

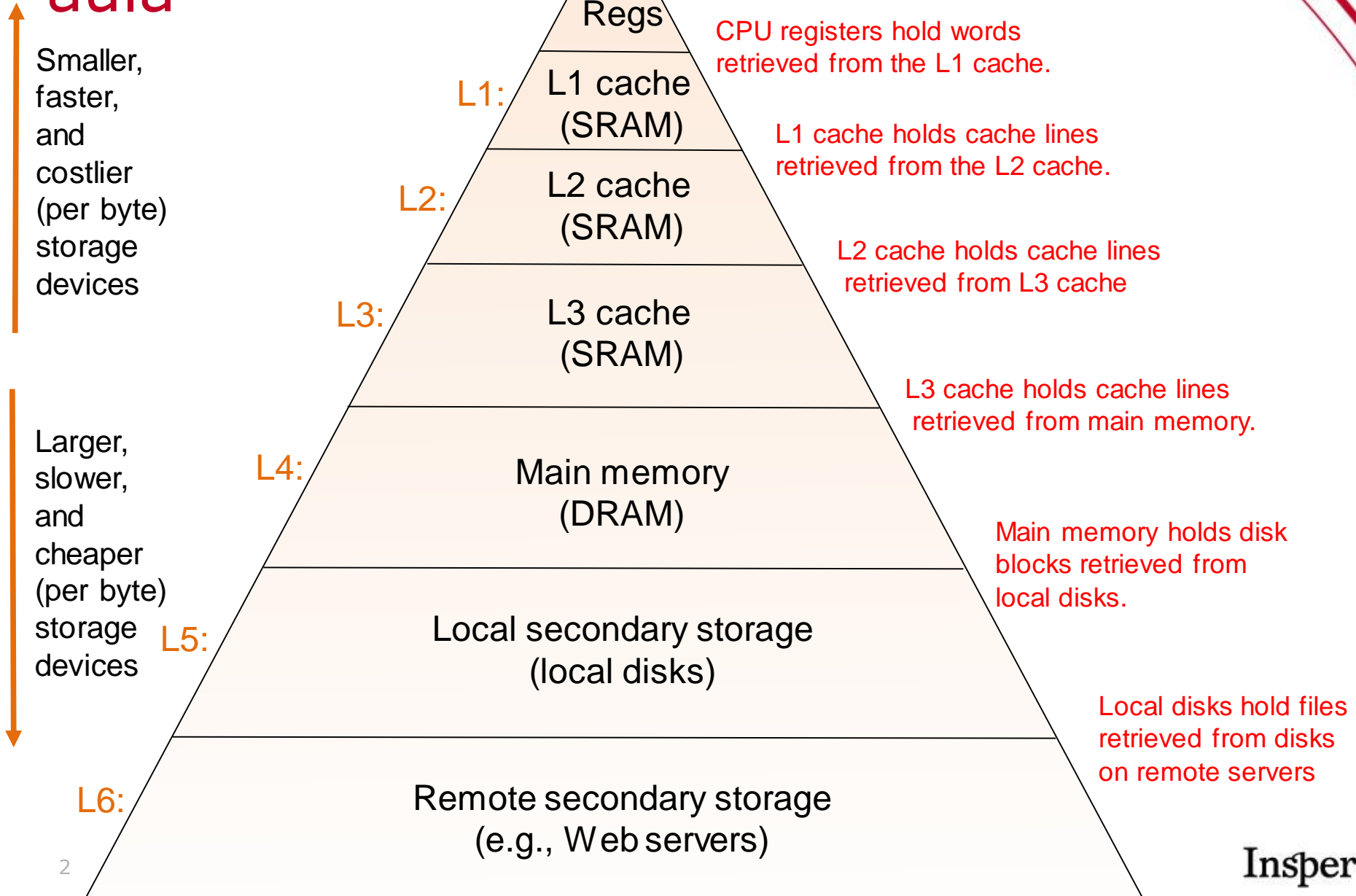
Sistemas Hardware-Software

Aula 14 – Hierarquia de memória

2019 – Engenharia

Igor Montagner, Fábio Ayres <igorsm1@insper.edu.br>

Resumo da aula



Agenda

Hierarquia de memória

Tecnologias de memória e armazenamento

Memória Cache

Grande ideia da aula de hoje

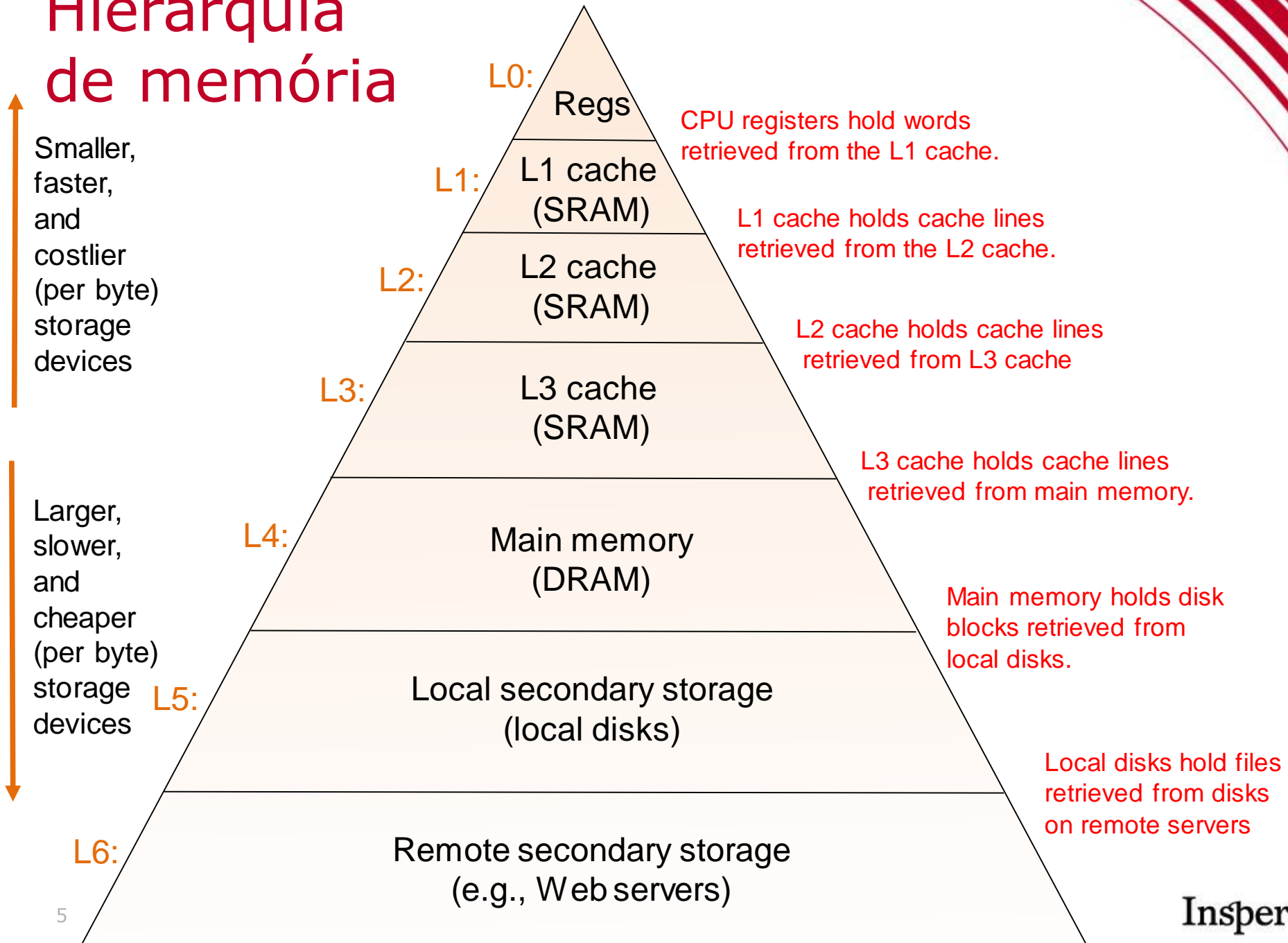
Um sistema possui vários tipos de memórias com diferentes velocidades. Armazenamos

