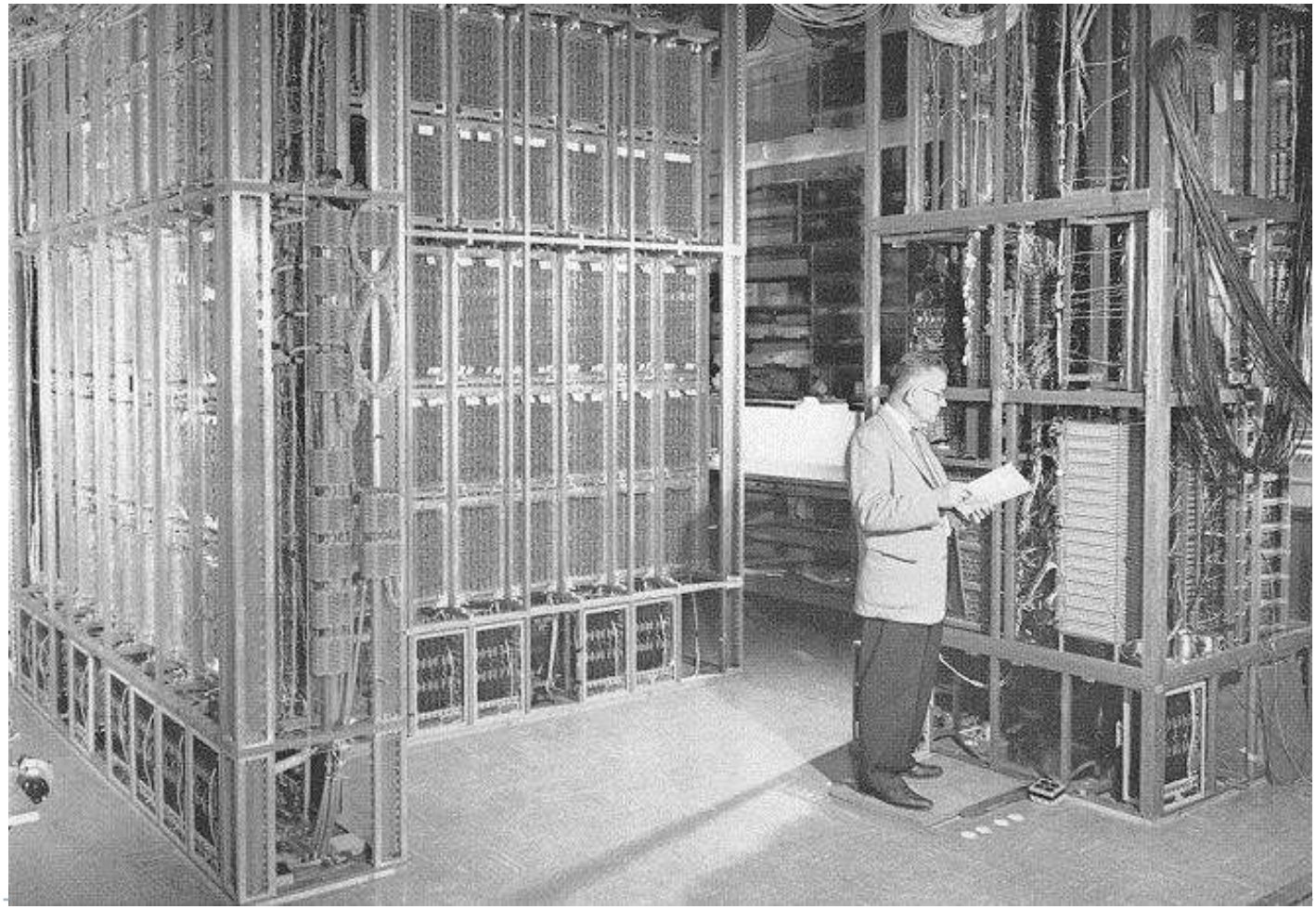
- Random Access Memory
  - Memoria que permite el acceso a cualquier zona (dirección) en cualquier momento
  - La información desaparece si se deja de alimentar (volátil)

Tema 3. Métricas

- Tiempo de acceso (ns)
  - Es el tiempo que transcurre desde el instante en el que se presenta una dirección a la memoria hasta que el dato, o ha sido memorizado (escrito) o está disponible para su uso (leído)
- Velocidad de reloj (GHz)
  - Tiempo de ciclo (inversa): en memorias RAM es igual al tiempo mínimo entre dos accesos consecutivos.
  - Hz efectivos o equivalentes. En algunos casos, la memoria transmite dos veces por cada ciclo de reloj (memorias DDR: Double Data Rate).
- Ancho de banda (GB/s) o velocidad de transferencia.
  - Cantidad de Bytes que pueden transferirse hacia o desde la memoria por segundo.
  - Velocidad reloj x ancho del bus de datos en Bytes

Tema 3. Breve historia

- 50’ Memorias de ferrita
  - Gran espacio físico, liberaban gran cantidad de calor

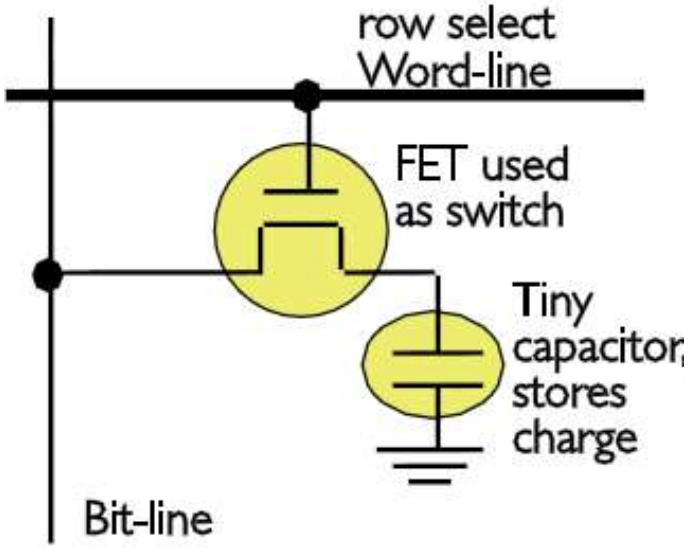


Tema 3. Breve historia

- 60’ Revolución de los circuitos integrados
  - Punto de memoria: pequeño condensador
    - Reducción tasa errores
    - Reducción tiempo acceso
    - Reducción espacio físico ocupado
    - Reducción consumo de potencia
    - Reducción coste económico
  - RAM: Random Access Memory
  - DRAM: Dynamic RAM

Tema 3. Breve historia
DRAM

- Celdas que consisten de diminutos acumuladores con una carga que se consume rápidamente
- Es necesario que se refresque o recargue
  - Tiempo de refresco
  - Obliga al procesador a hacer pauses continuas
  - Se hace durante el tiempo de precarga
- Ventaja
  - Alta densidad de almacenamiento con dimensiones reducidas
  - Coste bajo
- Desventaja
  - Refresco continuado de la información
- Uso
  - Memoria



Tema 3. Breve historia

- 70’ Gran progreso del procesador
  - Empieza un progreso más lento de las memorias

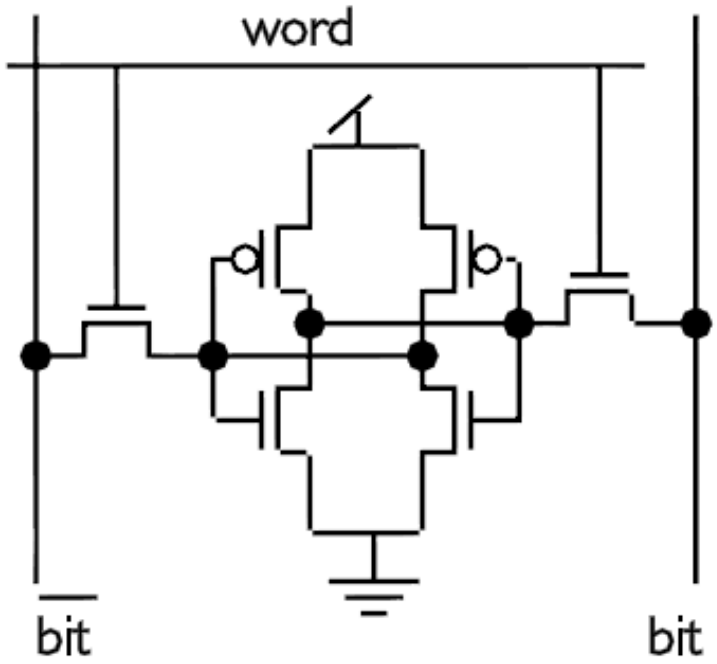
Tema 3. Breve historia

- 80' Aparecen las memorias caché
  - La elevada potencia del procesador requiere de una memoria de velocidad similar
  - Uso de transistores en lugar de condensadores
  - 6-8 transistores por punto
  - Memorias RAM de tipo Static RAM (SRAM)
  - 10 veces más rápidas

Tema 3. Breve historia

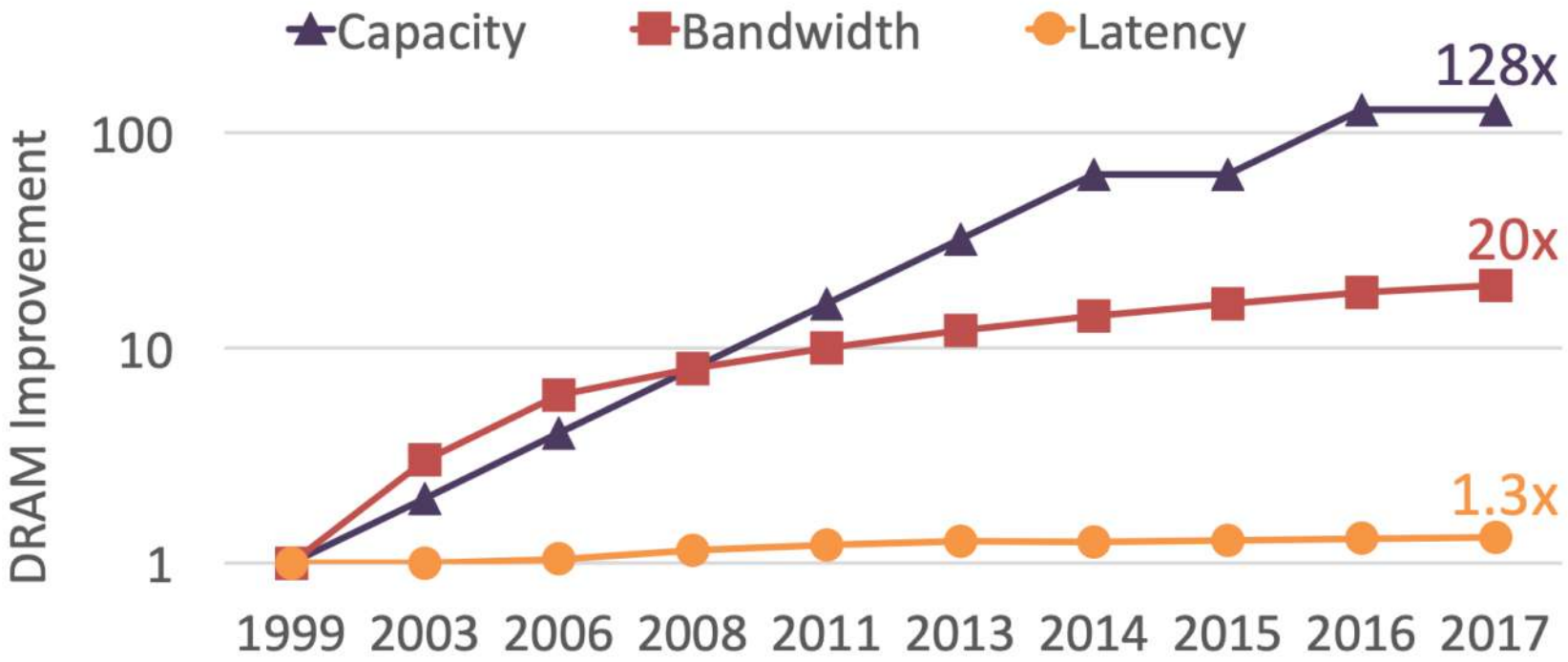
SRAM

- No necesita refresco continuado de su contenido
- La memoria retiene el contenido mientras se le aplica alimentación
- Para cada celda se necesitan 6 transistores
- Ventaja
  - Es más rápida que cualquiera DRAM
  - Los acumuladores de los puntos de memoria se sustituyen por transistores, no hay cargas y descargas de cada acumulador
- Desventaja
  - Tiene menos densidad de almacenamiento (es decir, ocupan más espacio)
  - Coste elevado
  - Sus transistores están siempre activos, gasto energético
- Uso
  - Memoria caché



Tema 3. Breve historia

- No todas las métricas de la RAM mejoran del mismo modo



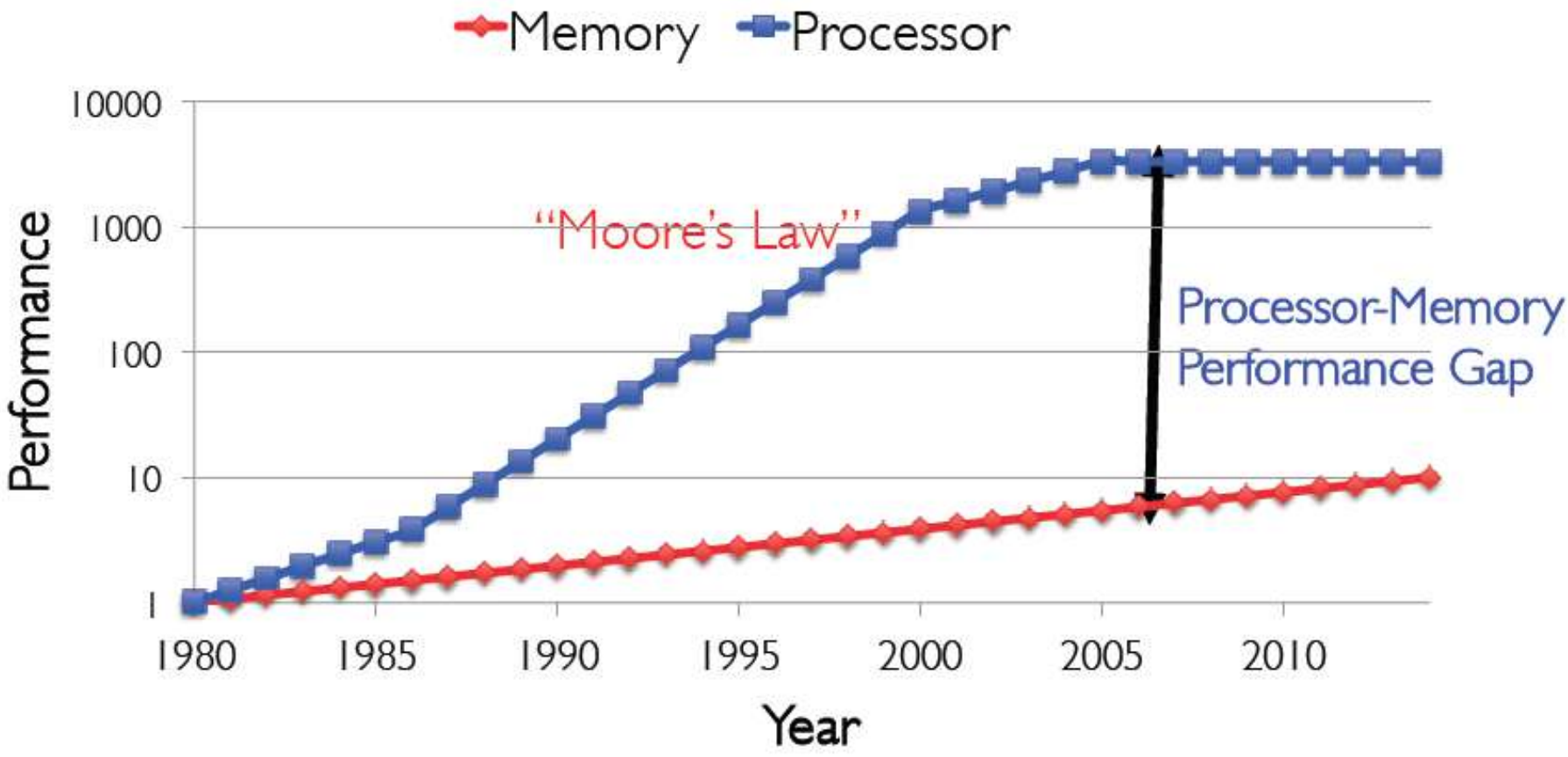
Year	Size	Cycle Time
1999	512 Mb	66 ns
2001	1 Gb	60 ns
2003	2 Gb	55 ns
2008	4 Gb	48 ns
2014	8 Gb	45 ns

16:1 (Size) < 1.5:1 (Cycle Time)

Fuente: K.K. Chang, "Understanding and Improving the Latency of DRAM-Based Memory Systems", arXiv:1712.08304v1, 2017.

Tema 3. Breve historia

- Pero respecto a la CPU, las memorias consiguen una mejora en el rendimiento inferior
- Procesador: 50%/año
- Memoria: 7%/año

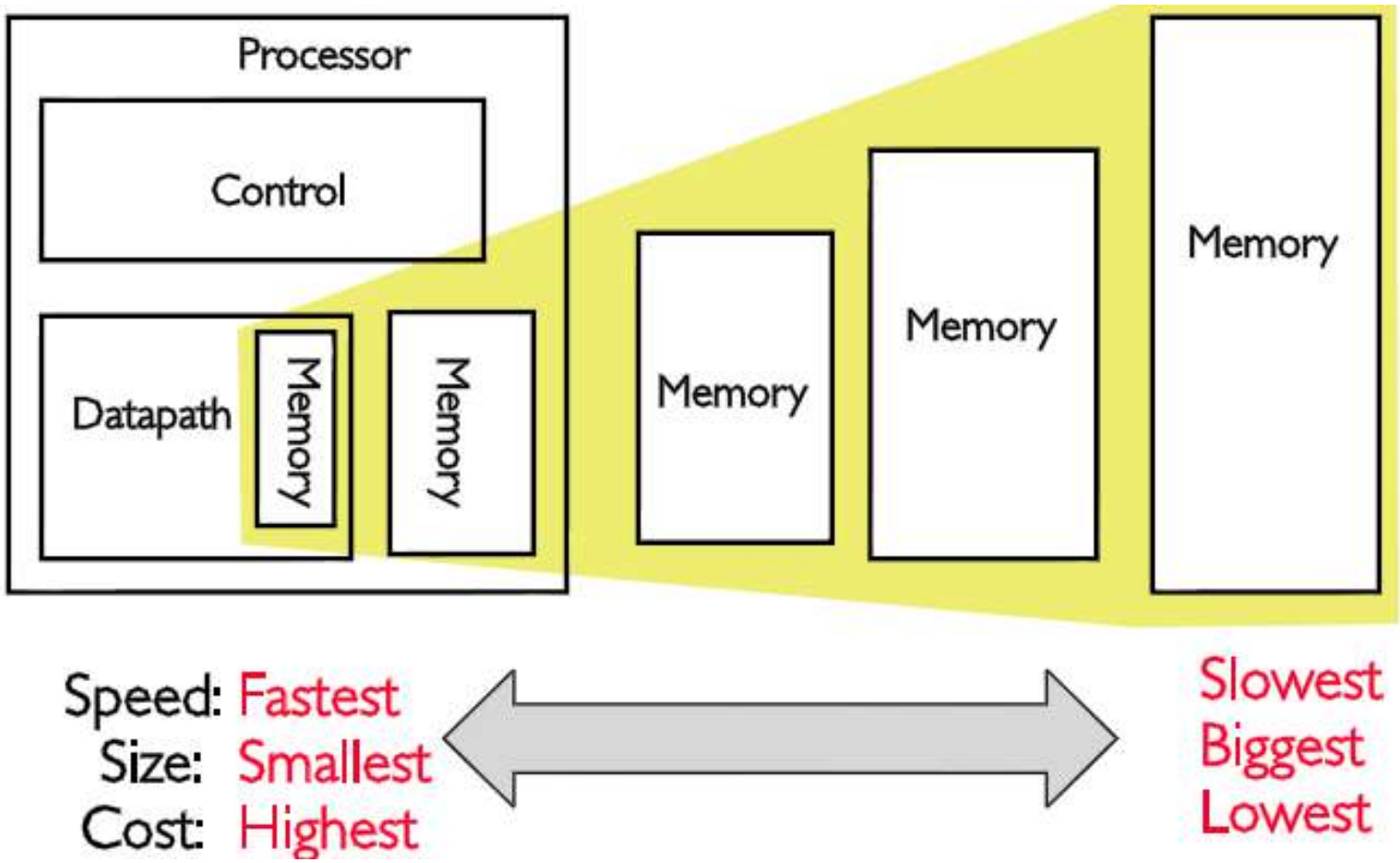


Tema 3. Breve historia

- 90’ Se define la jerarquía de memoria de un ordenador actual
  - Las memorias rápidas son caras y de poca capacidad
  - A mayor capacidad, más lentas y más baratas

Tema 3. Jerarquía de memorias

- Objetivo: crear la ilusión de tener una memoria grande y rápida

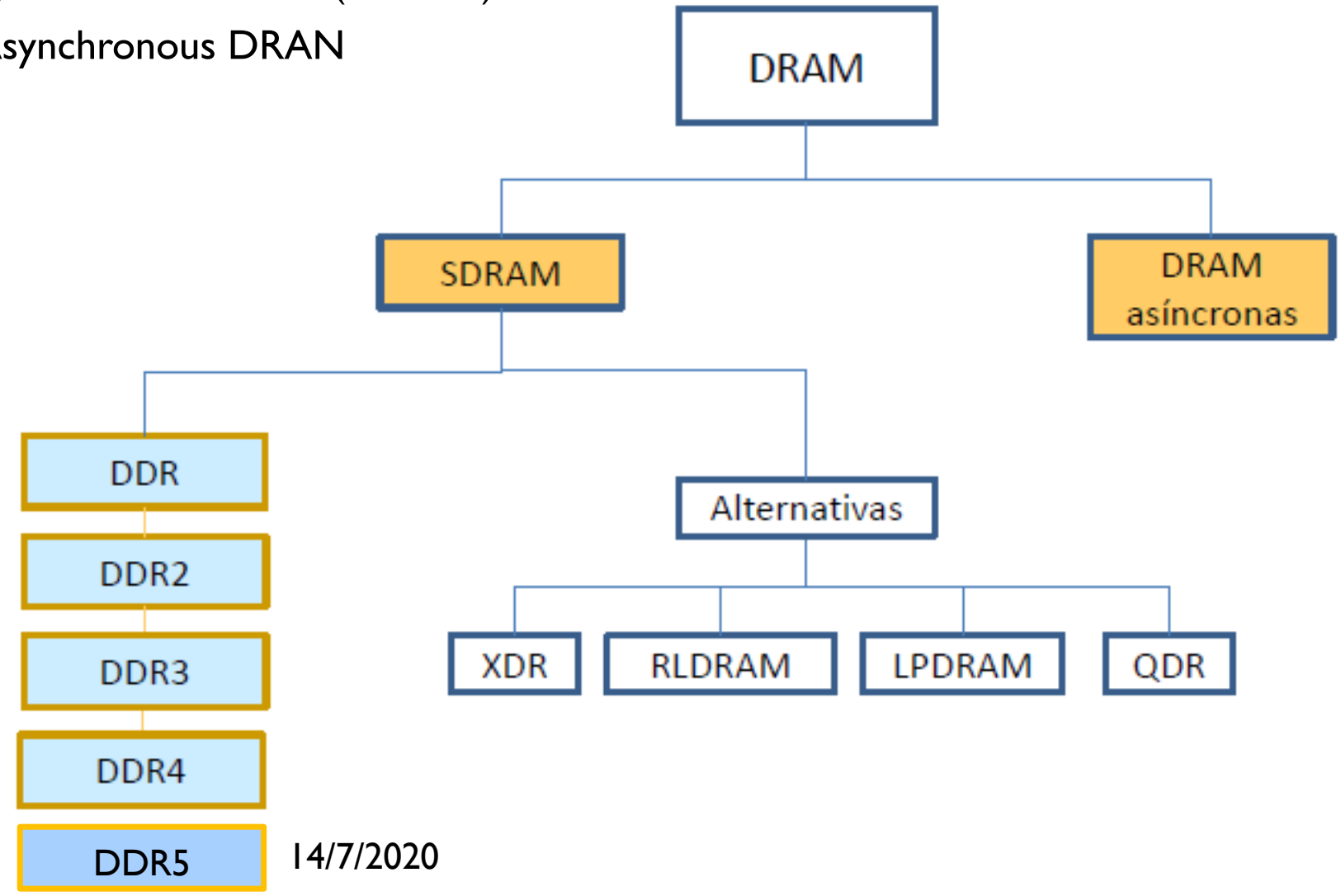


Tema 3. Tipos

- Static RAM (SRAM)
- Dynamic RAM (DRAM)