- Em memórias **mais rápidas** os dados **mais frequentemente** usados
- Em memórias **mais lentas** os dados **menos frequentemente** usados

Levando em conta

- Temporalidade: um dado é acessado repetidamente
- Localidade: dados são acessados de maneira conjunta

Hierarquia de memória



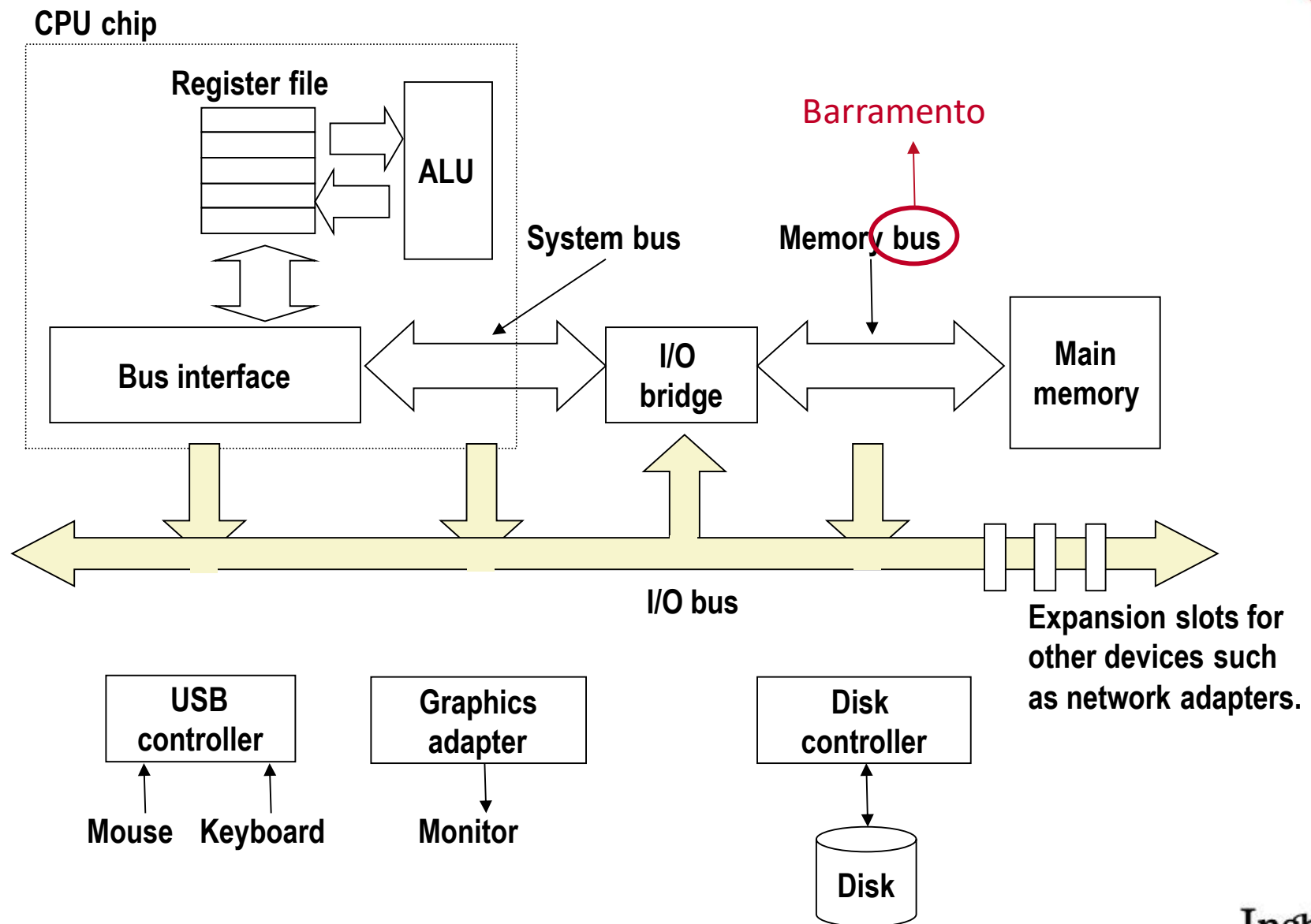
Agenda

Hierarquia de memória

Tecnologias de memória e armazenamento

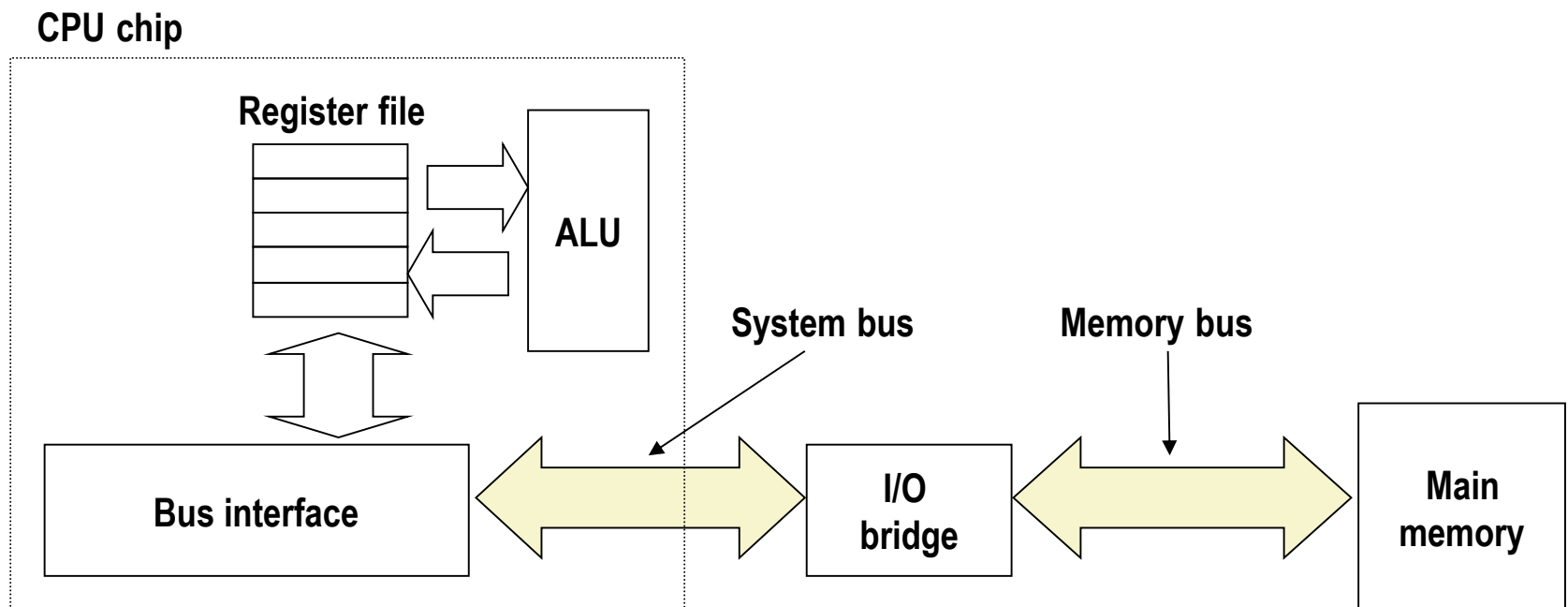
Memória Cache

Interface entre CPU, memória e disco



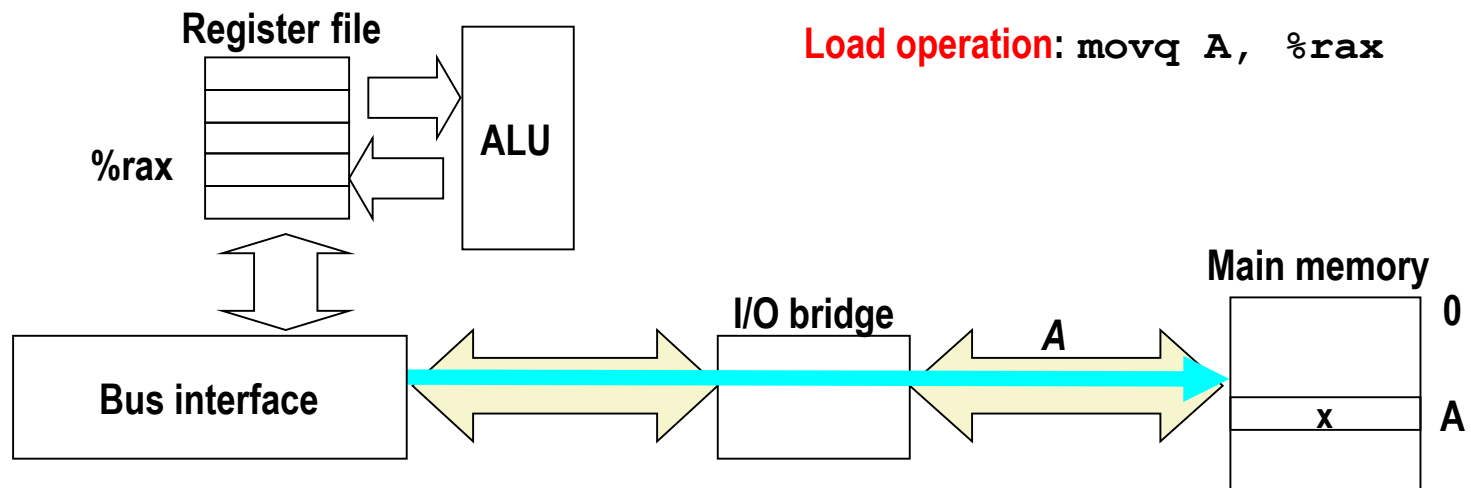
Conectando a CPU e a RAM

Uma *bus* é uma coleção de fios paralelos que carregam endereços, dados, e sinais de controle. *Buses* são normalmente compartilhadas entre vários dispositivos.



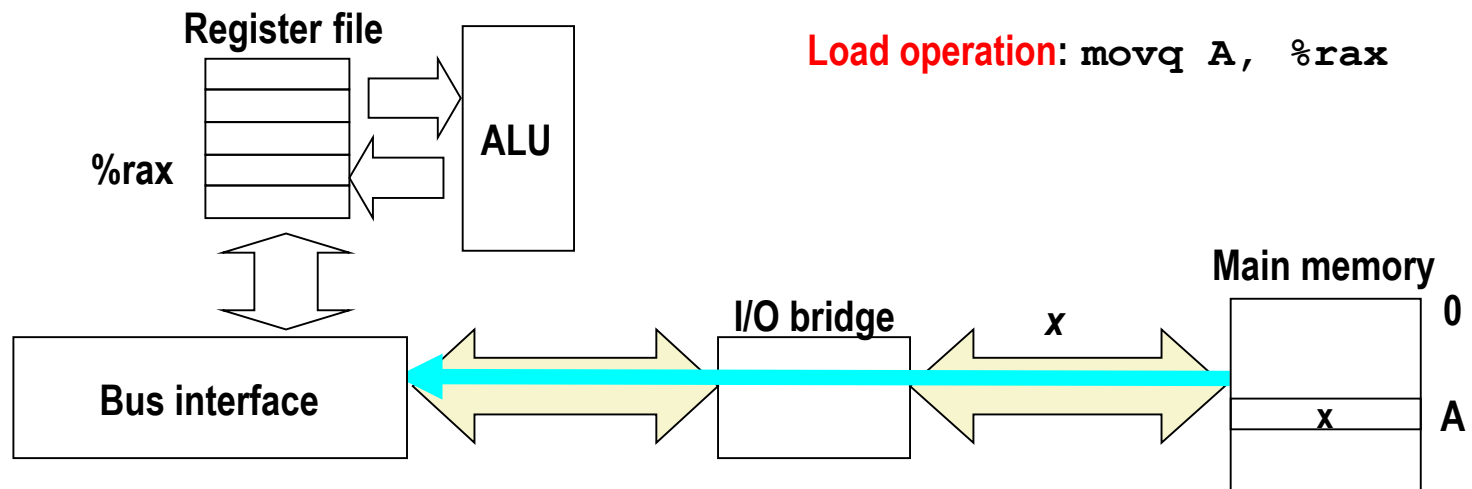
Leitura de dados

CPU coloca endereço **A** na bus de memória



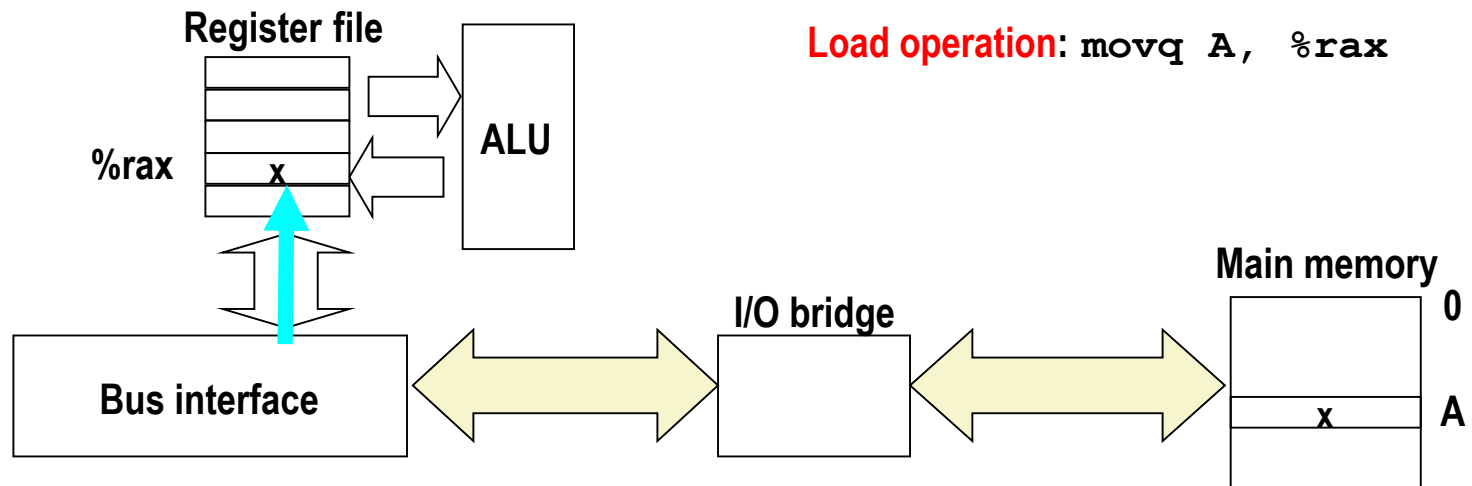
Leitura de dados (II)

Memória principal acessa o endereço **A**, recuperando o valor **x**, e coloca este valor na bus



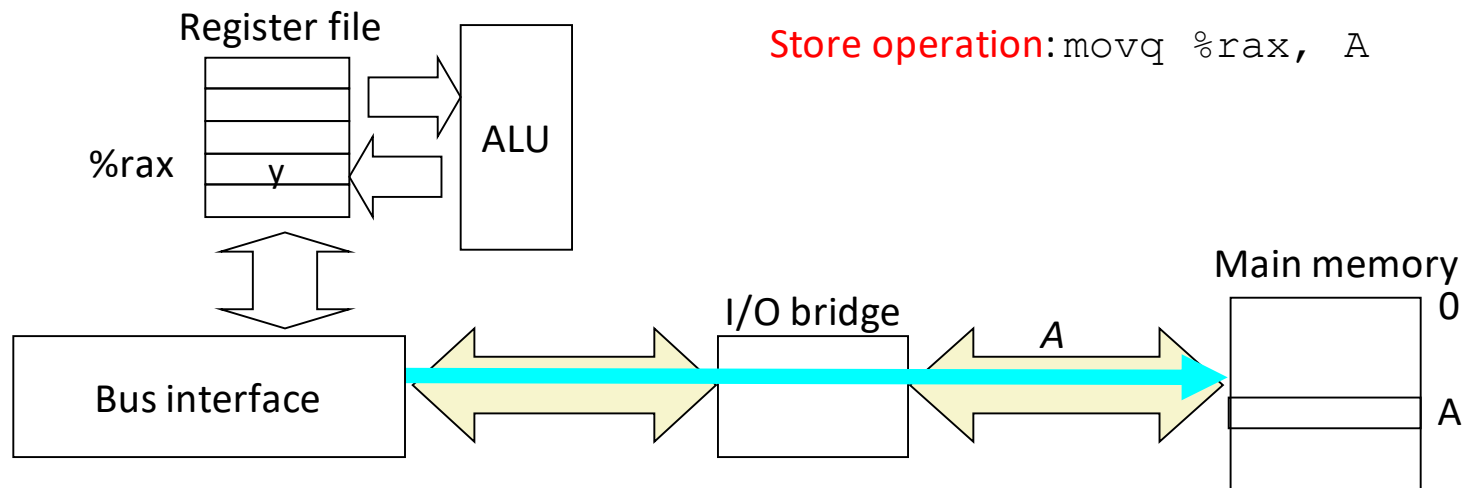