Tema 3. DRAM

- Synchronous DRAM (SDRAM)
- Asynchronous DRAN

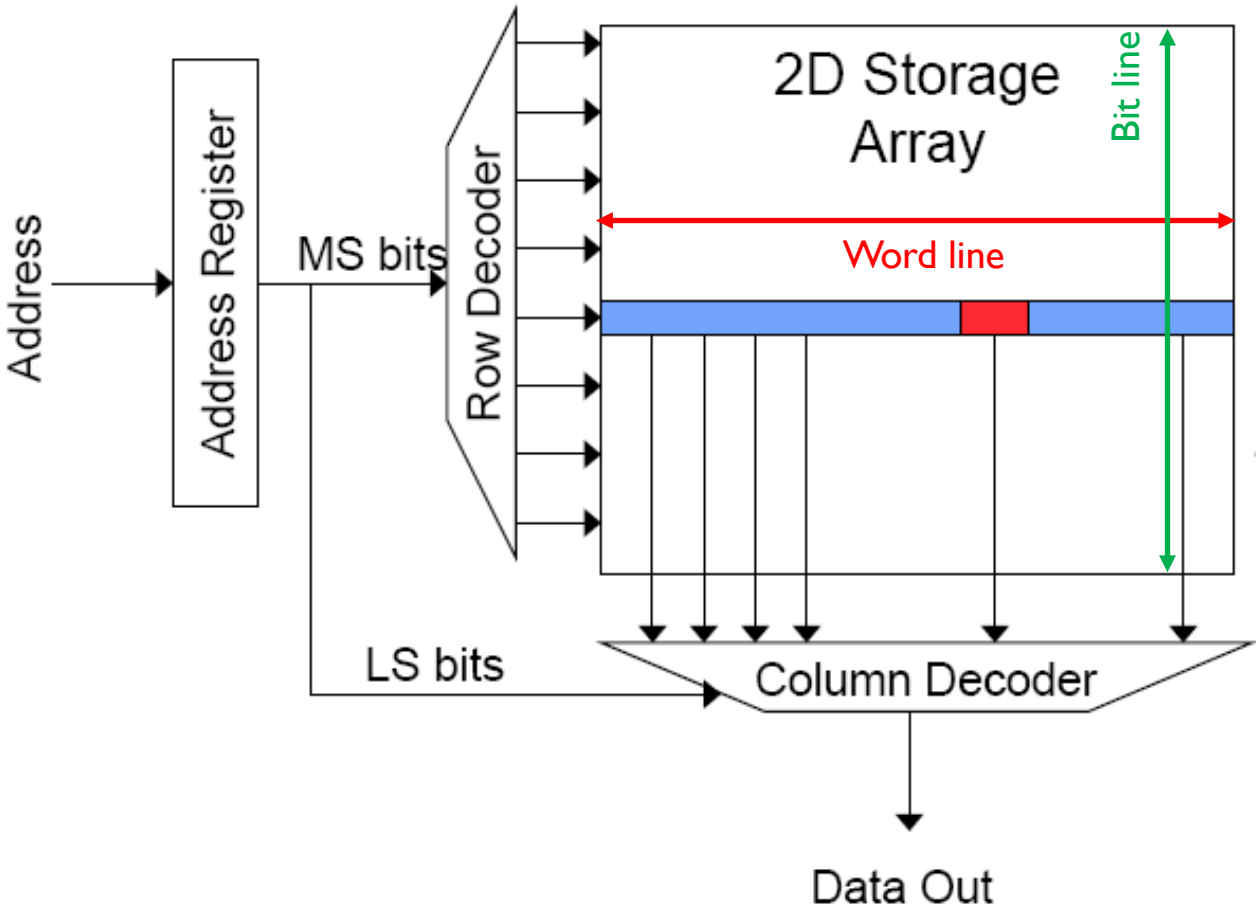


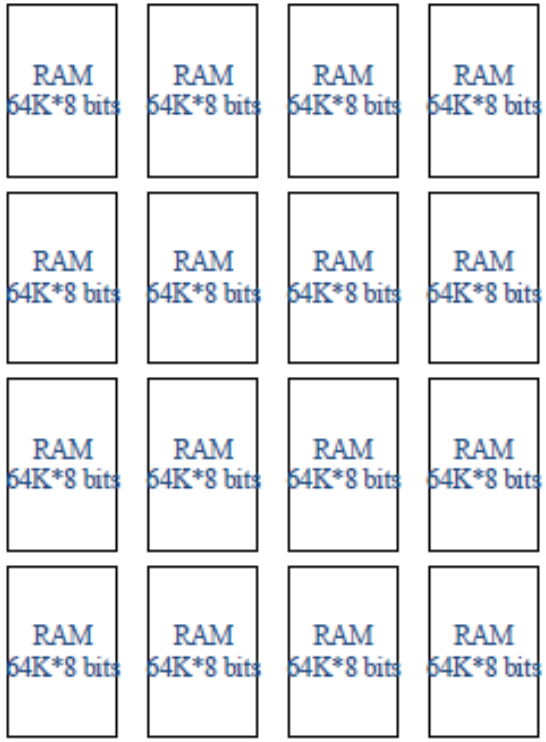
14/7/2020

Tema 3. DRAM

Estructura y funcionamiento

- Bloques entrelazados
- Estructurados en una matriz de 2 dimensiones

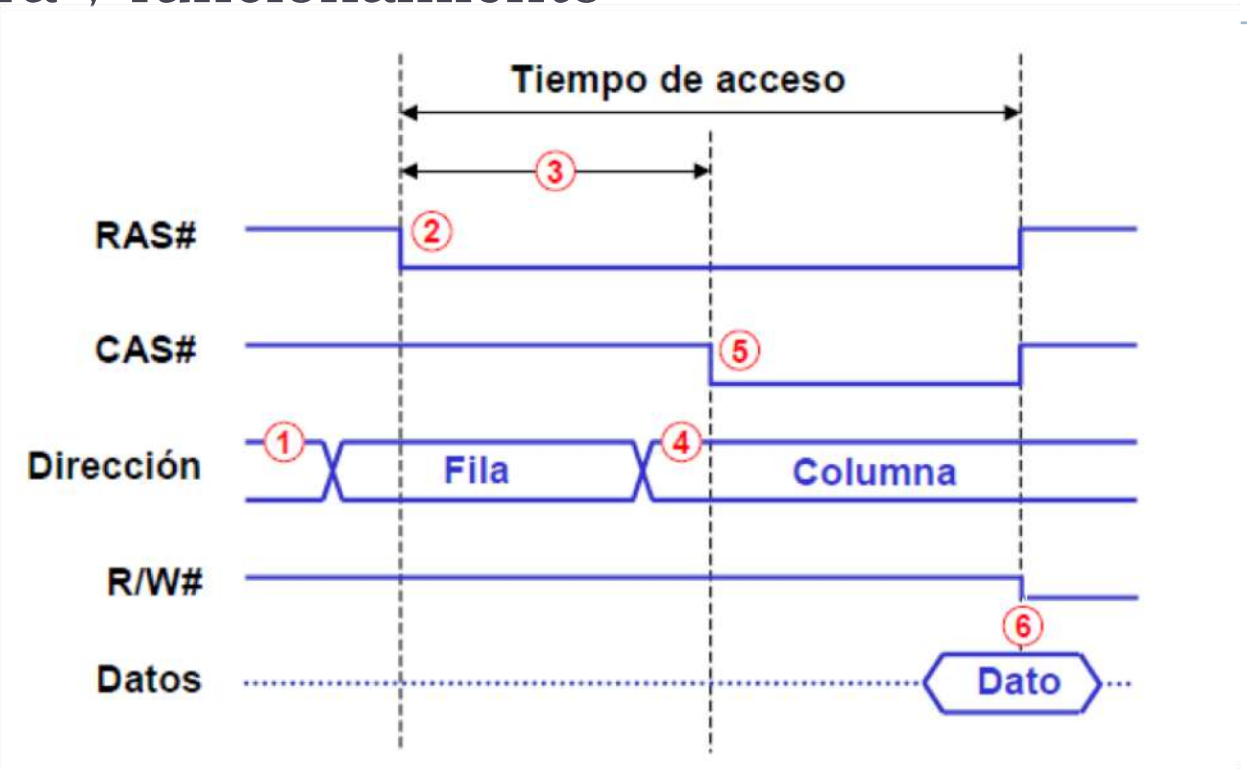




- Latencia t_{RCD}
  - Latencia entre el muestreo de dirección de fila RAS (Row Address Strobe) y el muestreo de dirección de columna CAS (Column Address Strobe)
  - Básicamente, el retardo entre el acceso a las direcciones de memoria de fila y de columna
- Latencia t_{CL}
  - Si se ha accedido a una fila, es el número de ciclos de reloj que hemos de esperar hasta que se obtiene el dato.
- $t_{RCD} + t_{CL}$ es la tiempo de acceso total para obtener el dato de la RAM
- Latencia t_{RP}
  - Precarga de fila (Row Precharge Delay)
  - Número de ciclos de reloj que tiene lugar desde que finaliza la latencia CAS del anterior acceso y el inicio al siguiente acceso (nueva dirección de fila)
- $t_{RCD} + t_{CL} + t_{RP}$ es la tiempo de ciclo de la RAM

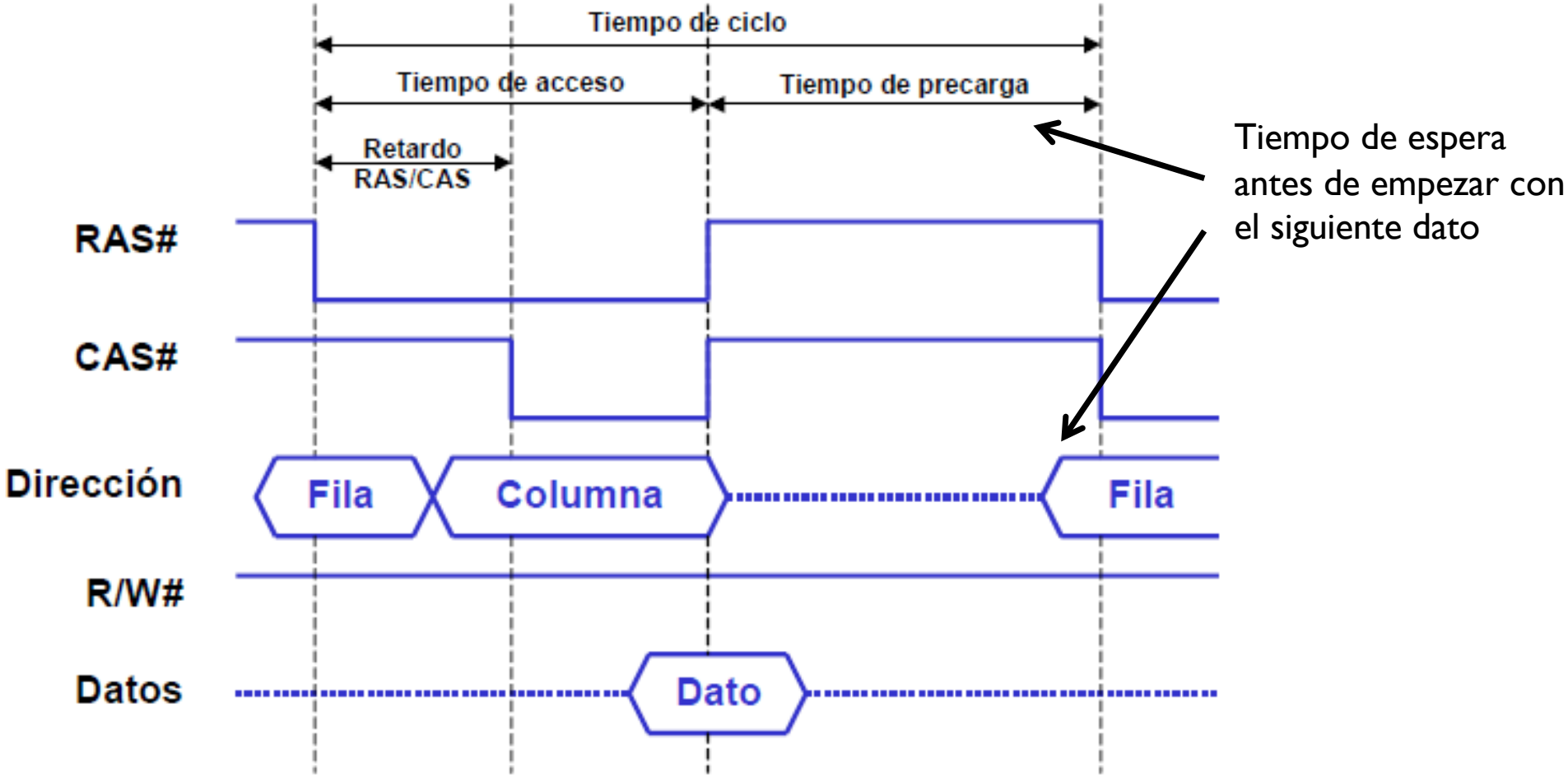
Tema 3. DRAM

Estructura y funcionamiento



1) Envío de la dirección de fila (bus de direcciones)
2) Activación de la señal RAS (bus de control)
3) Latencia t_{RCD}
4) Envío de la dirección de columna (bus de direcciones)
5) Activación de la señal CAS (bus de control)
6) Envío del dato (bus de dato) y activación señal de R/W (bus de control)

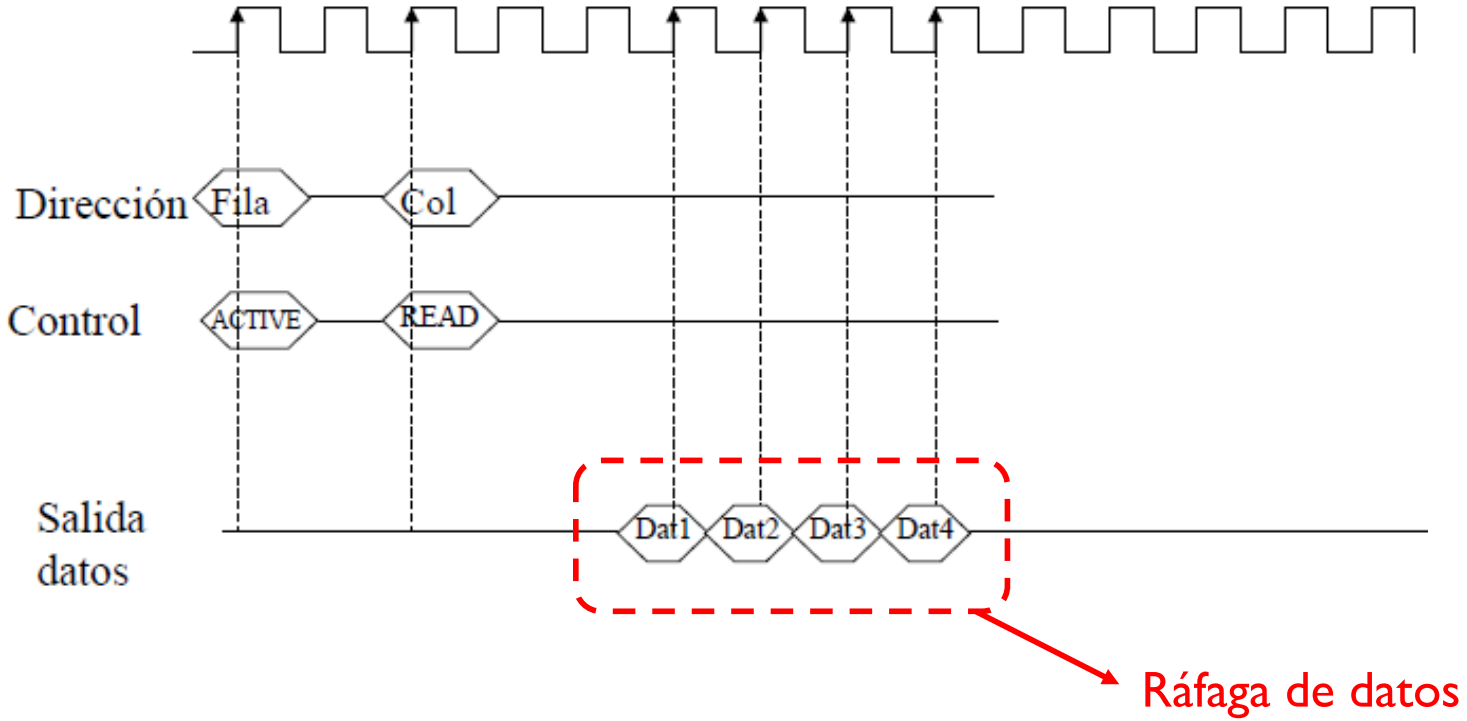
Tema 3. DRAM
Estructura y funcionamiento



Tema 3. SDRAM

- Synchronous DRAM, RAM Dinámica Síncrona
- Son capaces de trabajar de forma sincronizada con el reloj del sistema (el mismo de la CPU), sin tiempos de espera
  - RAS# y CAS# solamente se envían en sincronía con el reloj del sistema
- Al funcionar bajo el control del reloj del sistema, el procesador puede estar realizando otras tareas mientras la memoria realiza su operación
- Funciona en modo ráfagas para reducir el tiempo de acceso (no envía un único dato si no un conjunto a partir de una dirección)
- Aspecto
  - Tienen dos muescas de posicionamiento, una a 2.5cm del lateral izquierdo y el otro prácticamente en el centro
  - Su longitud es de133 mm
  - Tienen 168 contactos.

Tema 3. SDRAM


Dirección
Fila
Col
Control
ACTIVE
READ
Salida
datos
Dat1
Dat2
Dat3
Dat4
Ráfaga de datos

Tema 3. SDRAM
SDR SDRAM

- Single Data Rate SDRAM
- Velocidades de reloj de 66, 100 y 133 MHz (períodos de 15, 10, y el 7,5 ns)
- Módulos DIMM de 168 contactos
- Bus de memoria de 64 bits (ancho bus)
  - Transfiere 64 bits en cada ciclo de reloj
- Tipos
  - PC66: funciona a un máximo de 66 MHz y una capacidad de 533 MB/s
  - PC100: 100 MHz y 800 MB/s
  - PC133: 133 MHz y 1066 MB/s
- Usada en
  - Pentium II (1997), Pentium III (1999), AMD K6, (1997) AMD K7 (1999)

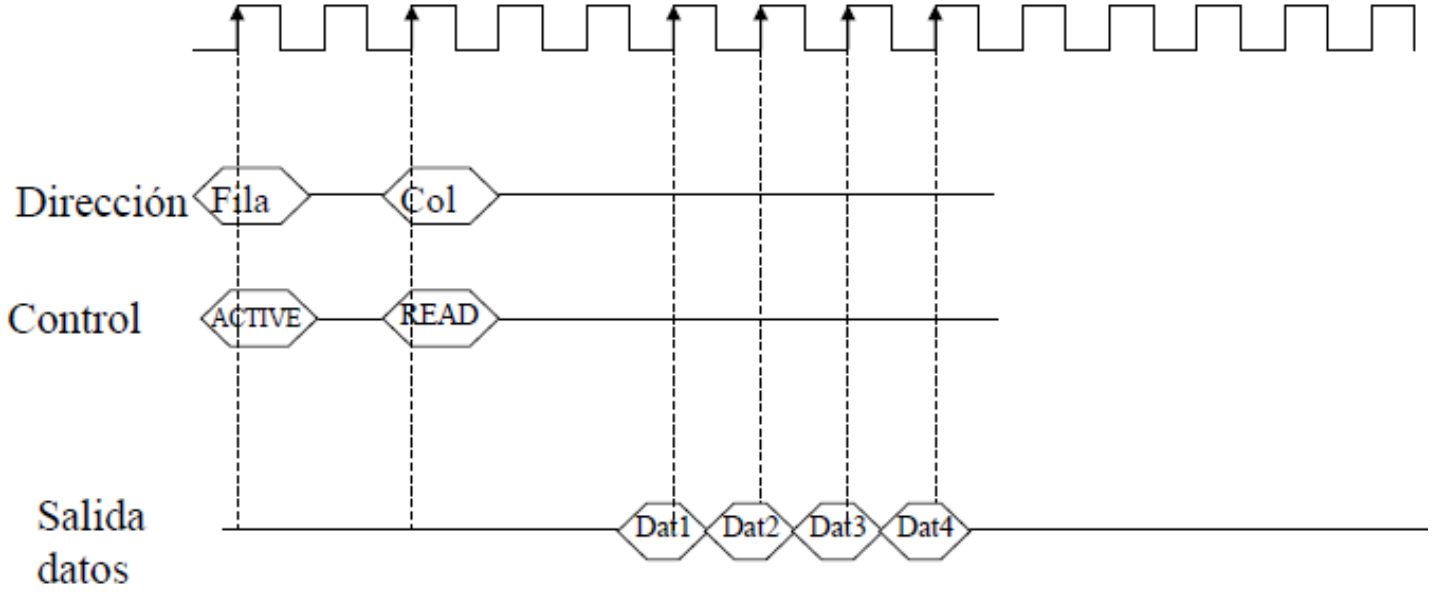
Tema 3. SDRAM
DDR SDRAM

- Double Data Rate SDRAM
- Velocidades de reloj de 100 a 200 MHz
- Módulos DIMM de 168 contactos
- Bus de memoria de 64 bits (ancho bus)
  - Transfiere 64 bits x 2 (dual rate) en cada ciclo de reloj
- Tipos
  - DDR-200: funciona a un máximo de 100 MHz y una capacidad de 1600 MB/s
  - DDR-266: 133 MHz y 2100 MB/s
  - DDR-333: 166 MHz y 2700 MB/s
  - DDR-400: 200 MHz y 3200 MB/s
- Double Data Rate porque aprovecha el doble de un tiempo de ciclo

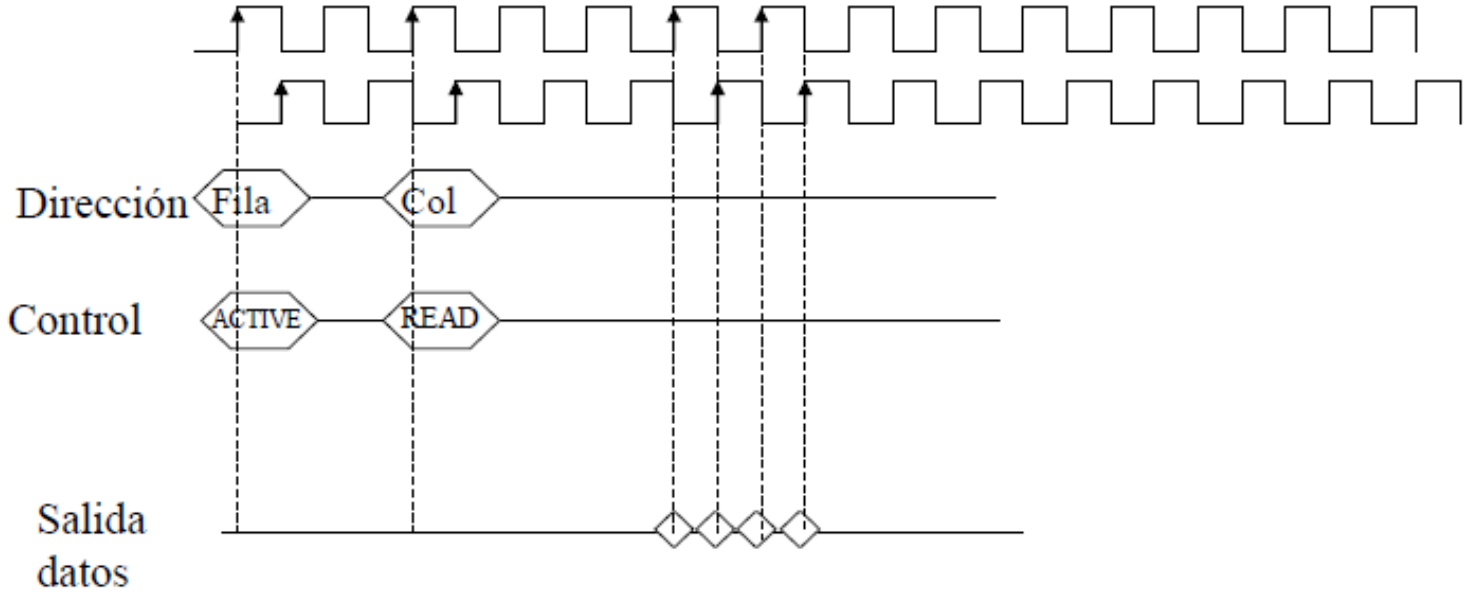
Tema 3. SDRAM

DDR vs SDR

SDR



DDR



Tema 3. SDRAM
DDR2 SDRAM

- Más velocidad, transferencia de datos superior, pero menor frecuencia interna, por lo tanto, menos potencia y disipa menos calor
- No compatible con DDR
- Menor voltaje que DDR (2,5V vs. 1,8 V).
- DDR2 tienen 240 contactos.
- DDR2 proporciona 4 transferencias de datos por ciclo, lo que hace que su velocidad de bus de memoria efectiva sea el resultado de multiplicar su velocidad real (física) por 4
  - Suponiendo una velocidad base del reloj de 100Mhz, la memoria DDR proporciona 1600 MB/s de ancho de banda, DDR2 proporciona 3200 MB/s

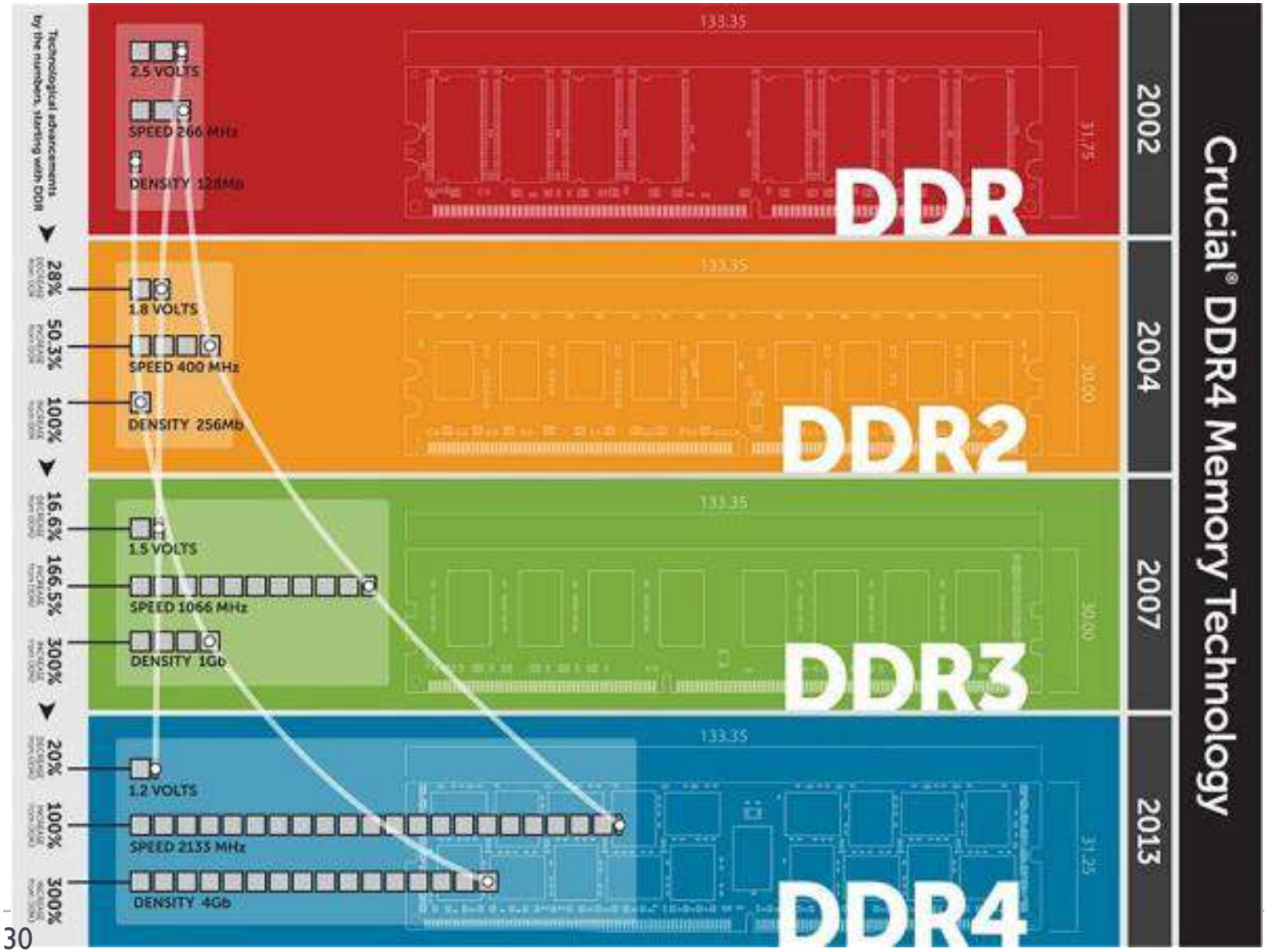
Tema 3. SDRAM
DDR3 SDRAM

- Menor voltaje que la DDR2 (1,5V, en lugar de 1,8V) pero mayores latencias que DDR2
- DDR3 tienen 240 contactos pero el formato es distinto de DDR2 (muesca)
- DDR3 proporciona 8 transferencias de datos por ciclo
  - Suponiendo una velocidad base del reloj de 100Mhz, la memoria DDR proporciona 1600 MB/s de ancho de banda, DDR2 proporciona 3200 MB/s, y DDR3 6400 MB/s
- Entre 6,4 GB/s y 12,8 GB/s

Tema 3. SDRAM
DDR4 SDRAM

- Gran capacidad, alta densidad del chip, bajo voltaje
- Frecuencias desde los 2133 hasta los 4266 MHz
- Capacidad de 17066 MB/s
- Diseñadas para soportar procesadores multinúcleo de alta velocidad
- Modulo de 288 contactos

Tema 3. SDRAM
DDR4 SDRAM



Tema 3. SDRAM
En tarjetas graficas

- GDDR2
  - Es un tipo de memoria adaptada para tarjetas de video, con características de la memoria DDR y DDR2.
- GDDR3
  - Es un tipo de memoria adaptada para el uso con tarjetas de video, con características de la memoria DDR2, mejoradas para reducir consumo eléctrico y hacer eficiente la disipación de calor.
- GDDR4
  - Es un tipo de memoria que también se basa en la tecnologíaDDR2, que mejora las características de consumo y ventilación con respecto a la GDDR3.
- GDDR5
  - Basada en tecnología DDR3, provee un doble ancho de banda a diferencia de GDDR4, que permite ser configurada a 32 y 64 bits.
- GDDR6 (2018)

Tema 3. Tipos

- Static RAM (SRAM)
- Dynamic RAM (DRAM)

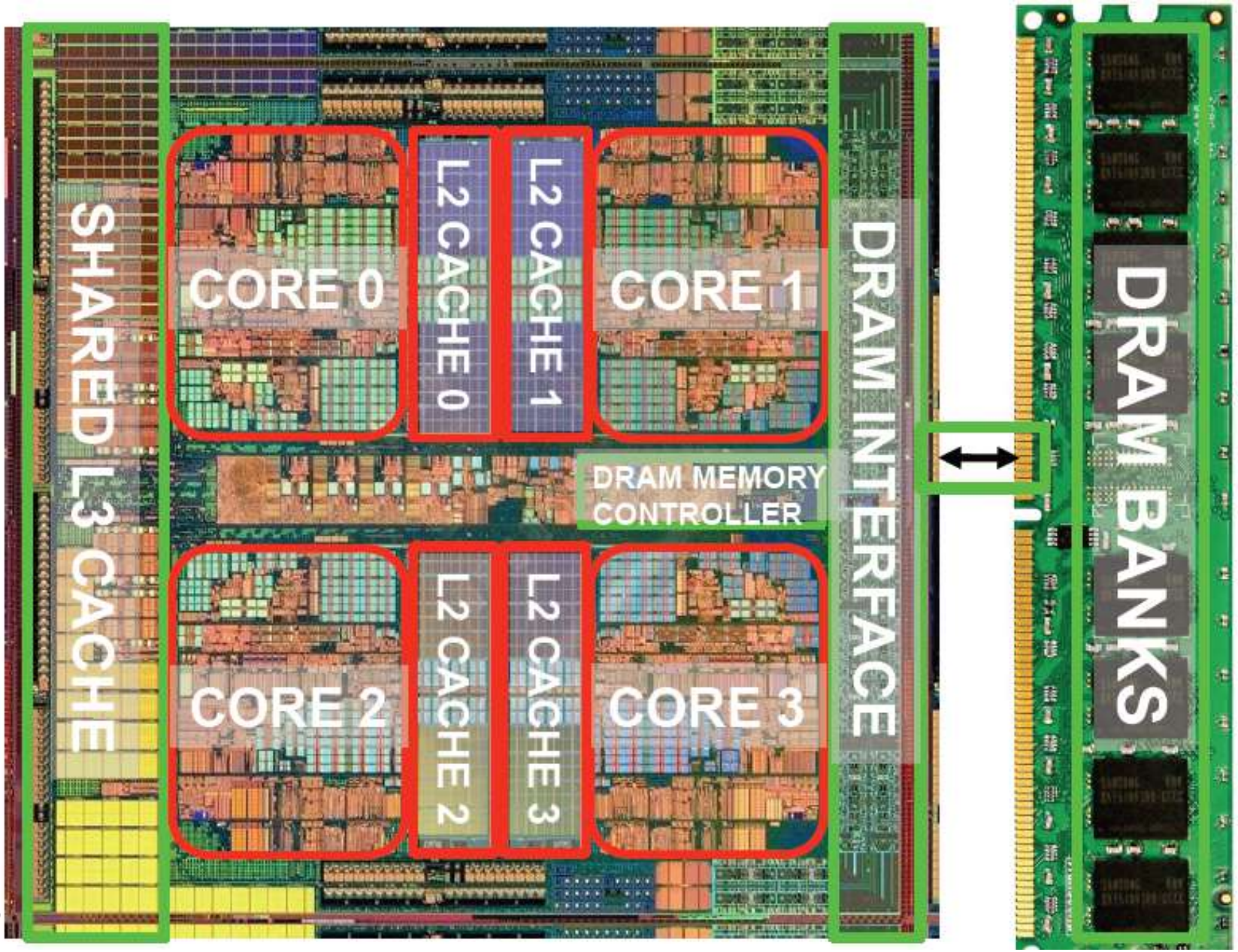
Tema 3. SRAM

- Funcionamiento
  - Misma estructura en matriz que la DRAM
  - Mismo método para leer celdas
  - Pero no hay tiempo de refresco ya que el bit no se pierde

Tema 3. SRAM

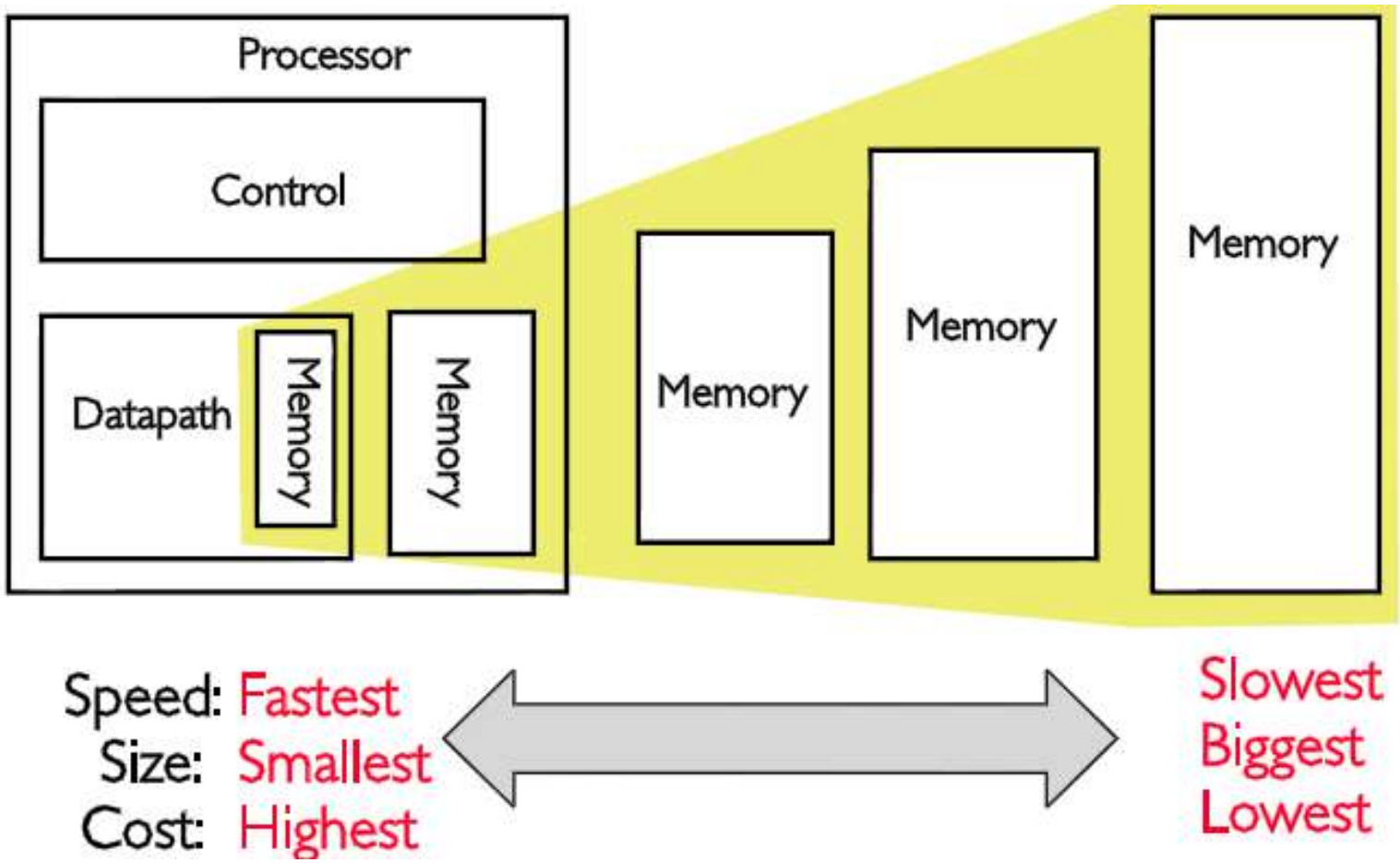
- De nivel 1: L1
  - Está integrada en el núcleo del procesador, trabajando a la misma velocidad que éste
  - La cantidad de memoria caché L1 varía de un procesador a otro, estando normalmente entra los 64kb y los 256kb
  - Esta memoria suele a su vez estar dividida en dos partes dedicadas, una para instrucciones y otra para datos
- De nivel 2: L2
  - Menos prestaciones de velocidad que L1 pero con más capacidad (512kB, hasta 2MB).
  - A diferencia de la caché 11, esta no está dividida, y su utilización está más encaminada a programas que al sistema
  - No puede ser de excesiva capacidad ya que sino el sistema es ineficiente.
- De nivel 3: L3
  - Del orden de MB. Con este nivel de memoria se agiliza el acceso a datos e instrucciones que no fueron localizadas en L1 o L2
  - Si no se encuentra el dato en ninguna de las 3, entonces se accederá a buscarlo en la memoria RAM (o L4 si se usa)
- De nivel 4: L4

Tema 3. SRAM y DRAM hoy

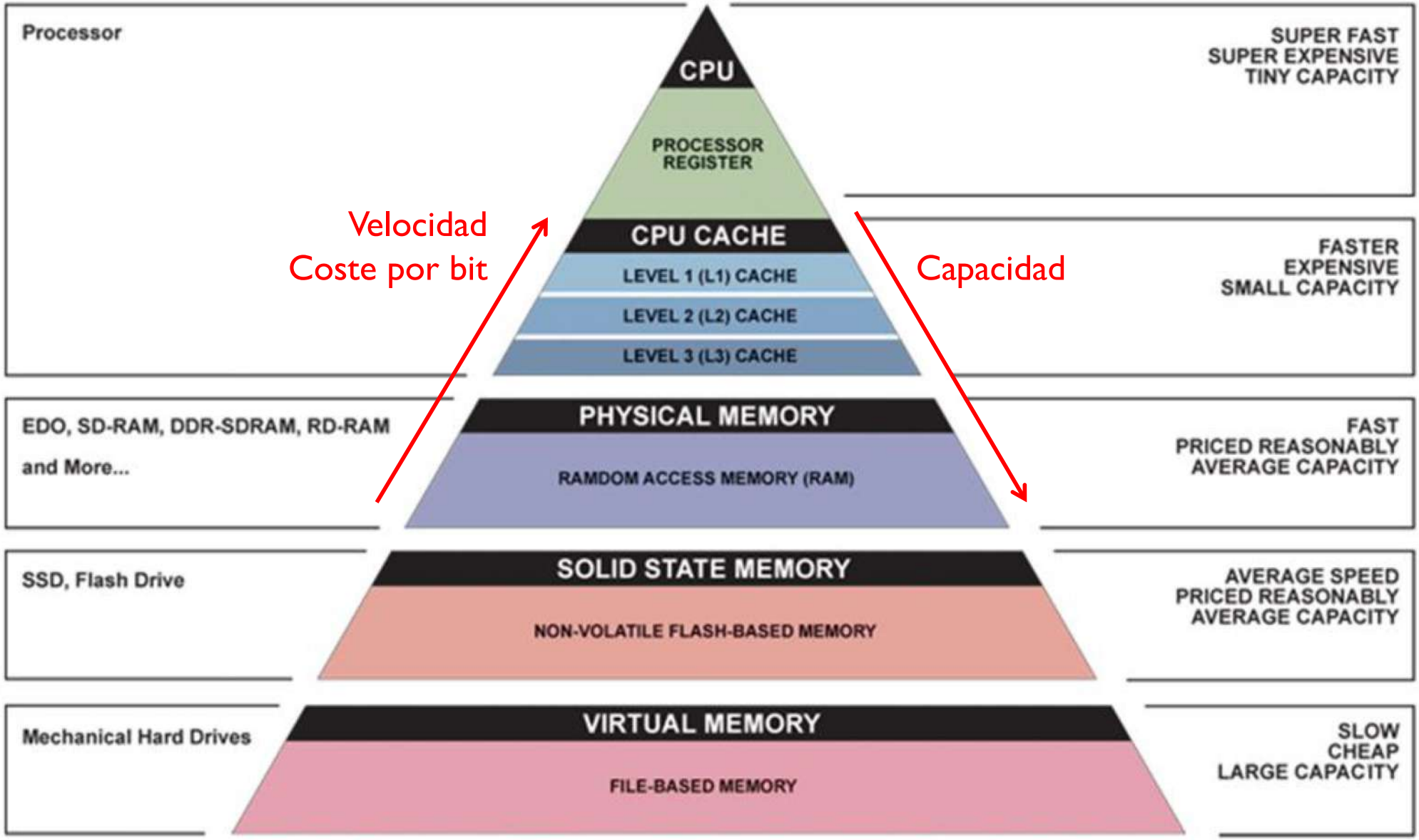


Tema 3. Jerarquía de memorias

- Objetivo: crear la ilusión de tener una memoria grande y rápida



Tema 3. Jerarquía de memorias



▲ Simplified Computer Memory Hierarchy
Illustration: Ryan J. Leng

Tema 3. Jerarquía de memorias

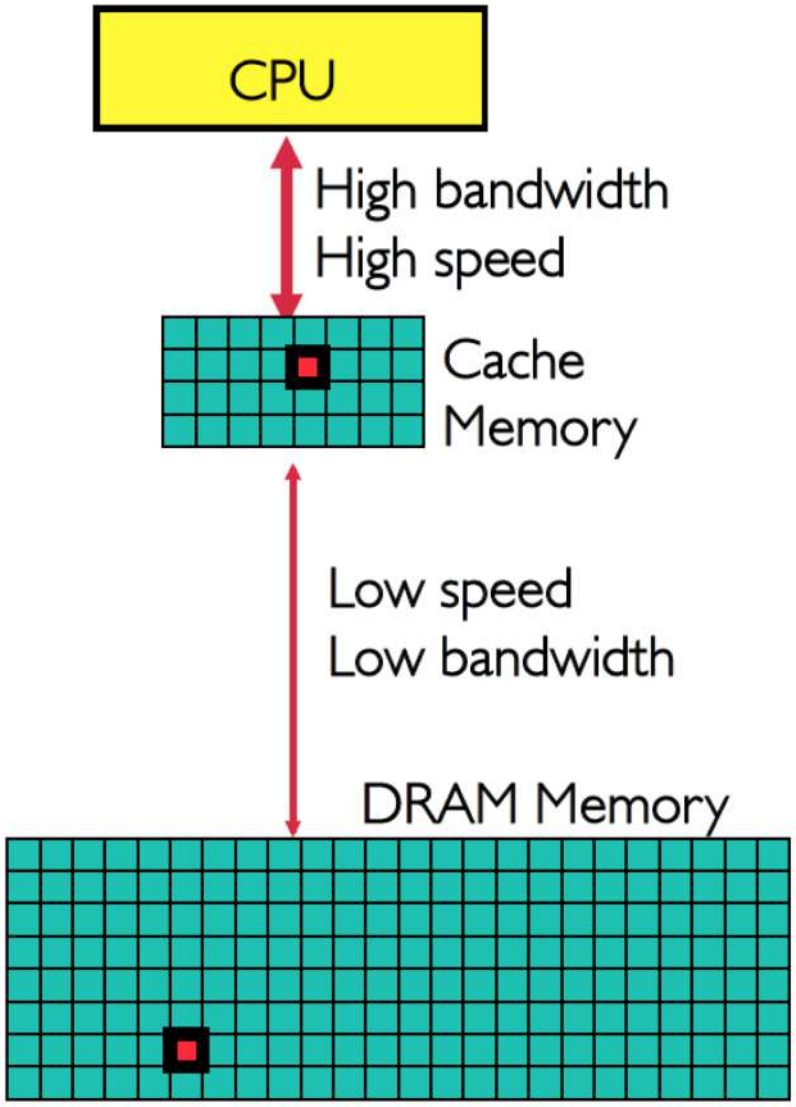
- La organización jerárquica de la memoria se basa en una característica que poseen la mayoría de los programas (al menos dentro de ciertos límites) → principio de localidad
- El principio de localidad establece que los programas acceden a una porción relativamente reducida del espacio de direcciones en un determinado lapso de tiempo.
- El principio tiene dos variantes
  - Localidad temporal: si un dato es referenciado en un determinado momento, es común que vuelva a ser referenciado poco tiempo después
  - Localidad espacial: cuando un dato es referenciado en un determinado momento, es común que los datos con direcciones "cercanas" también sea accedidos poco tiempo después