Leitura de dados (III)

CPU lê o valor **x** da bus e coloca no registrador **%rax**



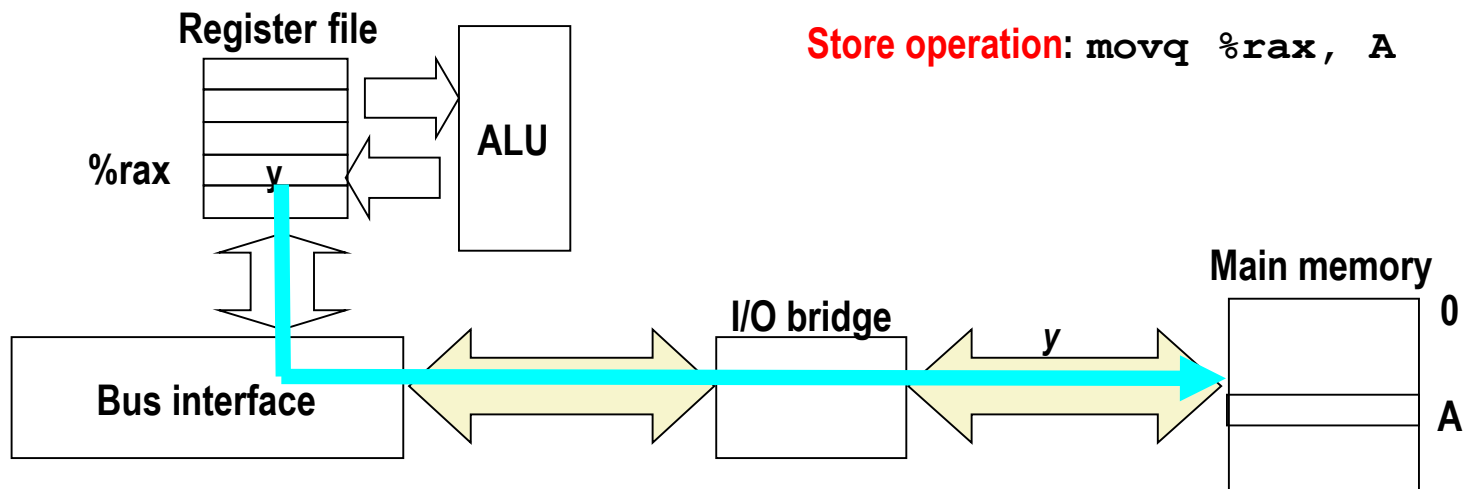
Escrita de dados

CPU escreve endereço **A** na bus. Memória principal guarda **A** em um buffer local e aguarda a palavra de dados correspondente.