Tema 3. Jerarquía de memorias

- Cómo se aprovecha el principio de localidad en la jerarquía de memoria?
  - Se hace manteniendo los datos recientemente accedidos en las jerarquías altas, más cerca de la CPU → localidad temporal
  - También moviendo bloques de memoria contiguos hacia las jerarquías más altas → localidad espacial
- El principio de localidad es aplicado por distintos actores dentro de un sistema:
  - registros <> memoria: lo aplica el compilador o el programador de bajo nivel
  - memoria cache <> memoria: lo aplica el hardware
  - memoria <> disco: lo aplica el sistema operativo (memoria virtual) o el programador

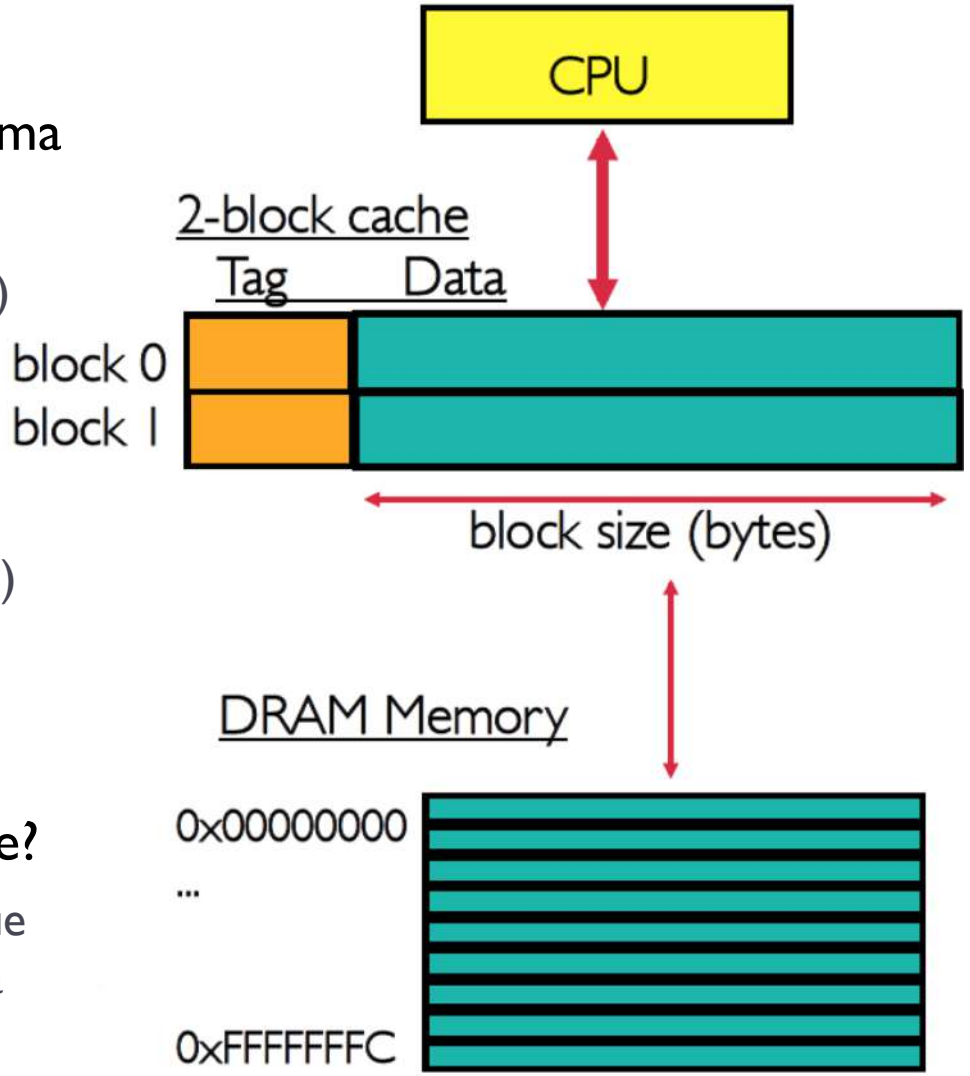
Tema 3. Jerarquía de memorias
Funcionamiento

- Se inserta una memoria SRAM entre la CPU y la DRAM
  - Esta memoria se llama cache
  - Funciona a la misma velocidad del procesador con latencia de 1 o 2 ciclos
- Contienen solo una porción de las instrucciones y los datos de un programa
- Generalmente no pueden mantener todas las instrucciones y todos los datos



Tema 3. Jerarquía de memorias
Funcionamiento

- Contiene solo una porción de las instrucciones y los datos de un programa
- ¿qué parte?
  - Solo las referencias (instrucciones y datos) usadas mas recientemente
  - Principio de localidad
- ¿cómo?
  - La cache está dividida en bloques (o líneas)
  - Cada bloque almacena el contenido de un conjunto contiguo de direcciones de memoria
- ¿cómo sabe la CPU que hay en la cache?
  - Cada bloque contiene un tag (etiqueta) que indica el inicio de la dirección de memoria almacenada en el bloque



Tema 3. Jerarquía de memorias
Funcionamiento

- Terminología
- **Hit**
  - Cuando se encuentra lo que se busca en en el nivel de memoria donde se lo está buscando
  - **Hit Rate**: tasa de acierto de encontrar un elemento de información en el lugar de la jerarquía en que se lo busca
  - **Hit Time**: tiempo de acceso promedio en el nivel de jerarquía considerado (donde se da el hit)
- **Miss**
  - Cuando no es encontrado en el lugar de la jerarquía donde se lo está buscando
  - En este caso es necesario ir a buscar el objeto a un nivel de jerarquía inferior
  - **Miss Rate**: tasa de fallos en encontrar un elemento de información en el lugar buscado (y coincide con 1 – Hit Rate)
  - **Miss Penalty**: tiempo de acceso promedio adicional requerido para acceder al elemento de información en el nivel de jerarquía inferior
    - Access time: tiempo para acceder a un nivel de memoria más bajo
    - Transfer time: tiempo para transferir el bloque
    - Típicamente ocurre que Hit Time << Miss Penalty

Tema 3. Jerarquía de memorias
Funcionamiento

- Comparación
- Supongamos este programa

```
0x00  label: add r1,r1,r2
0x04         bne r4,r1,label
0x08         sub r1,r1,r1
0x0C         exit
```

suma r2 a r1 y guarda en r1

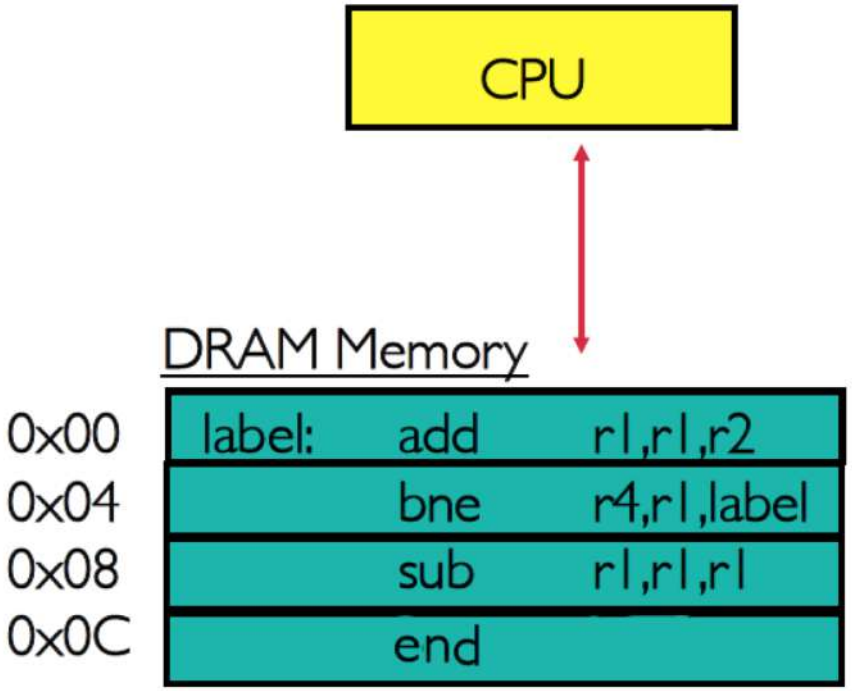
salta a label si r4 y r1 son diferentes

resta r1 a r1 y guarda en r1

- Asumimos
  - 5 ciclos para acceder a la memoria
  - 1 ciclo para acceder a la cache
  - 1 ciclo para ejecutar 1 instrucción

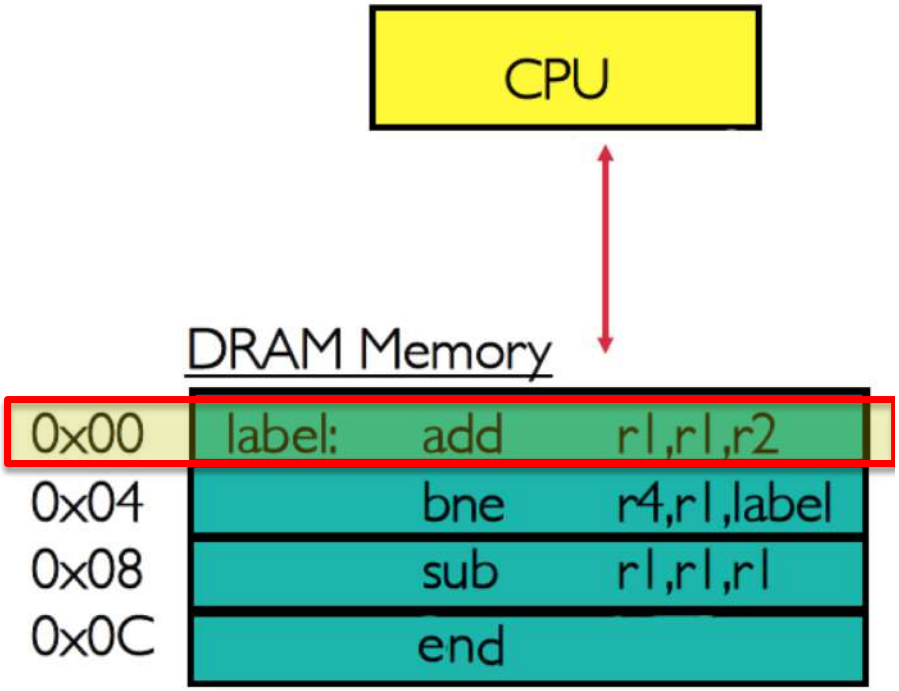
Tema 3. Jerarquía de memorias
Funcionamiento

- Sin cache
- Supongamos r1 = 0, r2 = 1, r4 = 2
  - 5 ciclos para acceder a la memoria
  - 1 ciclo para ejecutar 1 instrucción



Tema 3. Jerarquía de memorias
Funcionamiento

- Sin cache
- Supongamos r1 = 0, r2 = 1, r4 = 2
  - 5 ciclos para acceder a la memoria
  - 1 ciclo para ejecutar 1 instrucción



Fetch instrucción en 0x00 → 5 ciclos

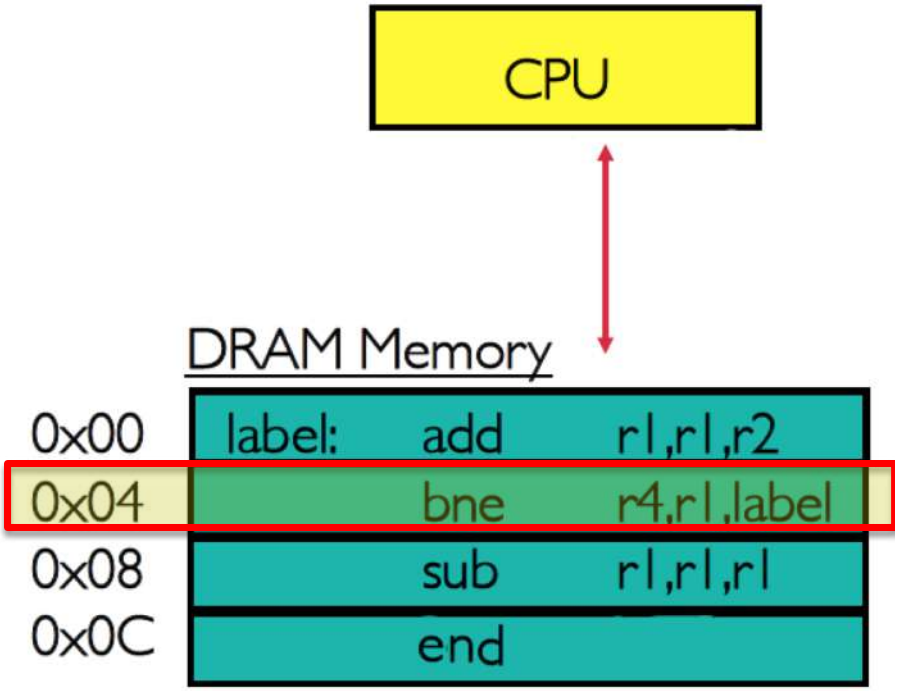
Añadir r2 a r1 y guardar en r1 → 1 ciclo

r1 = 1 + 0 = 1

Total = 5 + 1 +

Tema 3. Jerarquía de memorias
Funcionamiento

- Sin cache
- Supongamos r1 = 0, r2 = 1, r4 = 2
  - 5 ciclos para acceder a la memoria
  - 1 ciclo para ejecutar 1 instrucción



Fetch instrucción en 0x00 → 5 ciclos

Añadir r2 a r1 y guardar en r1 → 1 ciclo

r1 = 1 + 0 = 1

Fetch instrucción en 0x04 → 5 ciclos

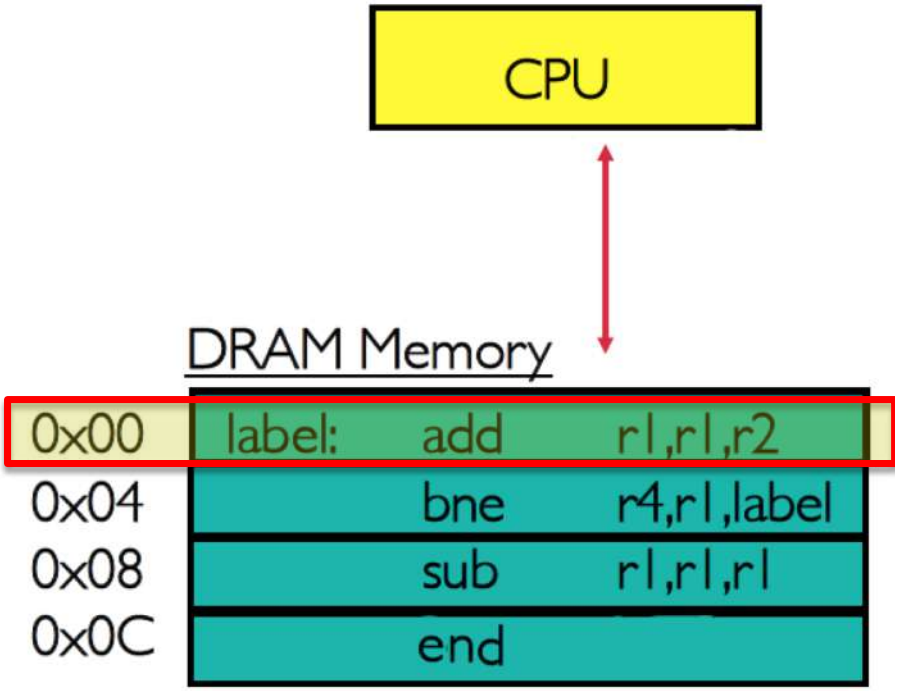
Comparar r4 con r1 y saltar si diferentes → 1 ciclo

r4 = 2, r1 = 1 → salta a label

Total = 5 + 1 + 5 + 1 +

Tema 3. Jerarquía de memorias
Funcionamiento

- Sin cache
- Supongamos r1 = 0, r2 = 1, r4 = 2
  - 5 ciclos para acceder a la memoria
  - 1 ciclo para ejecutar 1 instrucción



Fetch instrucción en 0x00 → 5 ciclos

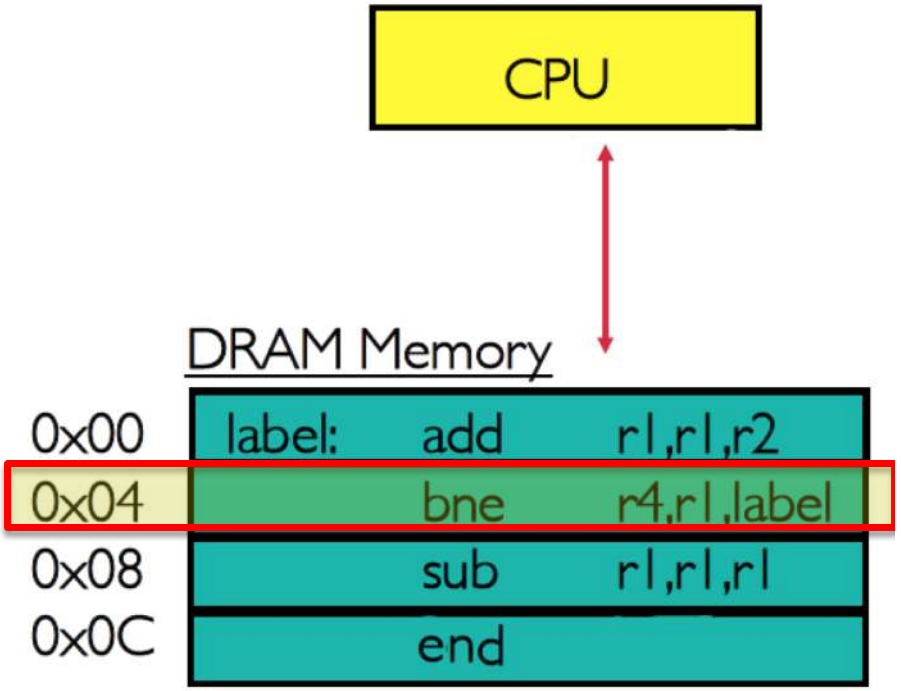
Añadir r2 a r1 y guardar en r1 → 1 ciclo

r1 = 1 + 1 = 2

Total = 5 + 1 + 5 + 1 + 5 + 1 +

Tema 3. Jerarquía de memorias
Funcionamiento

- Sin cache
- Supongamos r1 = 0, r2 = 1, r4 = 2
  - 5 ciclos para acceder a la memoria
  - 1 ciclo para ejecutar 1 instrucción



Fetch instrucción en 0x00 → 5 ciclos

Añadir r2 a r1 y guardar en r1 → 1 ciclo

r1 = 1 + 1 = 2

Fetch instrucción en 0x04 → 5 ciclos

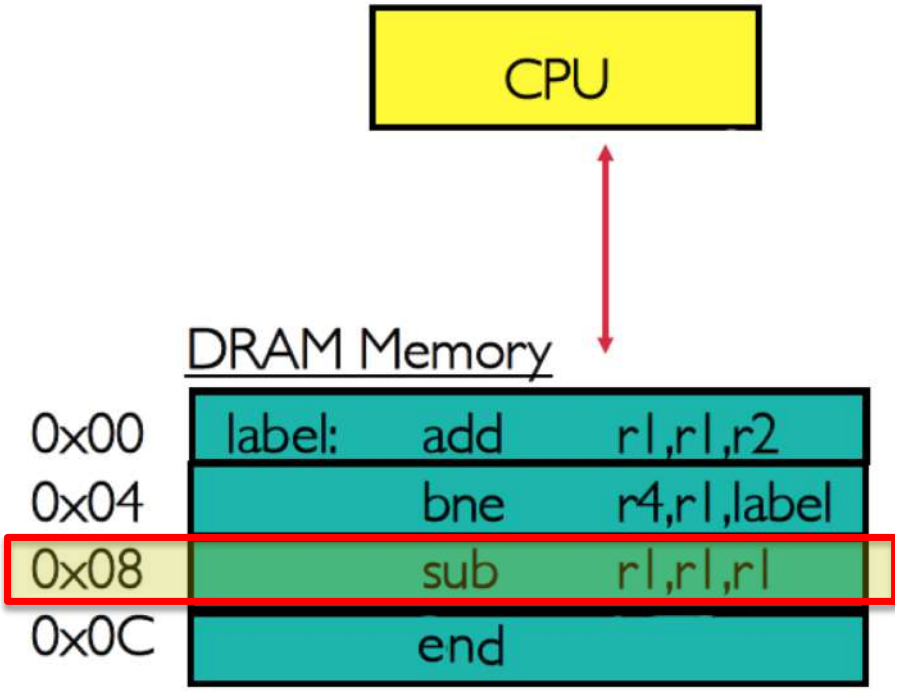
Comparar r4 con r1 y saltar se diferentes → 1 ciclo

r4 = 2, r1 = 2 → continua

Total = 5 + 1 + 5 + 1 + 5 + 1 + 5 + 1 +

Tema 3. Jerarquía de memorias
Funcionamiento

- Sin cache
- Supongamos r1 = 0, r2 = 1, r4 = 2
  - 5 ciclos para acceder a la memoria
  - 1 ciclo para ejecutar 1 instrucción