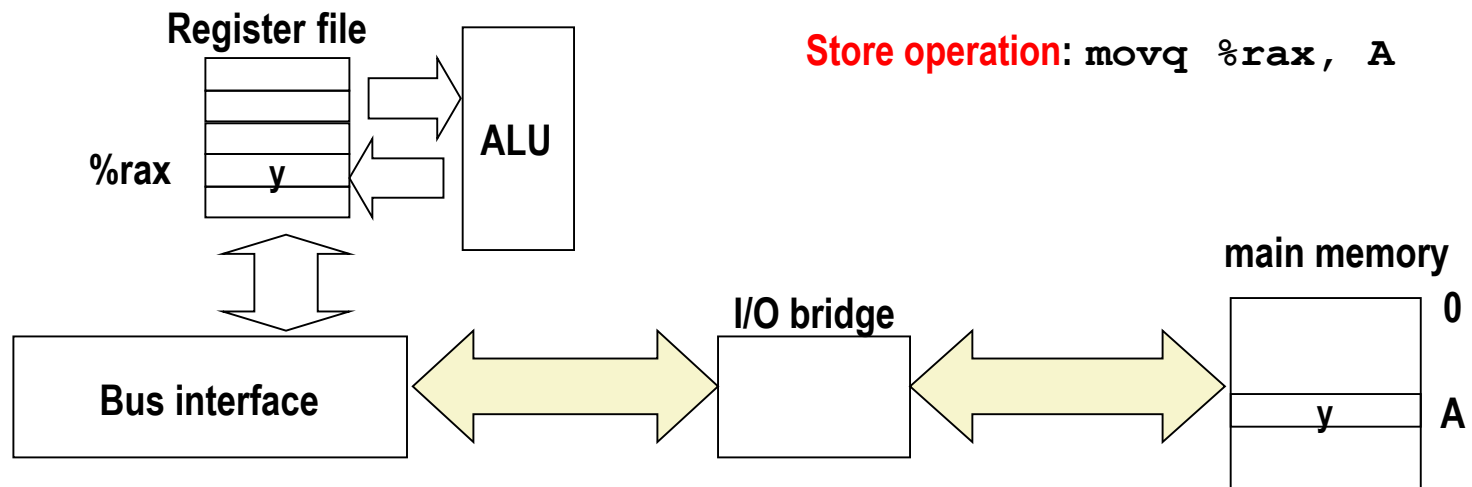
Escrita de dados (II)

CPU coloca dado **y** na bus

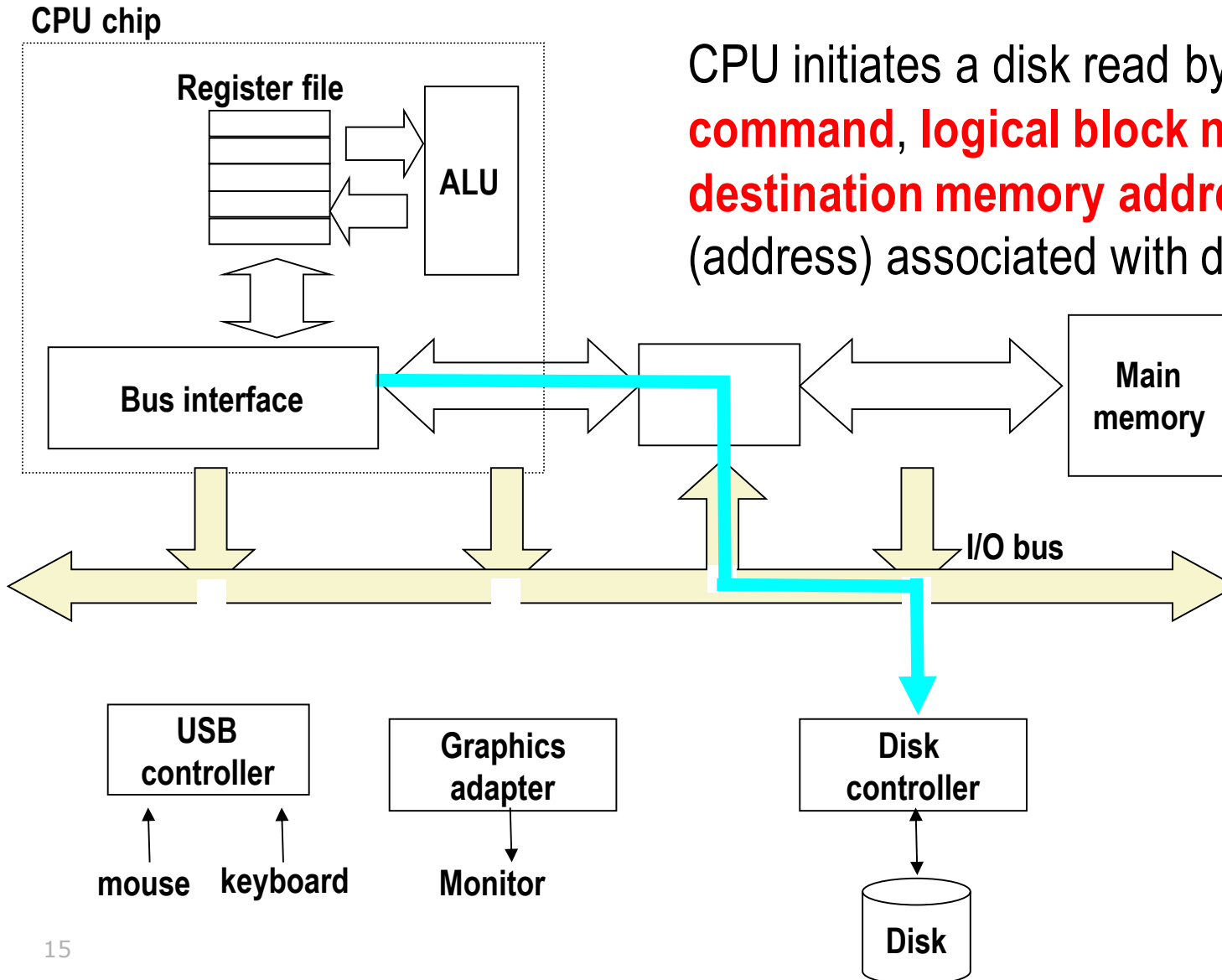


Escrita de dados (III)

Memória principal lê **y** da bus e guarda no endereço **A**

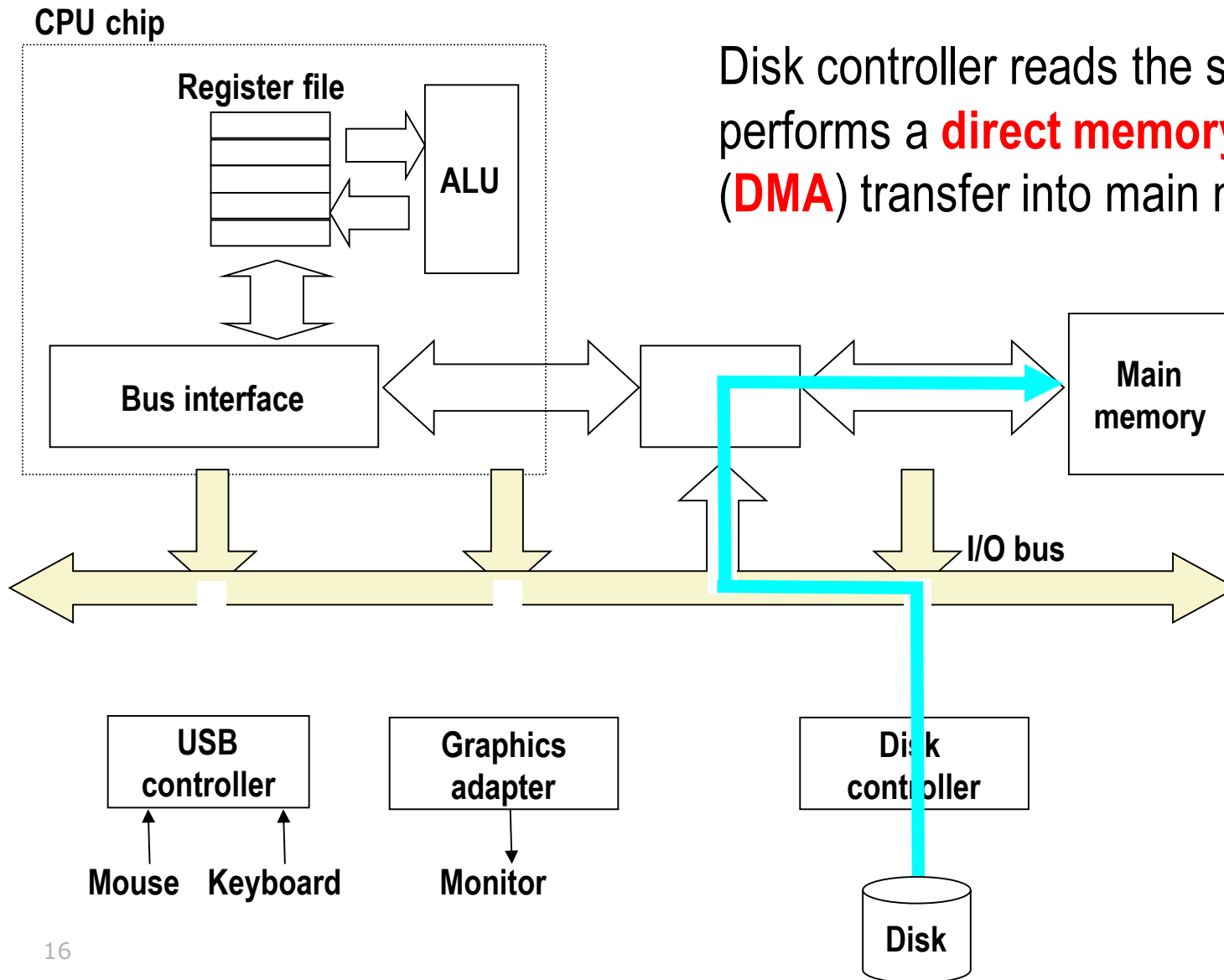


Lendo dados do disco



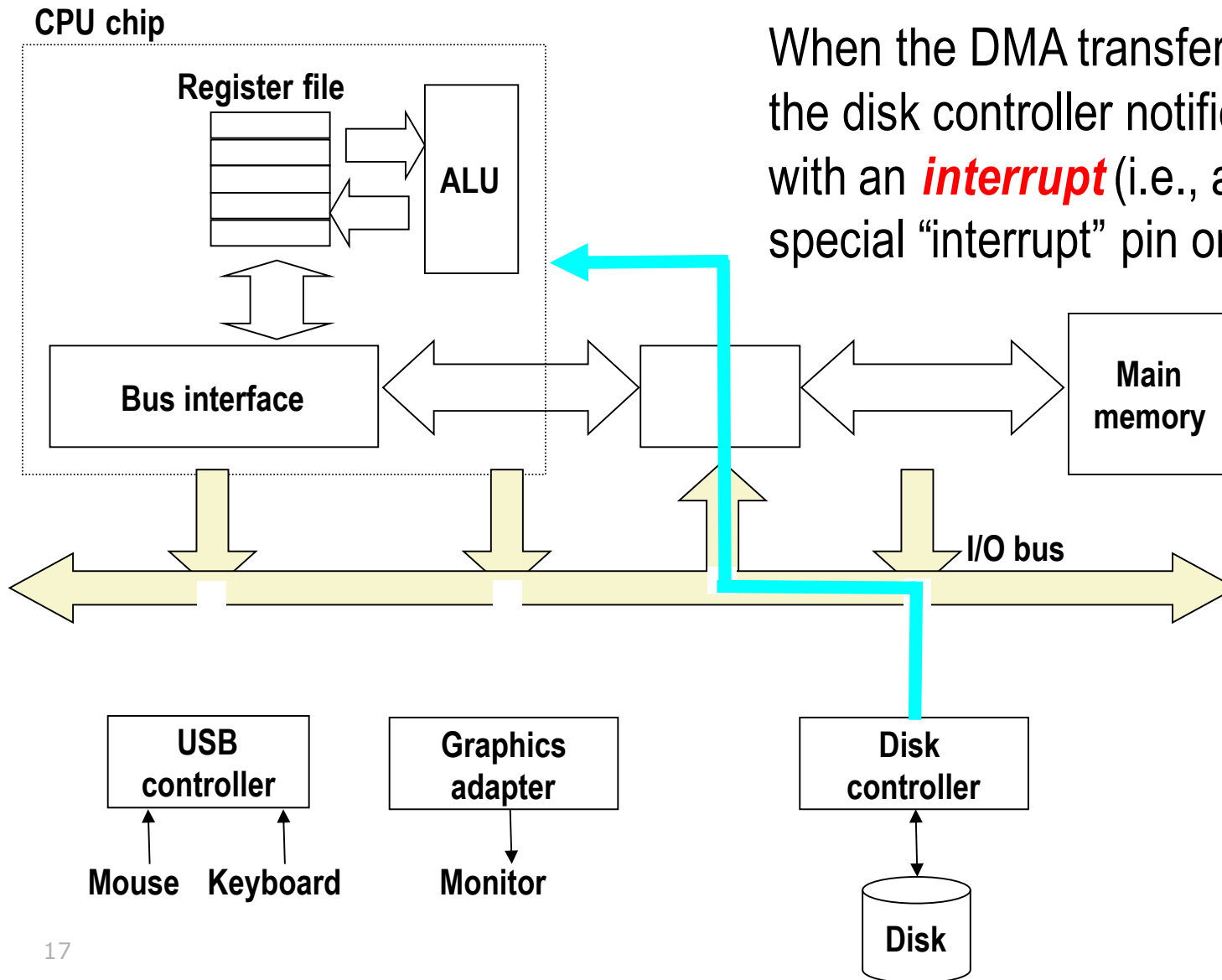
CPU initiates a disk read by writing a **command**, **logical block number**, and **destination memory address** to a **port** (address) associated with disk controller.

Lendo dados do disco (II)



Disk controller reads the sector and performs a **direct memory access (DMA)** transfer into main memory.