Fetch instrucción en 0x08 → 5 ciclos

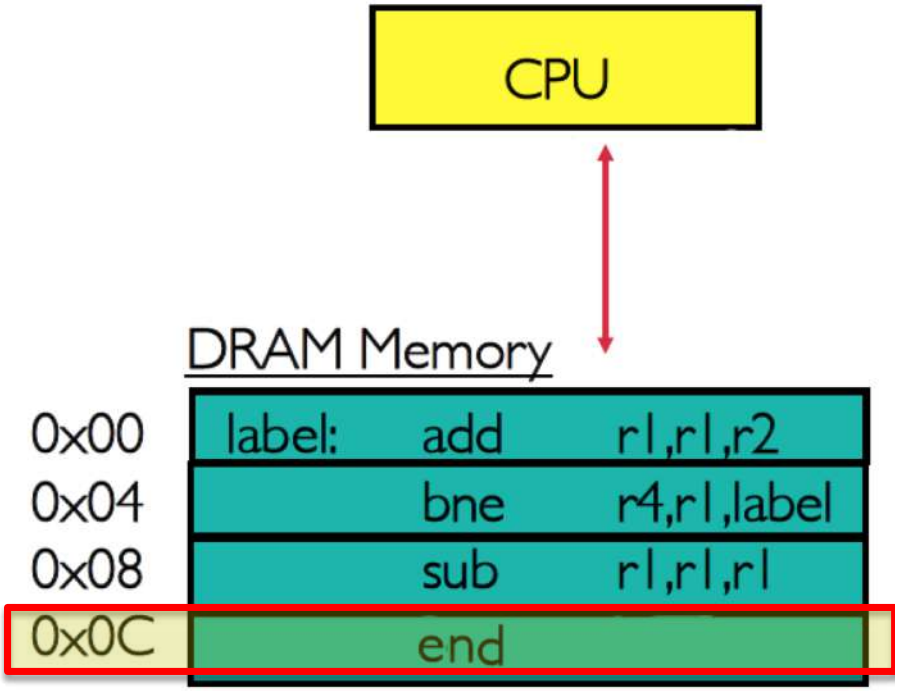
Restar r1 de r1 y guarda en r1 → 1 ciclo

r1 = 2 – 2 = 0

Total = 5 + 1 + 5 + 1 + 5 + 1 + 5 + 1 + 5 + 1 +

Tema 3. Jerarquía de memorias
Funcionamiento

- Sin cache
- Supongamos r1 = 0, r2 = 1, r4 = 2
  - 5 ciclos para acceder a la memoria
  - 1 ciclo para ejecutar 1 instrucción



Fetch instrucción en 0x08 → 5 ciclos

Restar r1 de r1 y guarda en r1 → 1 ciclo

r1 = 2 – 2 = 0

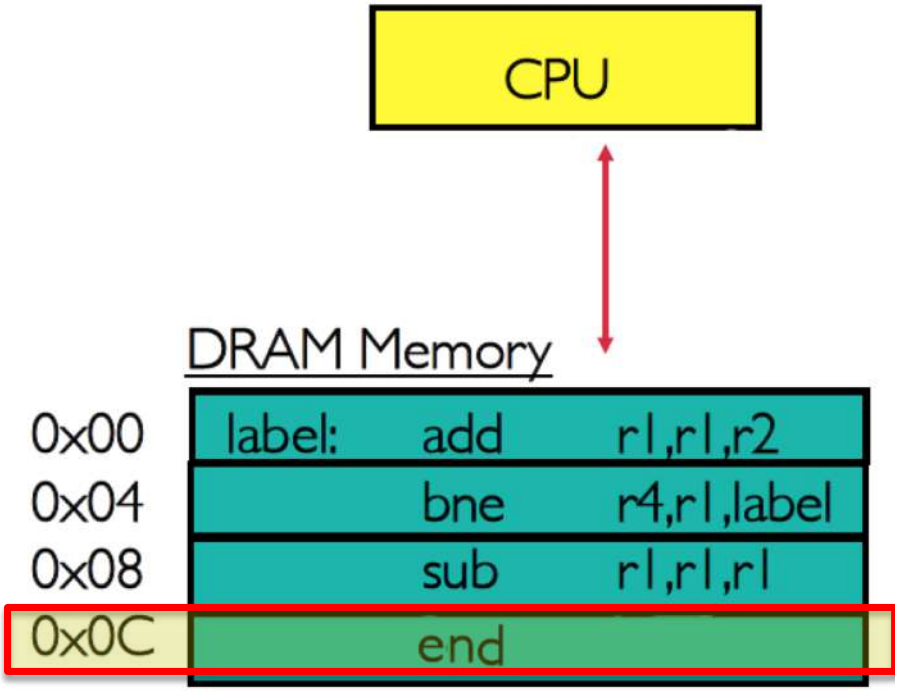
Fetch instrucción en 0x0c → 5 ciclos

Acabar → 1 ciclo

Total = 5 + 1 + 5 + 1 + 5 + 1 + 5 + 1 + 5 + 1 + 5 + 1 =

Tema 3. Jerarquía de memorias
Funcionamiento

- Sin cache
- Supongamos r1 = 0, r2 = 1, r4 = 2
  - 5 ciclos para acceder a la memoria
  - 1 ciclo para ejecutar 1 instrucción



Fetch instrucción en 0x08 → 5 ciclos

Restar r1 de r1 y guarda en r1 → 1 ciclo

r4 = 2 – 2 = 0

Fetch instrucción en 0x0c → 5 ciclos

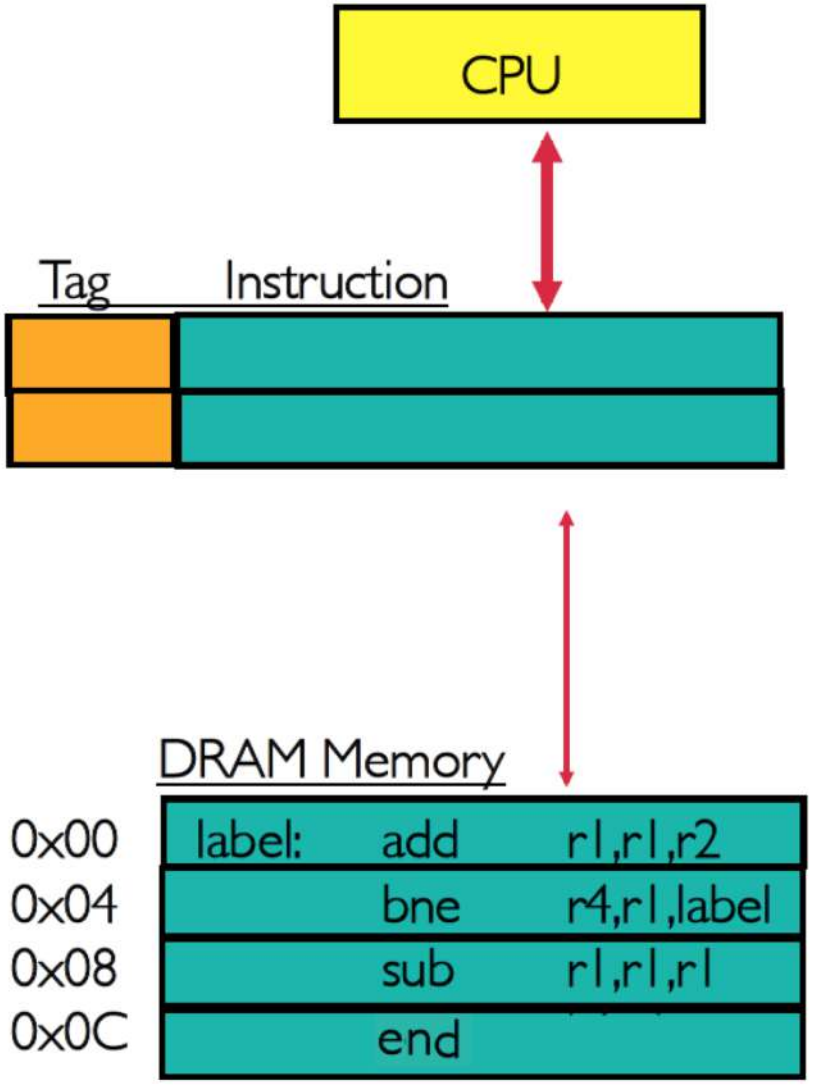
Acabar → 1 ciclo

Total = 5 + 1 + 5 + 1 + 5 + 1 + 5 + 1 + 5 + 1 + 5 + 1 = 36 ciclos

Si segmentado: fetch en paralelo a la ejecución = 5 + 1 + 5 + 5 + 5 + 5 + 5 = 31 ciclos

Tema 3. Jerarquía de memorias
Funcionamiento

- Con cache
- Supongamos r1 = 0, r2 = 1, r4 = 2
  - 5 ciclos para acceder a la memoria
  - 1 ciclo para acceder a la cache
  - 1 ciclo para ejecutar 1 instrucción

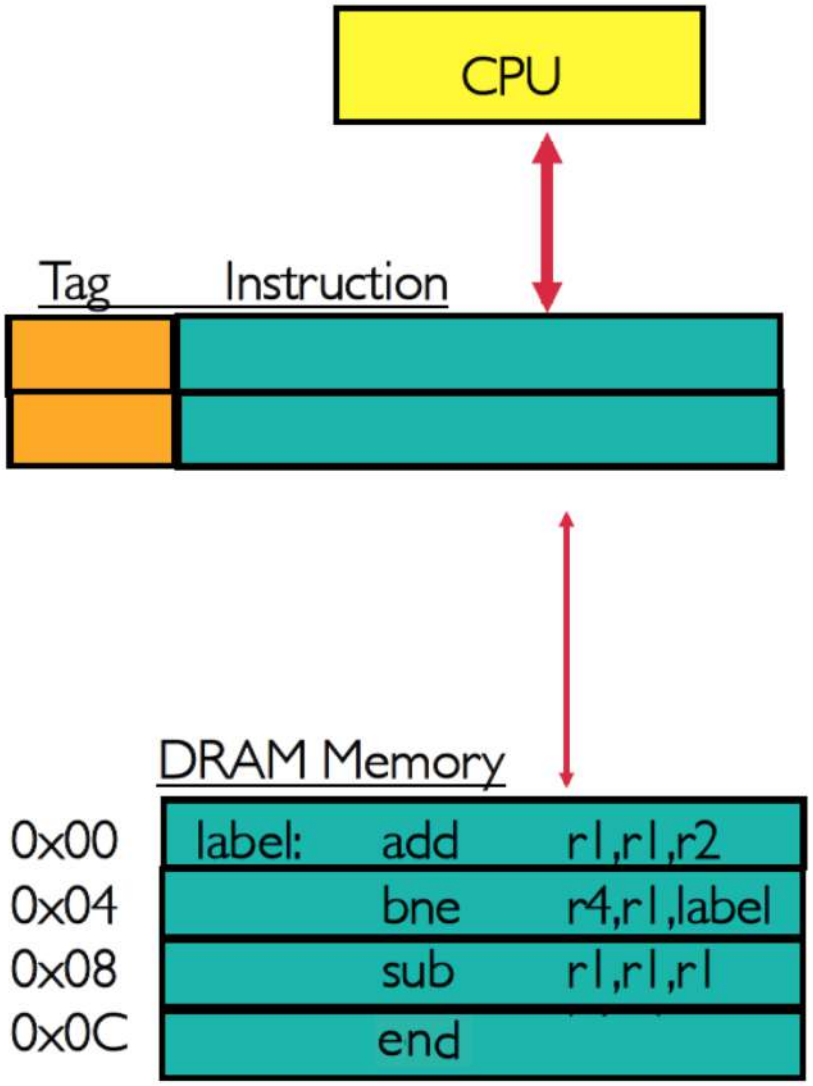


Tema 3. Jerarquía de memorias
Funcionamiento

- Con cache
- Supongamos r1 = 0, r2 = 1, r4 = 2
  - 5 ciclos para acceder a la memoria
  - 1 ciclo para acceder a la cache
  - 1 ciclo para ejecutar 1 instrucción

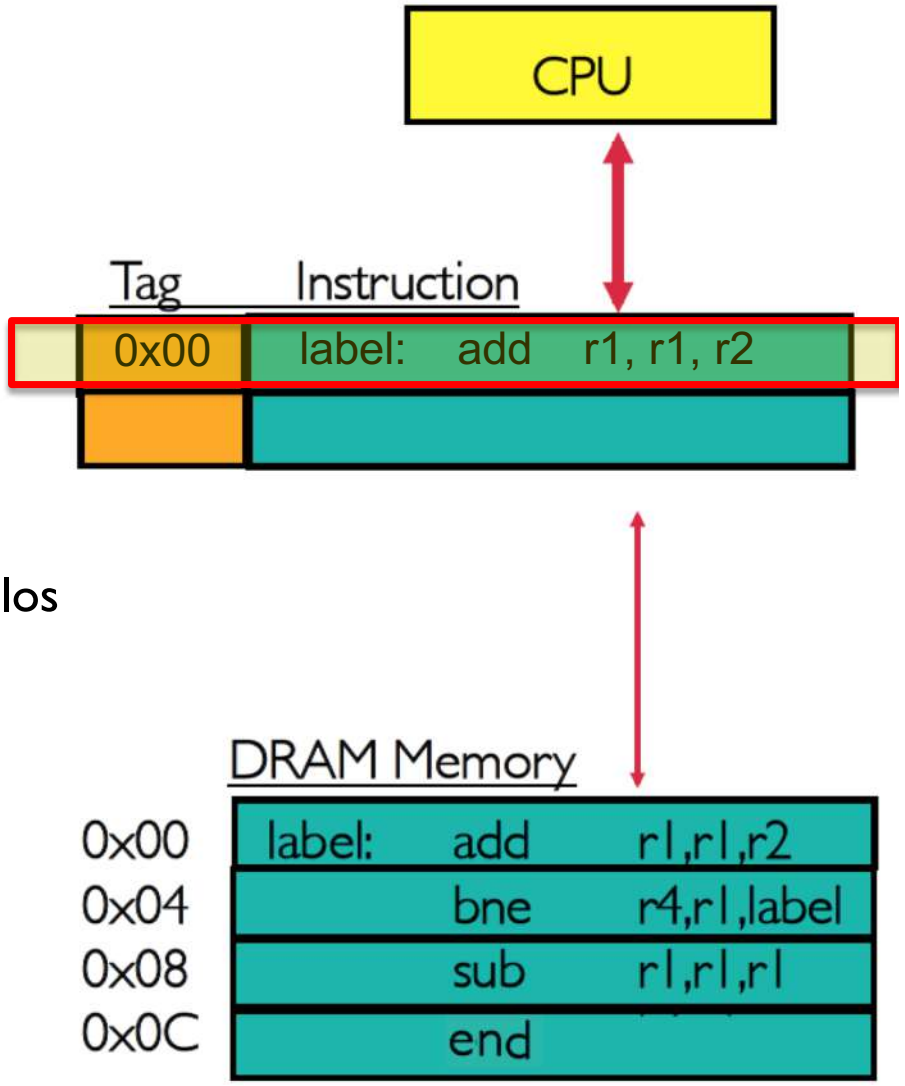
Fetch instrucción en 0x00 en la cache

→ No está (Miss) → se copia de la RAM → 5 ciclos



Tema 3. Jerarquía de memorias
Funcionamiento

- Con cache
- Supongamos r1 = 0, r2 = 1, r4 = 2
  - 5 ciclos para acceder a la memoria
  - 1 ciclo para acceder a la cache
  - 1 ciclo para ejecutar 1 instrucción



Fetch instrucción en 0x00 en la cache

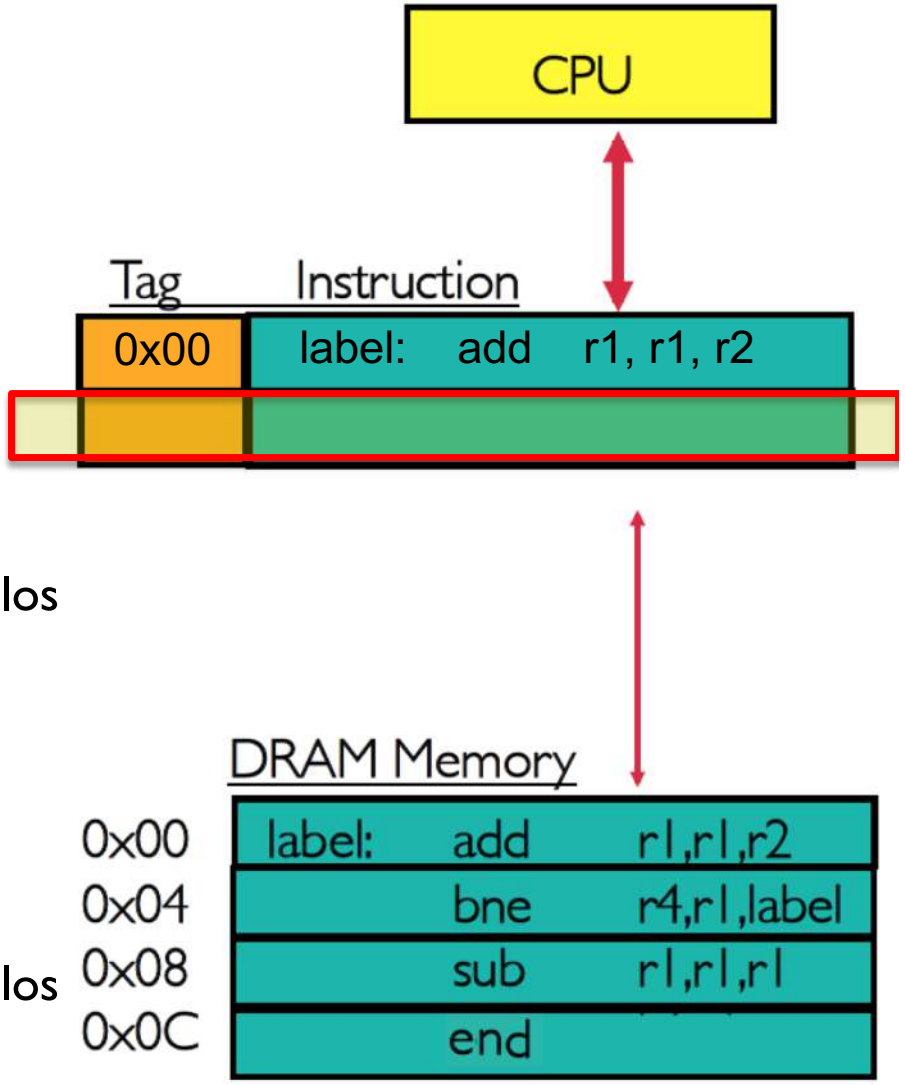
→ No está (Miss) → se copia de la RAM → 5 ciclos

Añadir r2 a r1 y guardar en r1 → 1 ciclo

r1 = 1 + 0 = 1

Tema 3. Jerarquía de memorias
Funcionamiento

- Con cache
- Supongamos r1 = 0, r2 = 1, r4 = 2
  - 5 ciclos para acceder a la memoria
  - 1 ciclo para acceder a la cache
  - 1 ciclo para ejecutar 1 instrucción



Fetch instrucción en 0x00 en la cache

→ No está (Miss) → se copia de la RAM → 5 ciclos

Añadir r2 a r1 y guardar en r1 → 1 ciclo

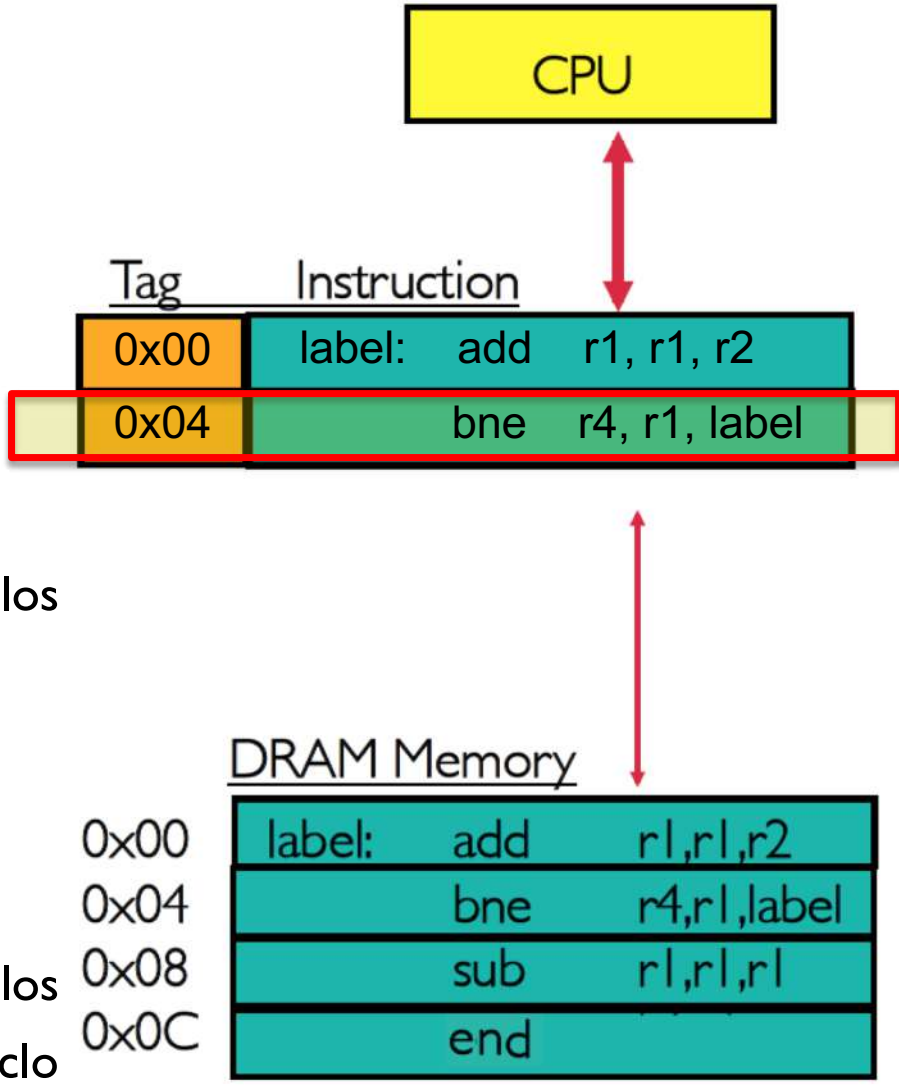
r1 = 1 + 0 = 1

Fetch instrucción en 0x04 en la cache

→ No está (Miss) → se copia de la RAM → 5 ciclos

Tema 3. Jerarquía de memorias
Funcionamiento

- Con cache
- Supongamos r1 = 0, r2 = 1, r4 = 2
  - 5 ciclos para acceder a la memoria
  - 1 ciclo para acceder a la cache
  - 1 ciclo para ejecutar 1 instrucción