Lendo dados do disco (III)



When the DMA transfer completes, the disk controller notifies the CPU with an **interrupt** (i.e., asserts a special “interrupt” pin on the CPU)

Agenda

Tecnologia de memória e armazenamento

Hierarquia de memória

Memória Cache

Armazenamento de dados

1. Static RAM
2. Dynamic RAM
3. Read Only Memory – (SSDs)
4. Disco rígido mecânico

Random Access Memory (RAM)

- Static RAM (SRAM)
 - Feita de células bi-estáveis (e.g. flip-flops)
 - Rápida
 - Cara
 - Usos: memória cache
- Dynamic RAM (DRAM)
 - Pequenos capacitores
 - Barata
 - Mais lenta
 - Usos: memória principal, memória de vídeo

SRAM versus DRAM

	Trans. per bit	Access time	Needs refresh?	Needs EDC?	Cost	Applications
SRAM	4 or 6	1X	No	Maybe	100x	Cache memories
DRAM	1	10X	Yes	Yes	1X	Main memories, frame buffers

Read-Only Memory (ROM)

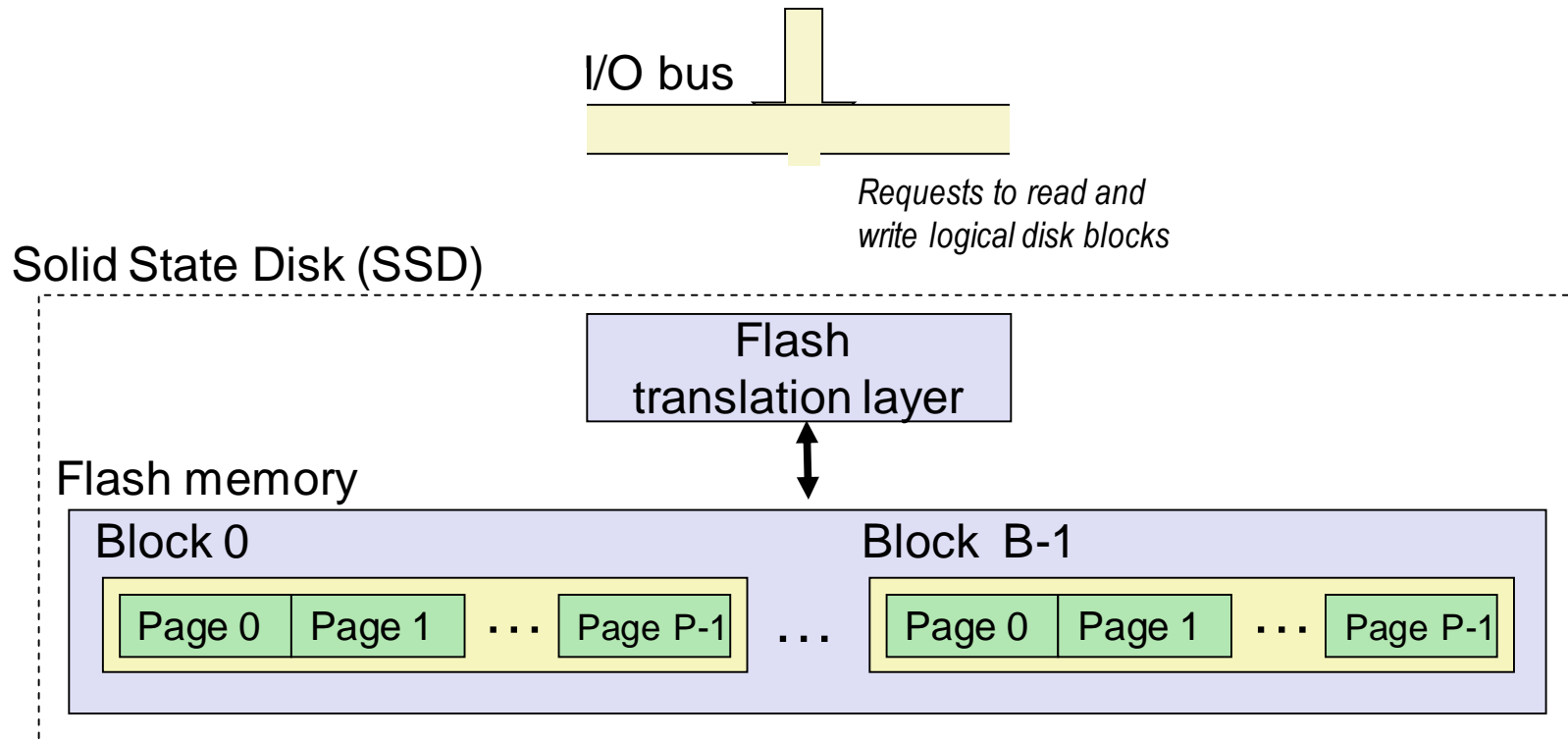
- SRAM e DRAM são voláteis: os dados são perdidos com o desligamento da máquina.
- Memórias não voláteis mantêm o armazenamento mesmo com a máquina desligada
 - Read-only memory (ROM): programada em produção
 - Programmable ROM (PROM): Pode ser programada apenas uma vez
 - Erasable PROM (EPROM): Pode ser apagada com equipamento especial (UV, X-Ray)
 - Electrically erasable PROM (EEPROM): pode ser apagada eletronicamente
 - Memória Flash: EEPROMs com capacidade de apagamento parcial (por blocos)
 - “Gasta” depois de cerca de 100 000 apagamentos

Usos da ROM

- Firmware
 - BIOS
 - Controladoras de disco, placas de rede, GPUs, etc
- Solid-state disks (SSDs)
- Flash memory