Fetch instrucción en 0x00 en la cache

→ No está (Miss) → se copia de la RAM → 5 ciclos

Añadir r2 a r1 y guardar en r1 → 1 ciclo

r1 = 1 + 0 = 1

Fetch instrucción en 0x04 en la cache

→ No está (Miss) → se copia de la RAM → 5 ciclos

Comparar r4 con r1 y saltar se diferentes → 1 ciclo

r4 = 2, r1 = 1 → salta a label

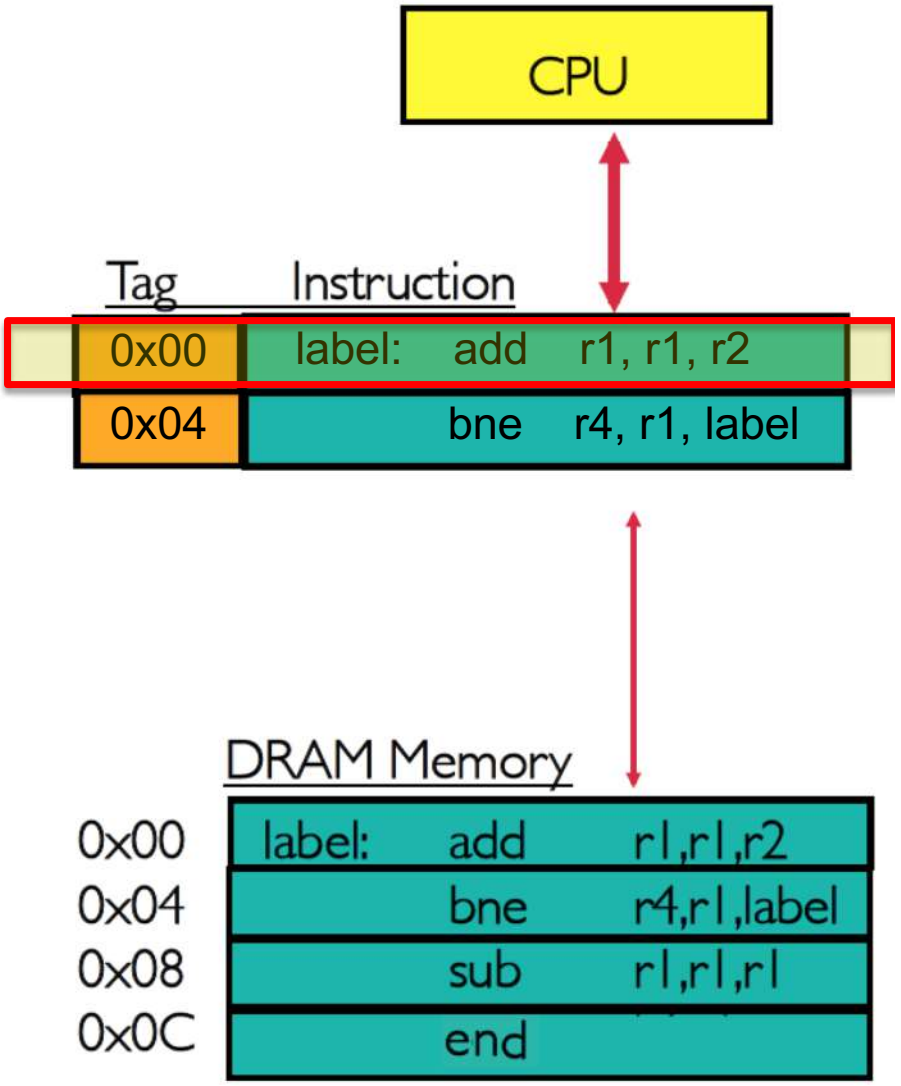
Tema 3. Jerarquía de memorias
Funcionamiento

- Con cache
- Supongamos r1 = 0, r2 = 1, r4 = 2
  - 5 ciclos para acceder a la memoria
  - 1 ciclo para acceder a la cache
  - 1 ciclo para ejecutar 1 instrucción

Fetch instrucción en 0x00 en la cache → 1 ciclo

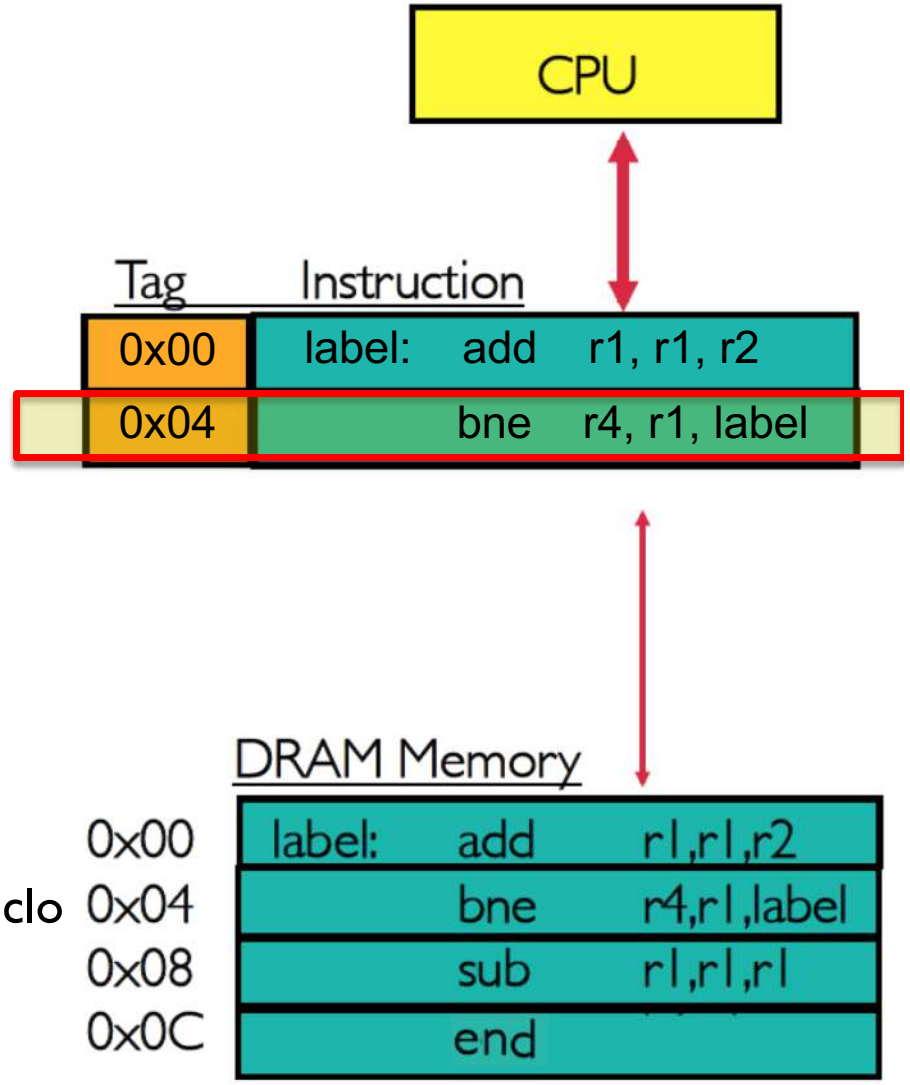
Añadir r2 a r1 y guardar en r1 → 1 ciclo

r1 = 1 + 1 = 2



Tema 3. Jerarquía de memorias
Funcionamiento

- Con cache
- Supongamos r1 = 0, r2 = 1, r4 = 2
  - 5 ciclos para acceder a la memoria
  - 1 ciclo para acceder a la cache
  - 1 ciclo para ejecutar 1 instrucción



Fetch instrucción en 0x00 en la cache → 1 ciclo

Añadir r2 a r1 y guardar en r1 → 1 ciclo

r1 = 1 + 1 = 2

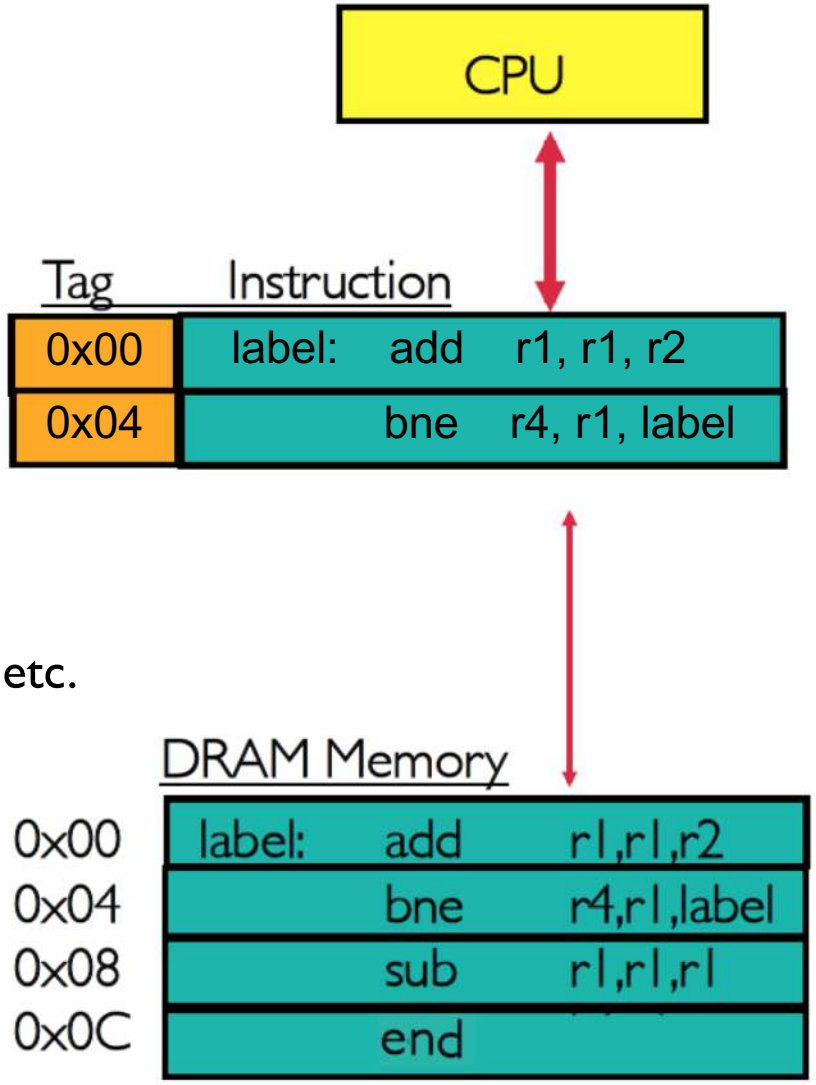
Fetch instrucción en 0x04 en la cache → 1 ciclo

Comparar r4 con r1 y saltar se diferentes → 1 ciclo

r4 = 2, r1 = 2 → continua

Tema 3. Jerarquía de memorias
Funcionamiento

- Con cache
- Supongamos r1 = 0, r2 = 1, r4 = 2
  - 5 ciclos para acceder a la memoria
  - 1 ciclo para acceder a la cache
  - 1 ciclo para ejecutar 1 instrucción



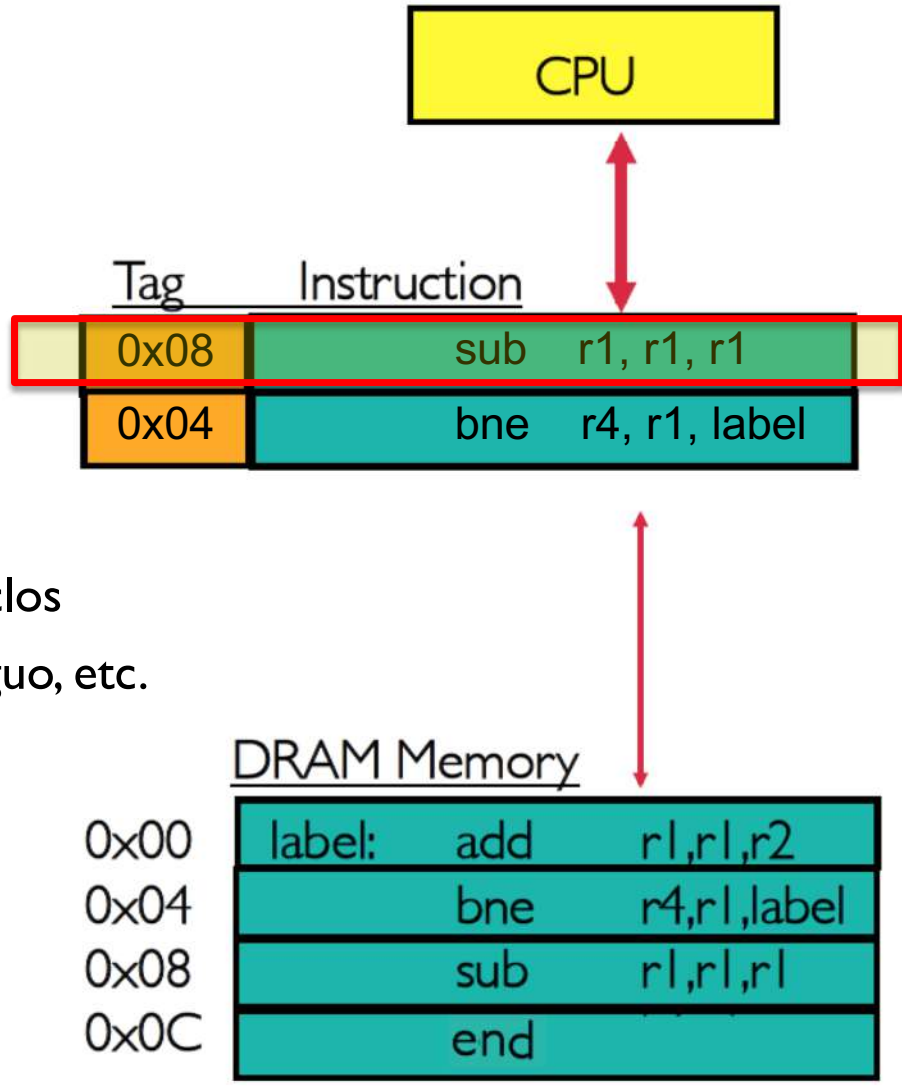
Fetch instrucción en 0x08 en la cache

→ No está (Miss) → se copia de la RAM → 5 ciclos

Donde? Varias técnicas: random, bloque más antiguo, etc.

Tema 3. Jerarquía de memorias
Funcionamiento

- Con cache
- Supongamos r1 = 0, r2 = 1, r4 = 2
  - 5 ciclos para acceder a la memoria
  - 1 ciclo para acceder a la cache
  - 1 ciclo para ejecutar 1 instrucción



Fetch instrucción en 0x08 en la cache

→ No está (Miss) → se copia de la RAM → 5 ciclos

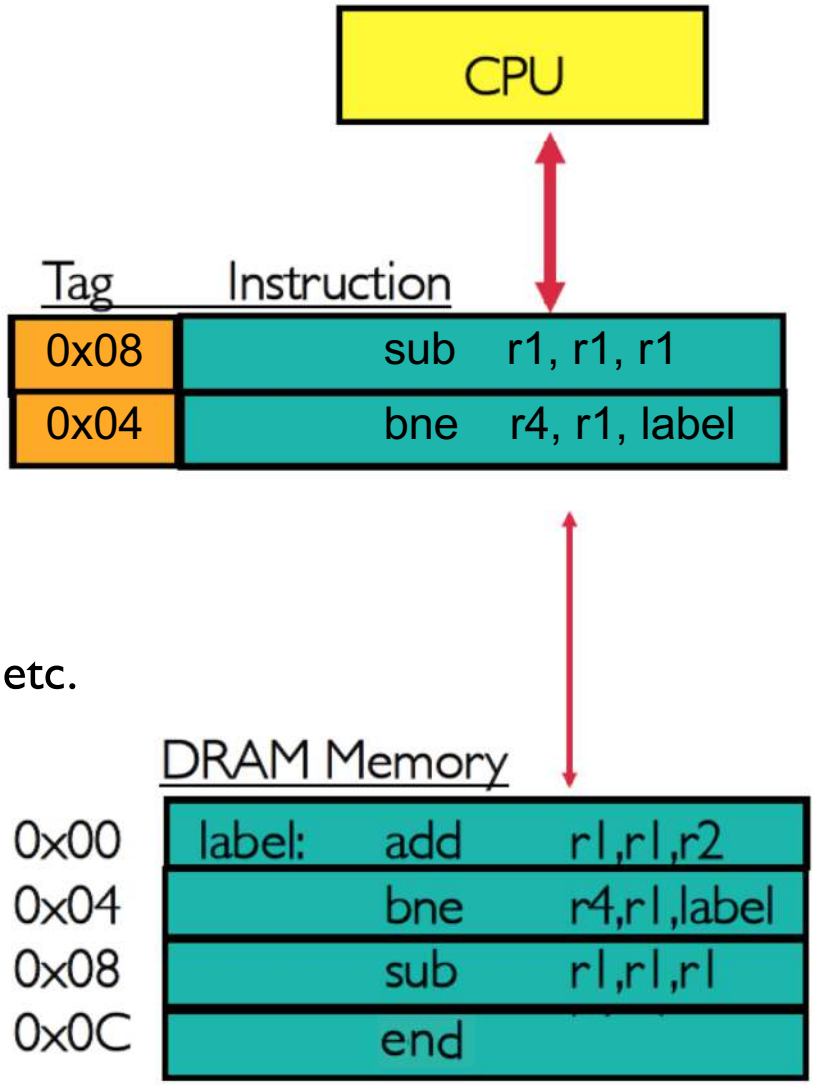
Donde? Varias técnicas: random, bloque más antiguo, etc.

Restar r1 a r1 y guardar en r1 → 1 ciclo

r1 = 2 - 2 = 0

Tema 3. Jerarquía de memorias
Funcionamiento

- Con cache
- Supongamos r1 = 0, r2 = 1, r4 = 2
  - 5 ciclos para acceder a la memoria
  - 1 ciclo para acceder a la cache
  - 1 ciclo para ejecutar 1 instrucción



Fetch instrucción en 0x08 en la cache

→ No está (Miss) → se copia de la RAM → 5 ciclos

Donde? Varias técnicas: random, bloque más antiguo, etc.

Restar r1 a r1 y guardar en r1 → 1 ciclo

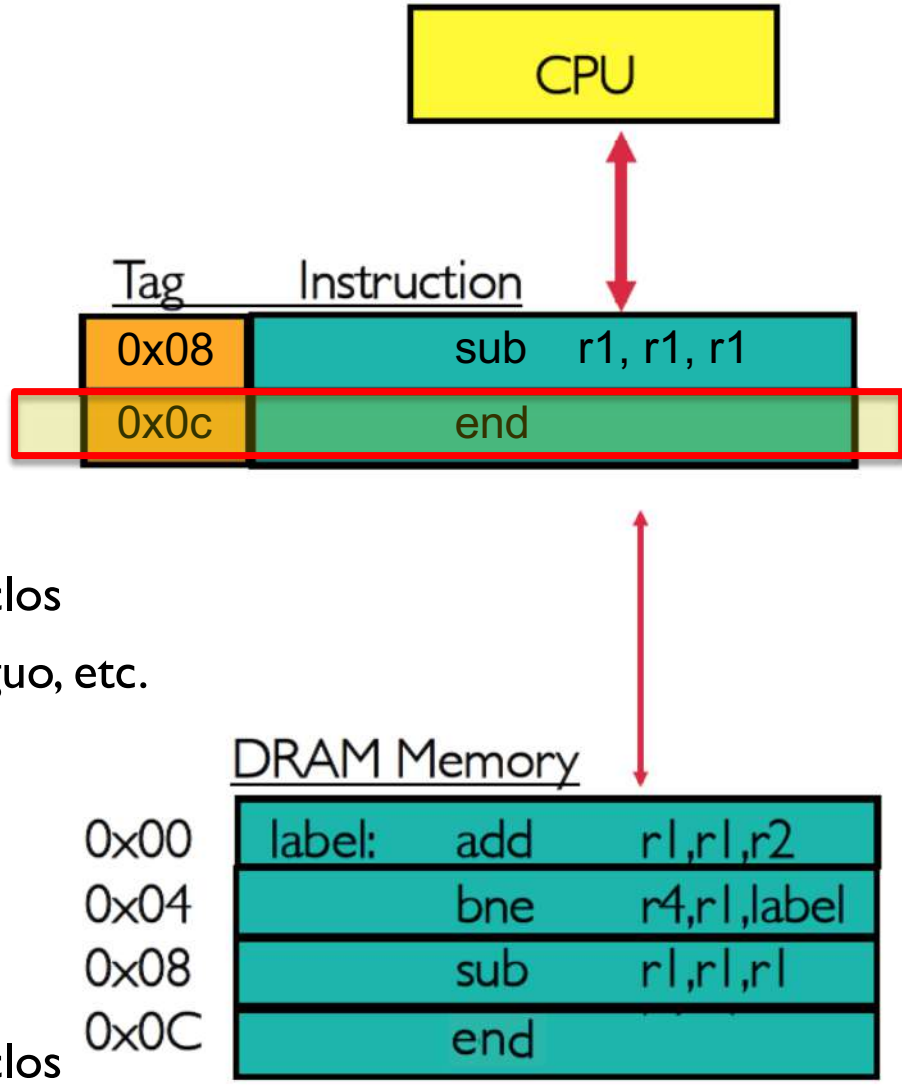
r1 = 2 - 2 = 0

Fetch instrucción en 0x0c en la cache

→ No está (Miss) → se copia de la RAM → 5 ciclos

Tema 3. Jerarquía de memorias
Funcionamiento

- Con cache
- Supongamos r1 = 0, r2 = 1, r4 = 2
  - 5 ciclos para acceder a la memoria
  - 1 ciclo para acceder a la cache
  - 1 ciclo para ejecutar 1 instrucción



Fetch instrucción en 0x08 en la cache

→ No está (Miss) → se copia de la RAM → 5 ciclos

Donde? Varias técnicas: random, bloque más antiguo, etc.

Restar r1 a r1 y guardar en r1 → 1 ciclo

r1 = 2 - 2 = 0

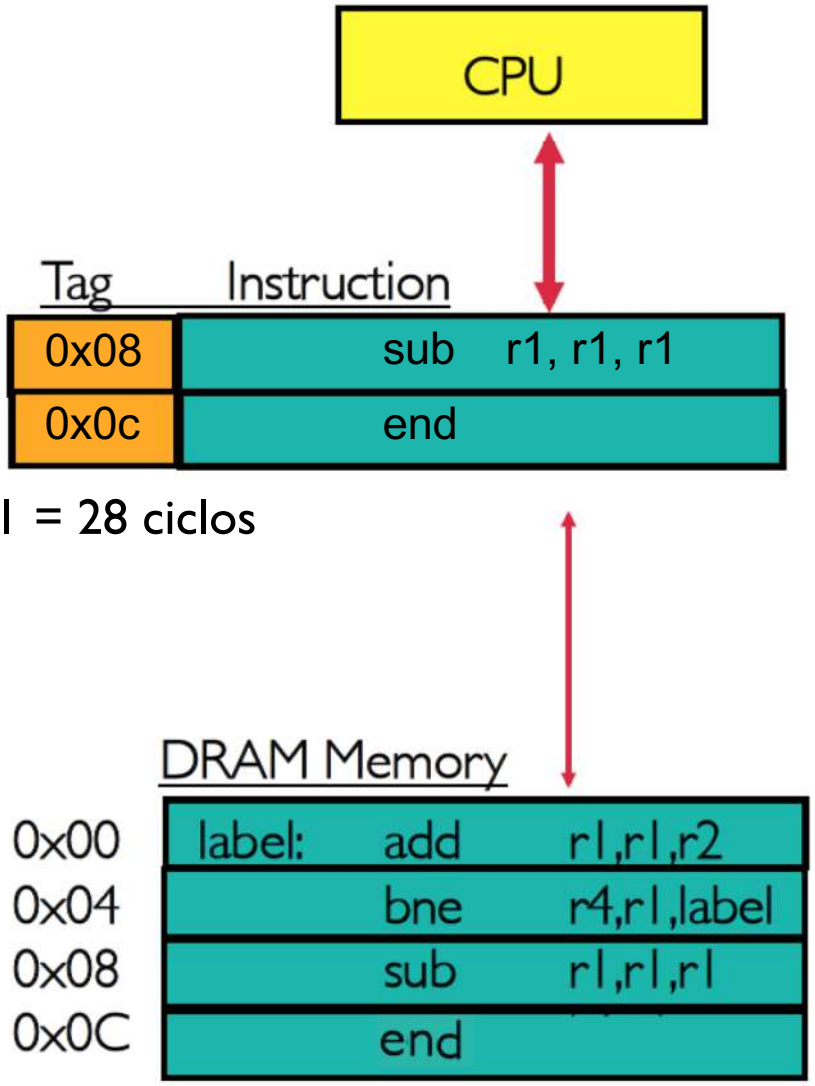
Fetch instrucción en 0x0c en la cache

→ No está (Miss) → se copia de la RAM → 5 ciclos

Acabar → 1 ciclo

Tema 3. Jerarquía de memorias
Funcionamiento

- Con cache
- Supongamos r1 = 0, r2 = 1, r4 = 2
  - 5 ciclos para acceder a la memoria
  - 1 ciclo para acceder a la cache
  - 1 ciclo para ejecutar 1 instrucción



Total = 5 + 1 + 5 + 1 + 1 + 1 + 1 + 1 + 5 + 1 + 5 + 1 = 28 ciclos

Si segmentado: fetch en paralelo a la ejecución =

= 5 + 1 + 5 + 1 + 1 + 5 + 5 = 23 ciclos

Tema 3. Jerarquía de memorias

Funcionamiento

Cycle	Address	Op/Instruction
1 - 5		FETCH 0x00
6	0x00	add r1,r1,r2
6-10		FETCH 0x04
11	0x04	bne r4,r1,label
11 - 15		FETCH 0x00
16	0x00	add r1,r1,r2
16 - 20		FETCH 0x04
21	0x04	bne r4,r1,label
21 - 25		FETCH 0x08
26	0x08	sub r1,r1,r1
26- 30		FETCH 0x0C
31	0x0C	

Cycle	Address	Op/Instruction
1 - 5		FETCH 0x00
6	0x00	add r1,r1,r2
6-10		FETCH 0x04
11	0x04	bne r4,r1,0x00
11		FETCH 0x00
12	0x00	add r1,r1,r2
12		FETCH 0x04
13	0x04	bne r4,r1,0x00
13 - 17		FETCH 0x08
18	0x08	sub r1,r1,r1
18-22		FETCH 0x0C
23	0x0C	

Hit in cache

Oops, missed in the cache

Tema 3. Jerarquía de memorias
Funcionamiento

- Cuatro problemas
- Donde hay que posicionar el bloque en la cache? Block placement
- Como identificar un bloque en la cache? Block identification
- Que bloque hay que sustituir cuando se da un Miss? Block replacement
- Que pasa cuando se escribe en la cache? Write strategy

Tema 3. Rendimiento

- Tiempo medio de acceso a la memoria Ta

$$Ta = \text{Hit Time} + (\text{Miss Rate} \times \text{Miss Penalty})$$

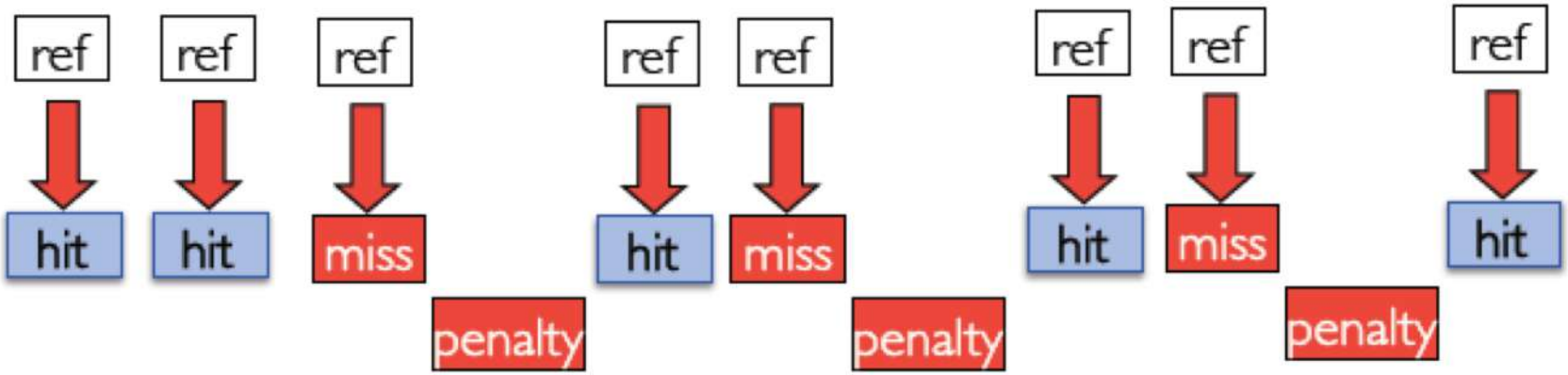
  - **Hit Time**: tiempo de acceso promedio en el nivel de jerarquía considerado (donde se da el hit)
  - **Miss Rate**: tasa de fallos en encontrar un elemento de información en el lugar buscado (y coincide con 1 – Hit Rate)
  - **Miss Penalty**: tiempo de acceso promedio adicional requerido para acceder al elemento de información en el nivel de jerarquía inferior

- Pueden ser en segundos (ns) o en ciclos del reloj

Tema 3. Rendimiento

- Tiempo medio de acceso a la memoria Ta

$$Ta = \text{Hit Time} + (\text{Miss Rate} \times \text{Miss Penalty})$$



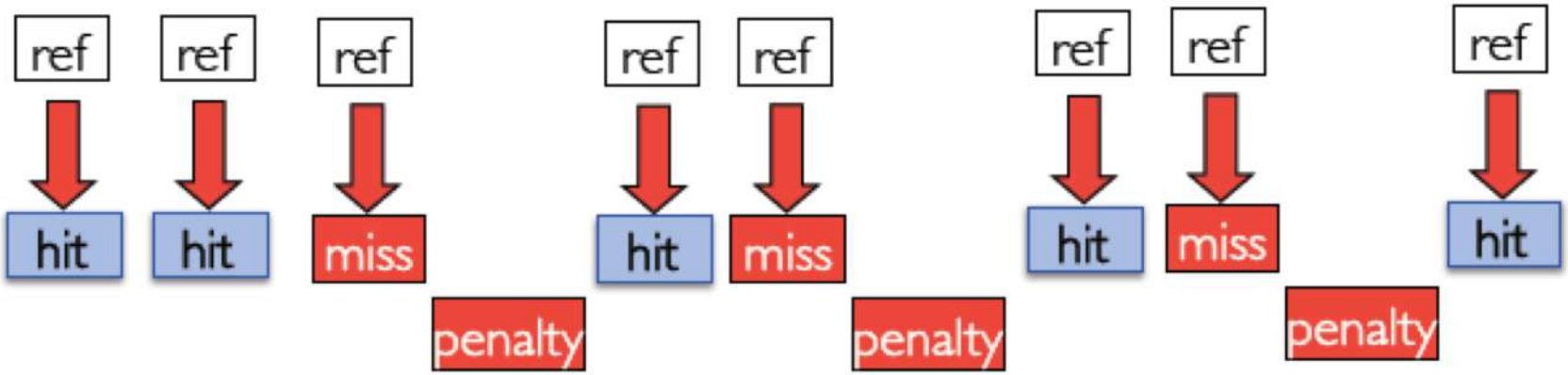
- Hit: 1 ciclo
- Penalty: 10 ciclos

Ta = ?

Tema 3. Rendimiento

- Tiempo medio de acceso a la memoria Ta

$$Ta = \text{Hit Time} + (\text{Miss Rate} \times \text{Miss Penalty})$$



- Hit: 1 ciclo
- Penalty: 10 ciclos

- MissRate = 3 miss / 8 total

$$Ta = 1 \text{ ciclo} + ((3 \text{ miss} / 8 \text{ total}) \times 10 \text{ ciclos}) = 4{,}75 \text{ ciclos}$$

Tema 3. Rendimiento

- Ejemplo
- Un ordenador tiene un Hit Time de 1 ciclo y un Miss Penalty de 10 ciclos. El Miss Rate es de 10%.
  Calcular el tiempo de acceso medio

Ta = ?

Tema 3. Rendimiento

- Ejemplo
- Un ordenador tiene un Hit Time de 1 ciclo y un Miss Penalty de 10 ciclos. El Miss Rate es de 10%.
  Calcular el tiempo de acceso medio

$$Ta = 1 + 10\% \times 10 = 2 \text{ ciclos}$$

Tema 3. Rendimiento

- Ejemplo
- Un ordenador tiene un Hit Time de 1 ciclo y un Miss Penalty de 10 ciclos. El Miss Rate es de 10%.
  Calcular el tiempo de acceso medio

$$Ta = 1 + 10\% \times 10 = 2 \text{ ciclos}$$

- Frecuencia del reloj f = 1 GHz

Tema 3. Rendimiento

- Ejemplo
- Un ordenador tiene un Hit Time de 1 ciclo y un Miss Penalty de 10 ciclos. El Miss Rate es de 10%.
  Calcular el tiempo de acceso medio

$$Ta = 1 + 10\% \times 10 = 2 \text{ ciclos}$$

- Frecuencia del reloj f = 1 GHz

$$1 \text{ ciclo} = 1 / f = 1 \text{ ns}$$

$$Ta = 2 \text{ ciclos} \times 1 \text{ ns} = 2 \text{ ns}$$

Tema 3. Rendimiento

- Ejemplo
- Un ordenador tiene un Hit Time de 1 ciclo y un Miss Penalty de 10 ciclos. Un programa accede a 1000 referencias (instrucciones y datos) en memoria, de estos 900 serán Hit a la primera y 100 serán Miss.
  Calcular el tiempo de acceso medio

  Hit Rate = 900 / 1000 = 0.9 = 90 %

  Miss Rate = 1 – Hit Rate = 10%

  Ta = 1 + 10% x 10 = 2 ciclos

Tema 3. Rendimiento

- Ejemplo
- Un ordenador tiene un Hit Time de 1 ciclo y un Miss Penalty de 10 ciclos. Un programa accede a 1000 referencias (instrucciones y datos) en memoria, de estos 900 serán Hit a la primera y 100 serán Miss.
  Calcular el tiempo de acceso medio

  Hit Rate = 900 / 1000 = 0.9 = 90 %

  Miss Rate = 1 – Hit Rate = 10%

  Ta = 1 + 10% x 10 = 2 ciclos

También se puede calcular como

Total ciclos = 900 Hits x 1 ciclo + 100 Miss x 11 ciclos (10 Miss + 1 Hit) = 2000

Ta = 2000 ciclos / 1000 referencias = 2 ciclos

Tema 3. Ejercicios

- Un sistema dispone de dos niveles de jerarquía de memoria: caché y memoria principal. Los tiempos de acceso son 5 ns y 50 ns, respectivamente y el porcentaje de aciertos en caché es del 95%. ¿Qué porcentaje de mejora se ha obtenido en el tiempo de ciclo al haber introducido la caché?

Tema 3. Ejercicios

- Un sistema dispone de dos niveles de jerarquía de memoria: caché y memoria principal. Los tiempos de acceso son 5 ns y 50 ns, respectivamente y el porcentaje de aciertos en caché es del 95%. ¿Qué porcentaje de mejora se ha obtenido en el tiempo de ciclo al haber introducido la caché?

Sin caché

Ta = 50 ns

Tema 3. Ejercicios

- Un sistema dispone de dos niveles de jerarquía de memoria: caché y memoria principal. Los tiempos de acceso son 5 ns y 50 ns, respectivamente y el porcentaje de aciertos en caché es del 95%. ¿Qué porcentaje de mejora se ha obtenido en el tiempo de ciclo al haber introducido la caché?

Sin caché

Ta = 50 ns

Con caché

Hit Time = 5 ns

Miss Penalty = 50 ns (+ transferir el bloque, despreciable)

Miss Rate = 1 - 95% = 5%

Ta = 5 ns + 0,05 x 50 ns = 7,5 ns

Speedup = 50 / 7,5 = 6,66

Tema 3. Ejercicios

- Un sistema dispone de dos niveles de jerarquía de memoria: caché y memoria principal. Los tiempos de acceso son 4 ns y 60 ns, respectivamente. Se quiere que el tiempo medio de acceso sea inferior a 6 ns, determina el porcentaje de aciertos.

Tema 3. Ejercicios

- Un sistema dispone de dos niveles de jerarquía de memoria: caché y memoria principal. Los tiempos de acceso son 4 ns y 60 ns, respectivamente. Se quiere que el tiempo medio de acceso sea inferior a 6 ns, determina el porcentaje de aciertos.

$$Ta = HitTime + MissRate \times MissPenalty =$$
$$= 4\ ns + MissRate \times 60\ ns =$$
$$= 4\ ns + (1 - HitRate) \times 60\ ns < 6\ ns$$

$$HitRate > 96{,}7\%$$

Tema 3. Ejercicios

- La frecuencia de una CPU es 1 GHz y tiene 1 nivel de cache. El CPI medio de ejecución de instrucciones es 1.1 y hay 50% aritméticas/lógicas, 30% cargar/guardar en la memoria y 20% saltos. El Hit Time es de 1 ciclo, Miss Rate de 1.5% y Miss Penalty de 50 ciclos. Determinar el CPI medio considerando la memoria.

Tema 3. Ejercicios

- La frecuencia de una CPU es 1 GHz y tiene 1 nivel de cache. El CPI medio de ejecución de instrucciones es 1,1 y hay 50% aritméticas/lógicas, 30% cargar/guardar en la memoria y 20% saltos. El Hit Time es de 1 ciclo, Miss Rate de 1.5% y Miss Penalty de 50 ciclos. Determinar el CPI medio considerando la memoria.

$CPI_{total} = CPI + MSI$

MSI: (average) Memory Stalls per Instruction

MSI = MAI x MissRate x MissPenalty

Tema 3. Ejercicios

- La frecuencia de una CPU es 1 GHz y tiene 1 nivel de cache. El CPI medio de ejecución de instrucciones es 1,1 y hay 50% aritméticas/lógicas, 30% cargar/guardar en la memoria y 20% saltos. El Hit Time es de 1 ciclo, Miss Rate de 1.5% y Miss Penalty de 50 ciclos. Determinar el CPI medio considerando la memoria.

CPI_{total} = CPI + MSI

MSI: (average) Memory Stalls per Instruction

MSI = MAI x MissRate x MissPenalty

MAI: (average) Memory Access per Instruction

MAI = (Fetch instrucción + Probabilidad de acceso)

Tema 3. Ejercicios

- La frecuencia de una CPU es 1 GHz y tiene 1 nivel de cache. El CPI medio de ejecución de instrucciones es 1,1 y hay 50% aritméticas/lógicas, 30% cargar/guardar en la memoria y 20% saltos. El Hit Time es de 1 ciclo, Miss Rate de 1.5% y Miss Penalty de 50 ciclos. Determinar el CPI medio considerando la memoria.

CPI_{total} = CPI + MSI

MAI = Fetch instrucción + Probabilidad de acceso = 1 + 0,3 = 1,3

MSI = MAI x MissRate x MissPenalty = 1,3 x 0,015 x 50 = 0,975 ciclos/instrucción

CPI_{total} = CPI + MSI = 1,1 + 0,975 = 2,075 ciclos/instrucción

Tema 3. Ejercicios

- Mismo problema anterior pero la frecuencia de una CPU es 2 GHz. Determinar el CPI medio considerando la jerarquía de memoria y el speedup conseguido.

Tema 3. Ejercicios

- Mismo problema anterior pero la frecuencia de una CPU es 2 GHz. Determinar el CPI medio considerando la jerarquía de memoria y el speedup conseguido.

→ La memoria no se acelera ya que sigue siendo la misma. Antes había un MissPenalty de 50 ciclos pero a 1 GHz es decir

1 ciclo anterior era 1 / 1 GHz = 1 ns → 50 ciclos = 50 ns

1 ciclo ahora es 1 / 2 Ghz = 0,5 ns (la mitad) → 50 ns = 100 ciclos

→ Por lo tanto, MissPenalty con esta nueva frecuencia es de 100 ciclos

Tema 3. Ejercicios

- Mismo problema anterior pero la frecuencia de una CPU es 2 GHz. Determinar el CPI medio considerando la jerarquía de memoria y el speedup conseguido.

→ La memoria no se acelera ya que sigue siendo la misma. Antes había un MissPenalty de 50 ciclos pero a 1 GHz es decir

1 ciclo anterior era 1 / 1 GHz = 1 ns → 50 ciclos = 50 ns

1 ciclo ahora es 1 / 2 Ghz = 0,5 ns (la mitad) → 50 ns = 100 ciclos

→ Por lo tanto, MissPenalty con esta nueva frecuencia es de 100 ciclos

→ En cambio no afecta la cache ya que trabaja a la misma frecuencia que la CPU

Tema 3. Ejercicios

- Mismo problema anterior pero la frecuencia de una CPU es 2 GHz. Determinar el CPI medio considerando la jerarquía de memoria y el speedup conseguido.

→ La memoria no se acelera ya que sigue siendo la misma. Antes había un MissPenalty de 50 ciclos pero a 1 GHz es decir

1 ciclo anterior era 1 / 1 GHz = 1 ns → 50 ciclos = 50 ns

1 ciclo ahora es 1 / 2 Ghz = 0,5 ns (la mitad) → 50 ns = 100 ciclos

→ Por lo tanto, MissPenalty con esta nueva frecuencia es de 100 ciclos

→ En cambio no afecta la cache ya que trabaja a la misma frecuencia que la CPU

MSI = 1,3 x 1,5% x 100 ciclos = 1,3 x 0,015 x 100 = 1,95

CPI = 1,1 + 1,95 = 3,05 (pero ahora 1 ciclo es 0,5 ns, antes era 1 ns)

Tema 3. Ejercicios

- Mismo problema anterior pero la frecuencia de una CPU es 2 GHz. Determinar el CPI medio considerando la jerarquía de memoria y el speedup conseguido.

→ La memoria no se acelera ya que sigue siendo la misma. Antes había un MissPenalty de 50 ciclos pero a 1 GHz es decir

1 ciclo anterior era 1 / 1 GHz = 1 ns → 50 ciclos = 50 ns

1 ciclo ahora es 1 / 2 Ghz = 0,5 ns (la mitad) → 50 ns = 100 ciclos

→ Por lo tanto, MissPenalty con esta nueva frecuencia es de 100 ciclos

→ En cambio no afecta la cache ya que trabaja a la misma frecuencia que la CPU

MSI = 1,3 x 1,5% x 100 ciclos = 1,3 x 0,015 x 100 = 1,95

CPI = 1,1 + 1,95 = 3,05 (pero ahora 1 ciclo es 0,5 ns, antes era 1 ns)

Para igualar la duración de los ciclos

speedup = 2,075 / (3,05 x 0,5) = 1,36

Duplicando la frecuencia, el ordenador no es 2 veces más rápido, si no solo 1,36

Tema 3. Ejercicios

- La frecuencia de una CPU es 500 MHz y tiene hasta 3 niveles de cache. El CPI medio de ejecución de instrucciones es 1.1 y hay 50% aritméticas/lógicas, 30% cargar/guardar en la memoria y 20% saltos
- L1 cache a 500 MHz, Miss Rate de 5%, Hit Time de 1 ciclo (va a 500 MHz)
- L2 cache a 250 MHz, Miss Rate de 40%, Hit Time de 2 ciclos (va a la mitad de 500 MHz)
- L3 cache a 100 MHz, Miss Rate de 50%, Hit Time de 5 ciclos (va a 1/5 de 500 MHz)
- Miss Penalty de 100 ciclos.
- Determinar el CPI medio considerando sin cache, con solo L1, con L1 y L2 y con todas.

Tema 3. Ejercicios

- La frecuencia de una CPU es 500 MHz y tiene hasta 3 niveles de cache. El CPI medio de ejecución de instrucciones es 1.1 y hay 50% aritméticas/lógicas, 30% cargar/guardar en la memoria y 20% saltos
- L1 cache a 500 MHz, Miss Rate de 5%, Hit Time de 1 ciclo
- L2 cache a 250 MHz, Miss Rate de 40%, Hit Time de 2 ciclos
- L3 cache a 100 MHz, Miss Rate de 50%, Hit Time de 5 ciclos
- Miss Penalty de 100 ciclos.
- Determinar el CPI medio considerando sin cache, con solo L1, con L1 y L2 y con todas.

Sin cache

MAI = (Fetch + Probabilidad de acceso) = 1 + 0,3 = 1,3

MSI = MAI x MissRate x MissPenalty = 1,3 x 1 x 100 = 130

CPI = 1,1 + 130 = 131,1

Tema 3. Ejercicios

- La frecuencia de una CPU es 500 MHz y tiene hasta 3 niveles de cache. El CPI medio de ejecución de instrucciones es 1.1 y hay 50% aritméticas/lógicas, 30% cargar/guardar en la memoria y 20% saltos
- L1 cache a 500 MHz, Miss Rate de 5%, Hit Time de 1 ciclo
- L2 cache a 250 MHz, Miss Rate de 40%, Hit Time de 2 ciclos
- L3 cache a 100 MHz, Miss Rate de 50%, Hit Time de 5 ciclos
- Miss Penalty de 100 ciclos.
- Determinar el CPI medio considerando sin cache, con solo L1, con L1 y L2 y con todas.

Con cache L1

MAI = (Fetch + Probabilidad de acceso) = 1 + 0,3 = 1,3

MSI = MAI x MissRate x MissPenalty = 1,3 x 0,05 x 100 = 6,5

CPI = 1,1 + 6,5 = 7,6

Tema 3. Ejercicios

- La frecuencia de una CPU es 500 MHz y tiene hasta 3 niveles de cache. El CPI medio de ejecución de instrucciones es 1.1 y hay 50% aritméticas/lógicas, 30% cargar/guardar en la memoria y 20% saltos
- L1 cache a 500 MHz, Miss Rate de 5%, Hit Time de 1 ciclo
- L2 cache a 250 MHz, Miss Rate de 40%, Hit Time de 2 ciclos
- L3 cache a 100 MHz, Miss Rate de 50%, Hit Time de 5 ciclos
- Miss Penalty de 100 ciclos.
- Determinar el CPI medio considerando sin cache, con solo L1, con L1 y L2 y con todas.

Con cache L1 y L2

MAI = (Fetch + Probabilidad de acceso) = 1 + 0,3 = 1,3

$MSI = MAI \times MissRate_{L1} \times (HitRate_{L2} \times HitTime_{L2} + MissRate_{L2} \times MissPenalty) =$
$= 1,3 \times 0,05 \times (0,6 \times 2 + 0,4 \times 100) = 2,678$

CPI = 1,1 + 2,678 = 3,778

Tema 3. Ejercicios

- La frecuencia de una CPU es 500 MHz y tiene hasta 3 niveles de cache. El CPI medio de ejecución de instrucciones es 1.1 y hay 50% aritméticas/lógicas, 30% cargar/guardar en la memoria y 20% saltos
- L1 cache a 500 MHz, Miss Rate de 5%, Hit Time de 1 ciclo
- L2 cache a 250 MHz, Miss Rate de 40%, Hit Time de 2 ciclos
- L3 cache a 100 MHz, Miss Rate de 50%, Hit Time de 5 ciclos
- Miss Penalty de 100 ciclos.
- Determinar el CPI medio considerando sin cache, con solo L1, con L1 y L2 y con todas.

Con cache L1, L2 y L3

MAI = (Fetch + Probabilidad de acceso) = 1 + 0,3 = 1,3

$$MSI = MAI \times MissRate_{L1} \times (HitRate_{L2} \times HitTime_{L2} + MissRate_{L2} \times (HitRate_{L3} \times HitTime_{L3} + MissRate_{L3} \times MissPenalty)) =$$
$$= 1,3 \times 0,05 \times (0,6 \times 2 + 0,4 \times (0,5 \times 5 + 0,5 \times 100) = 1,443$$

CPI = 1,1 + 1,443 = 2,543

Tema 3. Ejercicios

- La frecuencia de una CPU es 500 MHz y tiene hasta 3 niveles de cache. El CPI medio de ejecución de instrucciones es 1.1 y hay 50% aritméticas/lógicas, 30% cargar/guardar en la memoria y 20% saltos
- L1 cache a 500 MHz, Miss Rate de 5%, Hit Time de 1 ciclo
- L2 cache a 250 MHz, Miss Rate de 40%, Hit Time de 2 ciclos
- L3 cache a 100 MHz, Miss Rate de 50%, Hit Time de 5 ciclos
- Miss Penalty de 100 ciclos.
- Determinar el CPI medio considerando sin cache, con solo L1, con L1 y L2 y con todas.

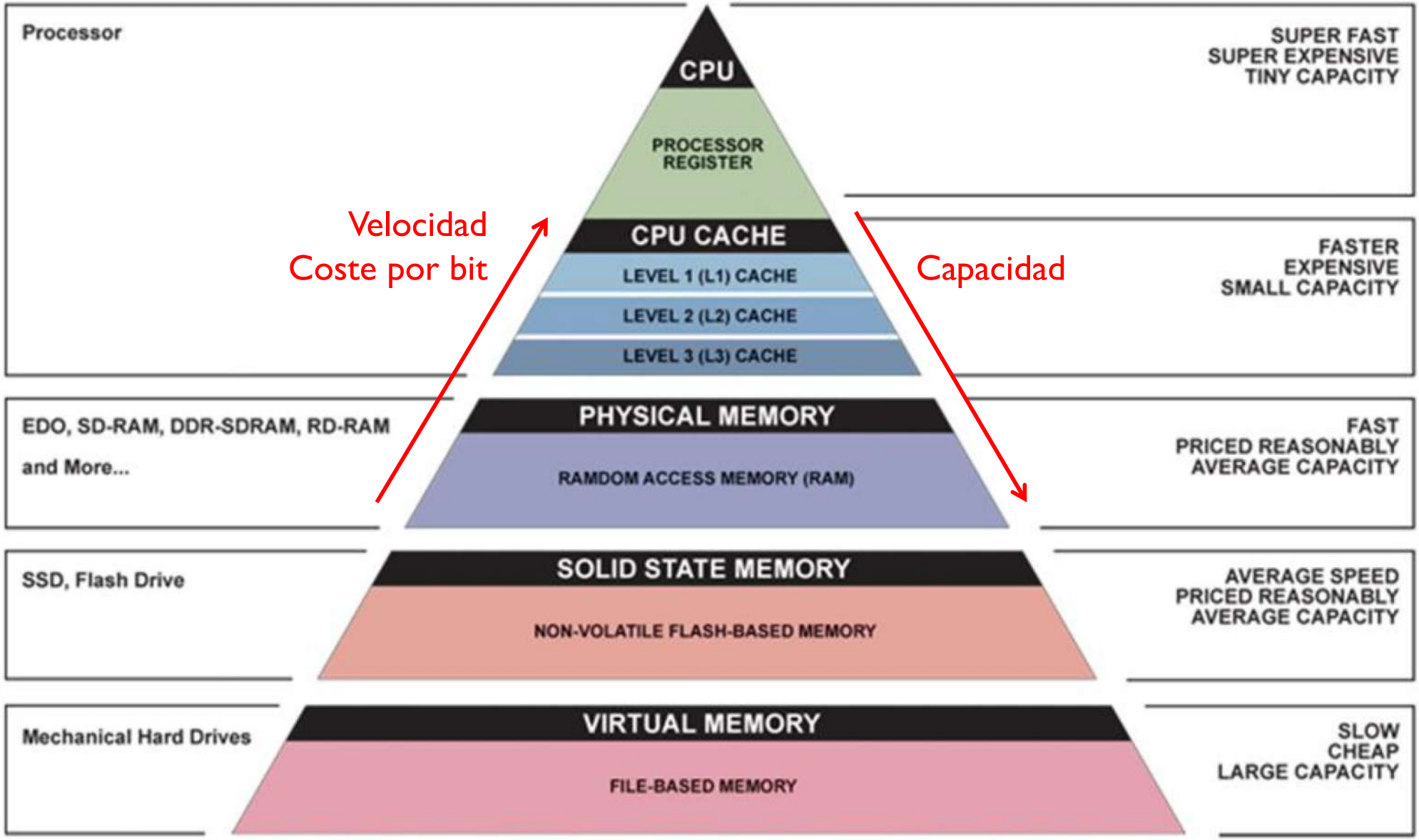
CPI = 131,1

CPI_1 = 7,6 $S_{no->1}$ = 131,1 / 7,6 = 17,25

CPI_2 = 3,778 $S_{1->2}$ = 7,6 / 3,778 = 2,01 $S_{no->2}$ = 34,7

CPI_3 = 2,543 $S_{2->3}$ = 3,778 / 2,543 = 1,49 $S_{1->3}$ = 3 $S_{no->3}$ = 51,55

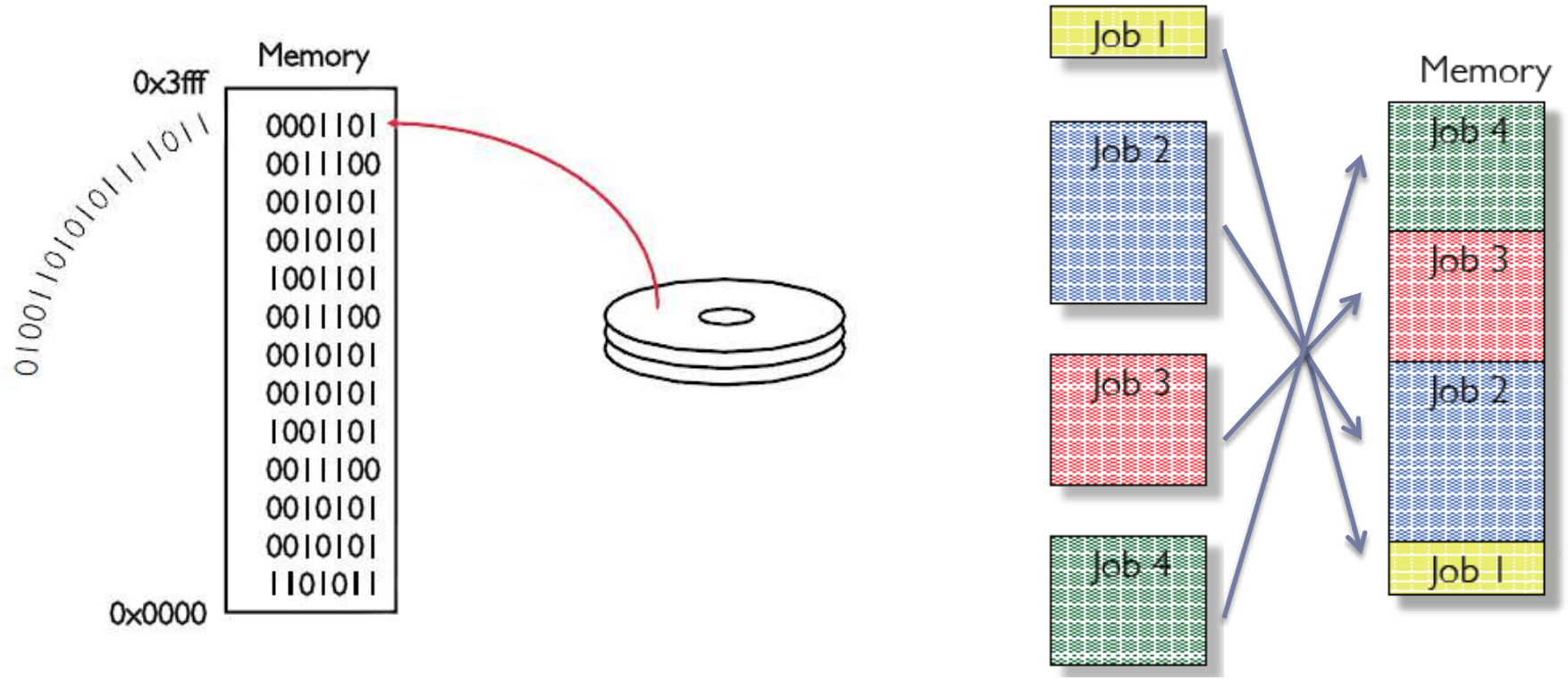
Tema 3. Jerarquía de memorias



▲ Simplified Computer Memory Hierarchy
Illustration: Ryan J. Leng

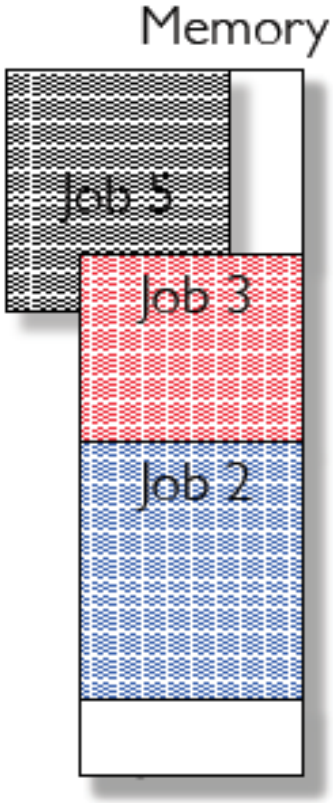
Tema 3. Memorias virtuales

- Cuando se ejecuta un programa, se copian las instrucciones y los datos de un disco a la memoria
- El Program Counter (o Instruction Pointer o …) apunta luego a la primera dirección de memoria donde empieza el programa para su ejecución
- Y la memoria es un espacio compartido entre varios programas



Tema 3. Memorias virtuales

- Y la memoria, siendo espacio compartido y limitado, puede que no sea suficiente para mantener todos los programas activos en la memoria
- Por ejemplo el programa 5 no cabe en la memoria
- Por eso principalmente, pero también por otra razón, existe la memoria virtual



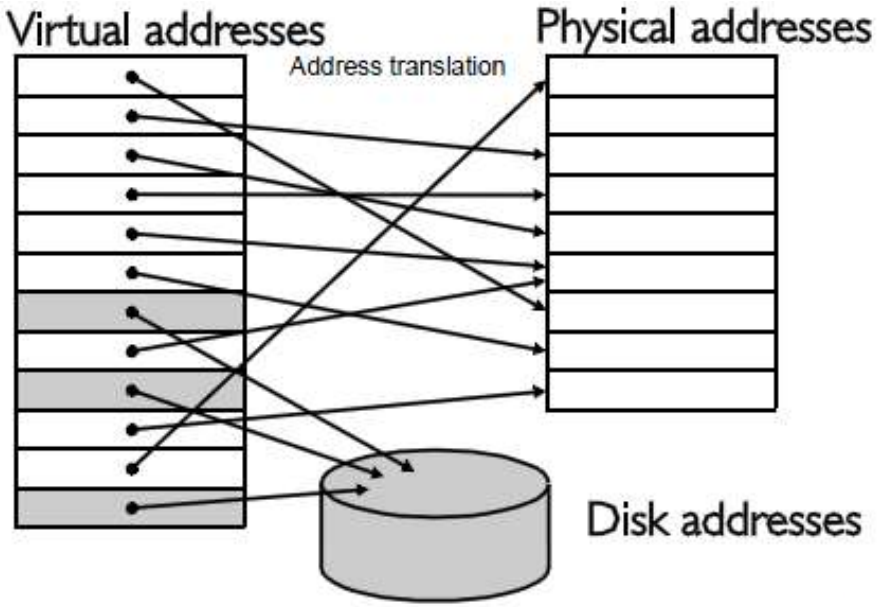
Tema 3. Memorias virtuales

- ¿Que es la memoria virual?
- Tecnica que permite la ejecución de un programa que
  - Puede estar guardado en la memoria en celdas no contiguas
  - Puede no estar completamente guardado en la memoria
- Permite a un ordenador hacer creer a un programa que
  - Sus instruccones y datos residen en celdas contiguas en la memoria
  - La memoria es más grande de lo que es realmente

- ¿por que es importante la memoria virtual?
  - Se consigue espacio de memoria mas barata
  - Los programas actuales pueden ser más grandes
  - Gestión automatica entre memoria y disco
  - Permite un arranque más rapido desde el disco

Tema 3. Memorias virtuales

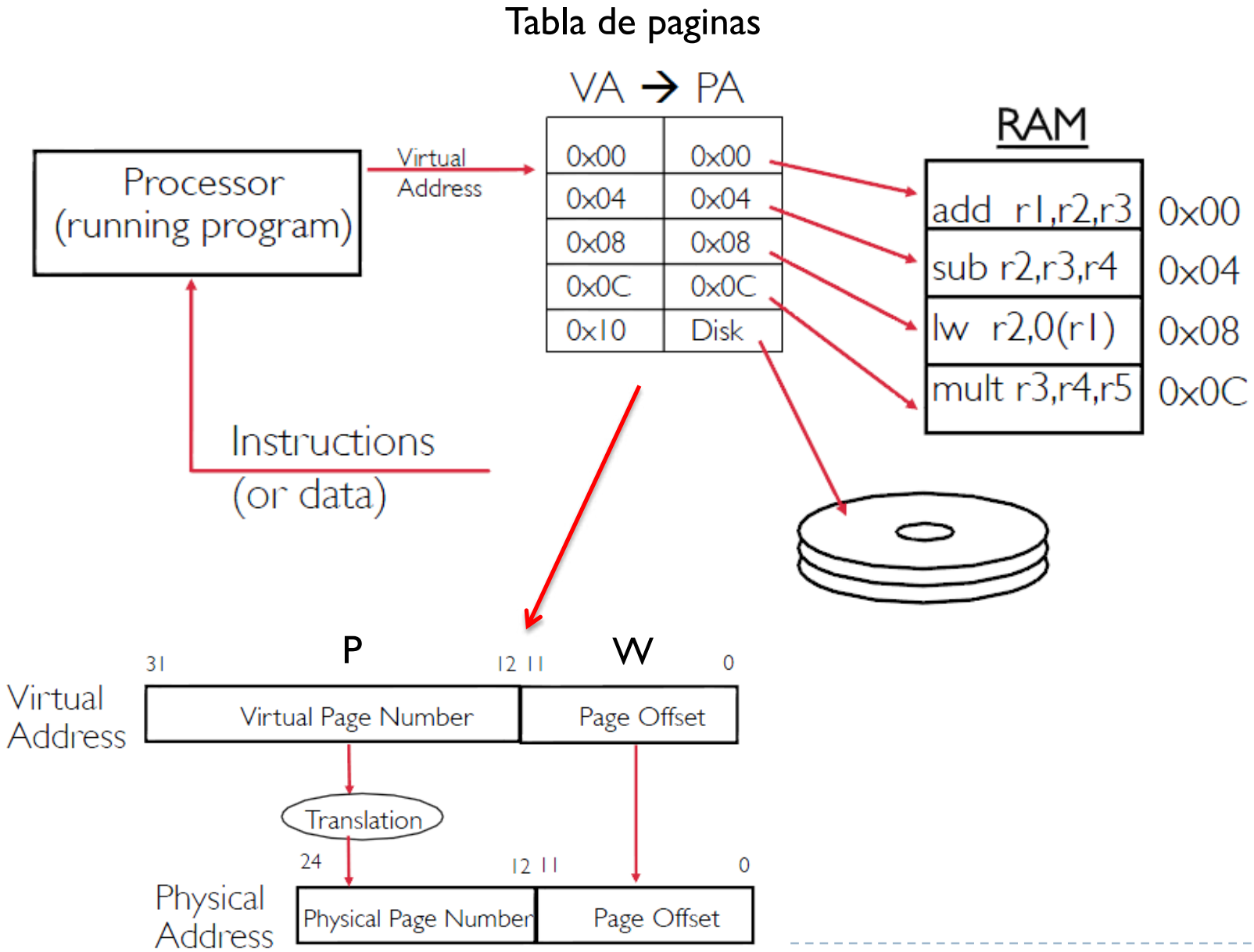
- El espacio de direcciones virtual está mapeado de tal forma que una pequeña parte de él está en memoria real y el resto almacenado en el disco
- Cuando se quiere acceder a una dirección que se halla en el disco duro, hay que realizar un intercambio (swap) entre la información de la memoria física y del disco
- La memoria actúa como una caché del disco
  - La memoria está dividida en paginas (como la caché en bloques)
  - Una pagina puede residir en el disco o en la memoria y pasar de uno a otro según algún criterio



Tema 3. Memorias virtuales

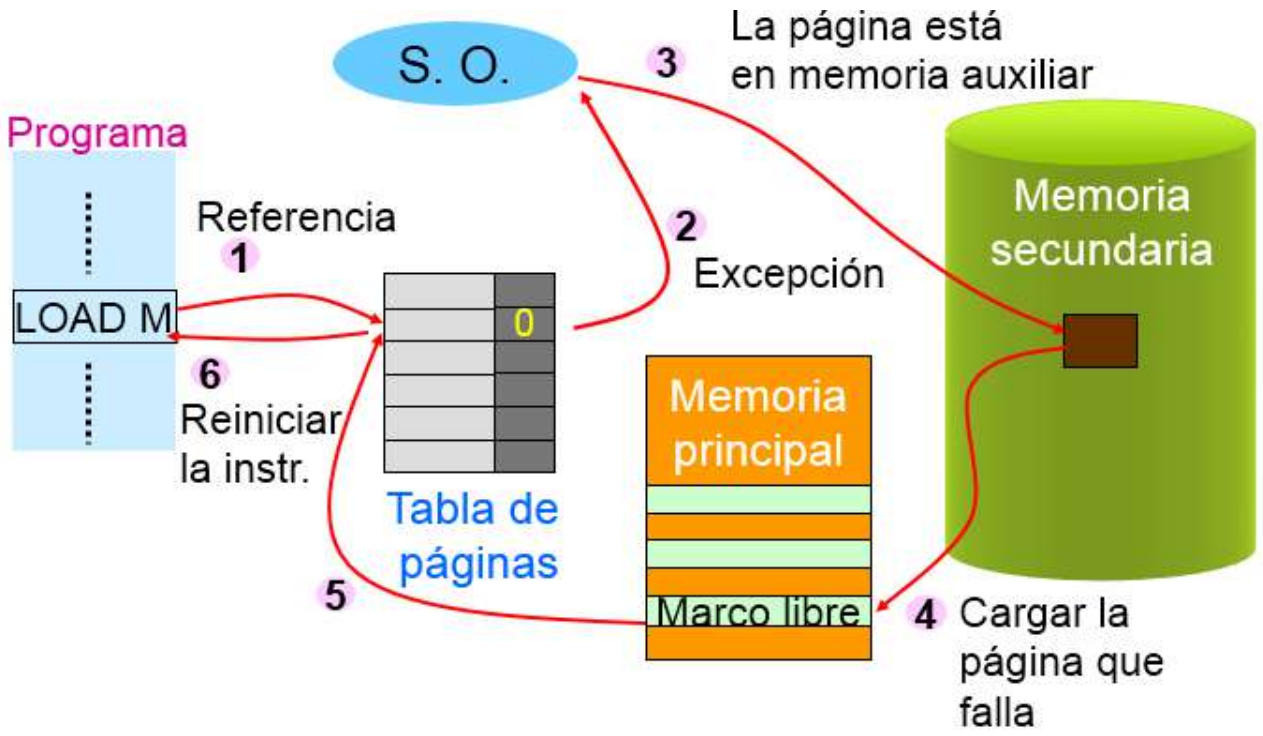
- La memoria principal se divide en un conjunto de marcos de igual tamaño
- Cada proceso se divide en una serie de páginas del tamaño de los marcos
- Un proceso se carga en los marcos que requiera (todas las páginas), no necesariamente contiguos
- El SO mantiene una tabla de paginas para cada proceso, que contiene la lista de marcos para cada pagina
- No es necesario cargar todas las páginas
- Las páginas no residentes se cargan por demanda
- El hardware de gestión de memoria (MMU, Memory Management Unit) usa la tabla de páginas para traducir todas las direcciones que genera un programa.
  - La MMU incluye internamente una especie de cache de traducciones llamada TLB (Translation Lookaside Buffer)
  - Se trata de una pequeña memoria interna a la MMU que mantiene información sobre las últimas páginas accedidas.
- Una dirección de memoria es un número de página (P) y un desplazamiento dentro de la página (W)

Tema 3. Memorias virtuales



Tema 3. Memorias virtuales

- Fallo de página: Ocurre cuando se referencia a una dirección virtual y no reside en la memoria real → acceso a disco duro
- Como respuesta al fallo de página, el SO:
  - Selecciona una página poco usada del proceso
  - Intercambia la página a disco
  - Asigna el marco de la página liberada a la página virtual que se intenta acceder
  - Carga la página que se necesita en ese marco
  - Cambia la tabla de traducción de la MMU
  - Comienza de nuevo esa instrucción interrumpida



Arquitectura i Configuracions Informàtiques

Tema 3. La memoria

Davide Careglio