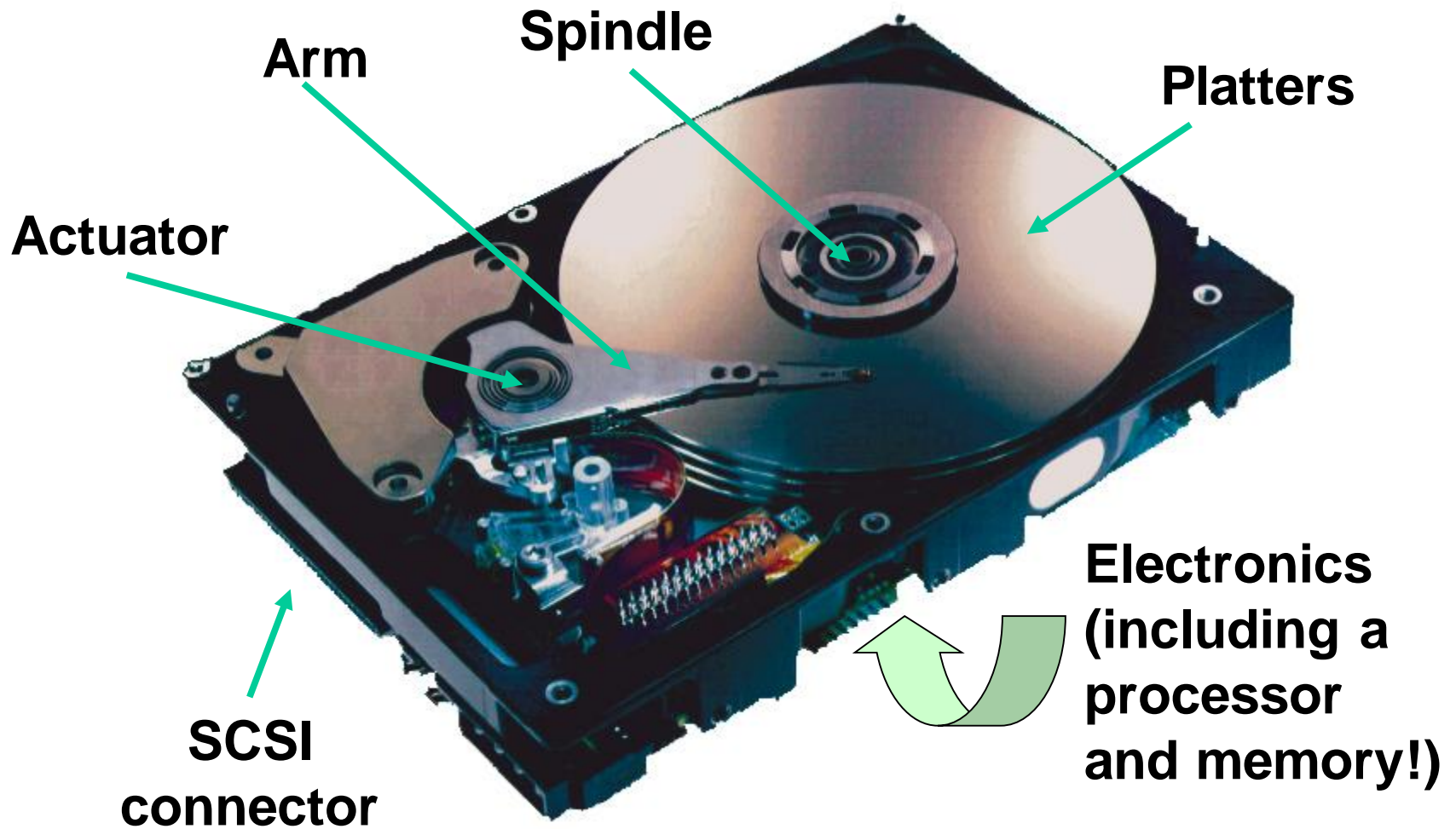


Solid State Disks (SSD)



- Páginas (*pages*): 4KB a 512KB. Blocos (*blocks*): 32 a 128 páginas
- Dados são lidos e escritos em unidades de páginas
- Página só pode ser escrita quando o bloco é apagado
- Um bloco desgasta depois de cerca de 100 000 escritas

Discos



Performance SSD vs HDD

Intel SSD 730 (480Gb)

Sequential read tput	517 MB/s	Sequential write tput	476 MB/s
Random (4k) read tput	32 MB/s	Random (4k) write tput	105 MB/s

Source: <http://ssd.userbenchmark.com/Intel-730-Series-480GB/Rating/3479>

Barracuda 1Tb, 7200rpm

Sequential read tput	160 MB/s	Sequential write tput	150 MB/s
Random (4k) read tput	0.66 MB/s	Random (4k) write tput	1.56 MB/s

Source: <http://hdd.userbenchmark.com/SpeedTest/2870/HGST-HTS545050A7E680>

Performance SSD vs HDD

Intel SSD 730 (480Gb)

Sequential read tput	517 MB/s	Sequential write tput	476 MB/s
Random (4k) read tput	32 MB/s	Random (4k) write tput	105 MB/s

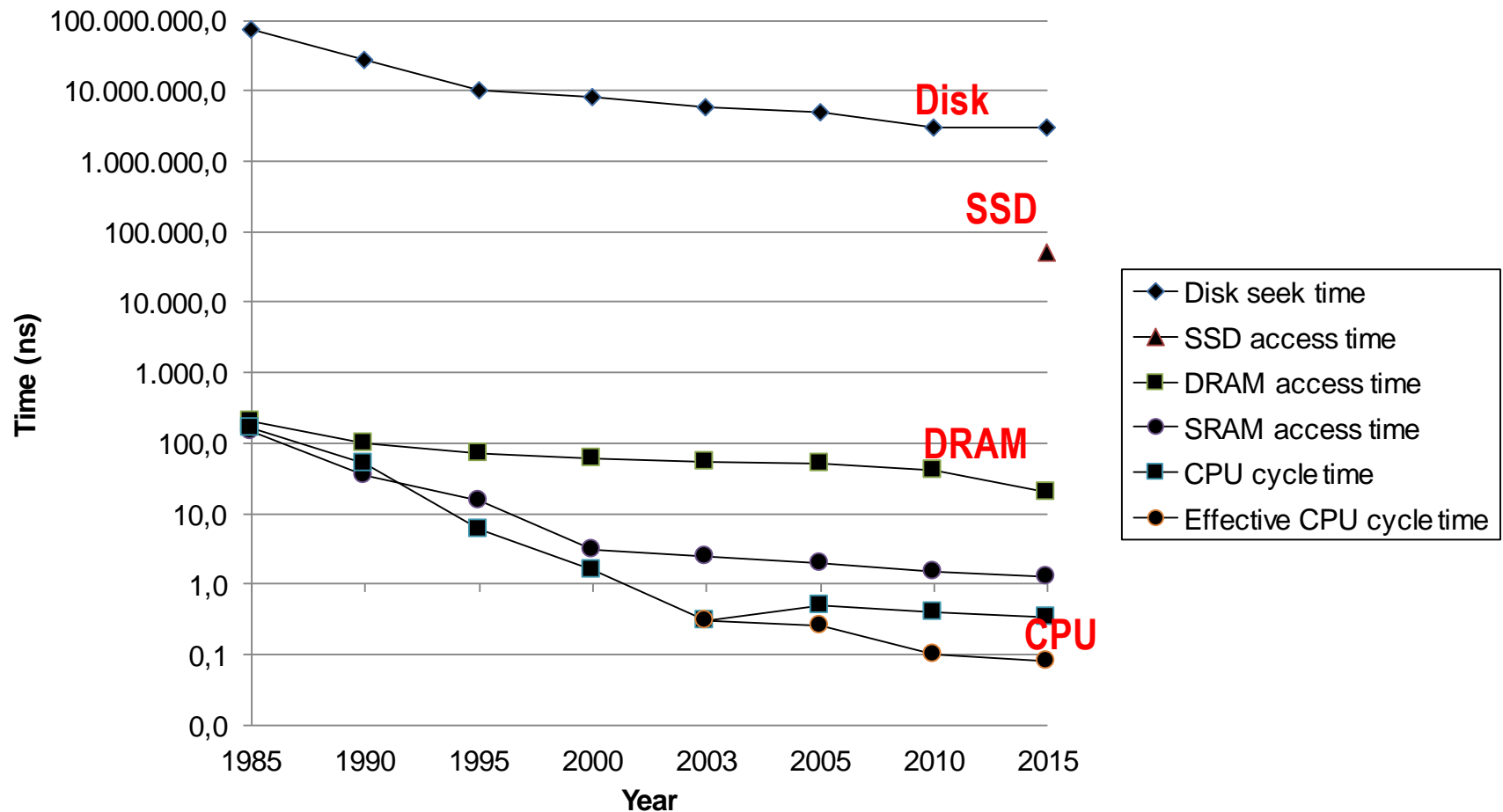
Source: <http://ssd.userbenchmark.com/Intel-730-Series-480GB/Rating/3479>

Meu HDD (Hitachi, 5400rpm, 500Gb)

Sequential read tput	93.9 MB/s	Sequential write tput	82.4 MB/s
Random (4k) read tput	0.21 MB/s	Random (4k) write tput	1.15 MB/s

Source: <http://hdd.userbenchmark.com/SpeedTest/2870/HGST-HTS545050A7E680>

Comparação entre a evolução da CPU, memória e discos

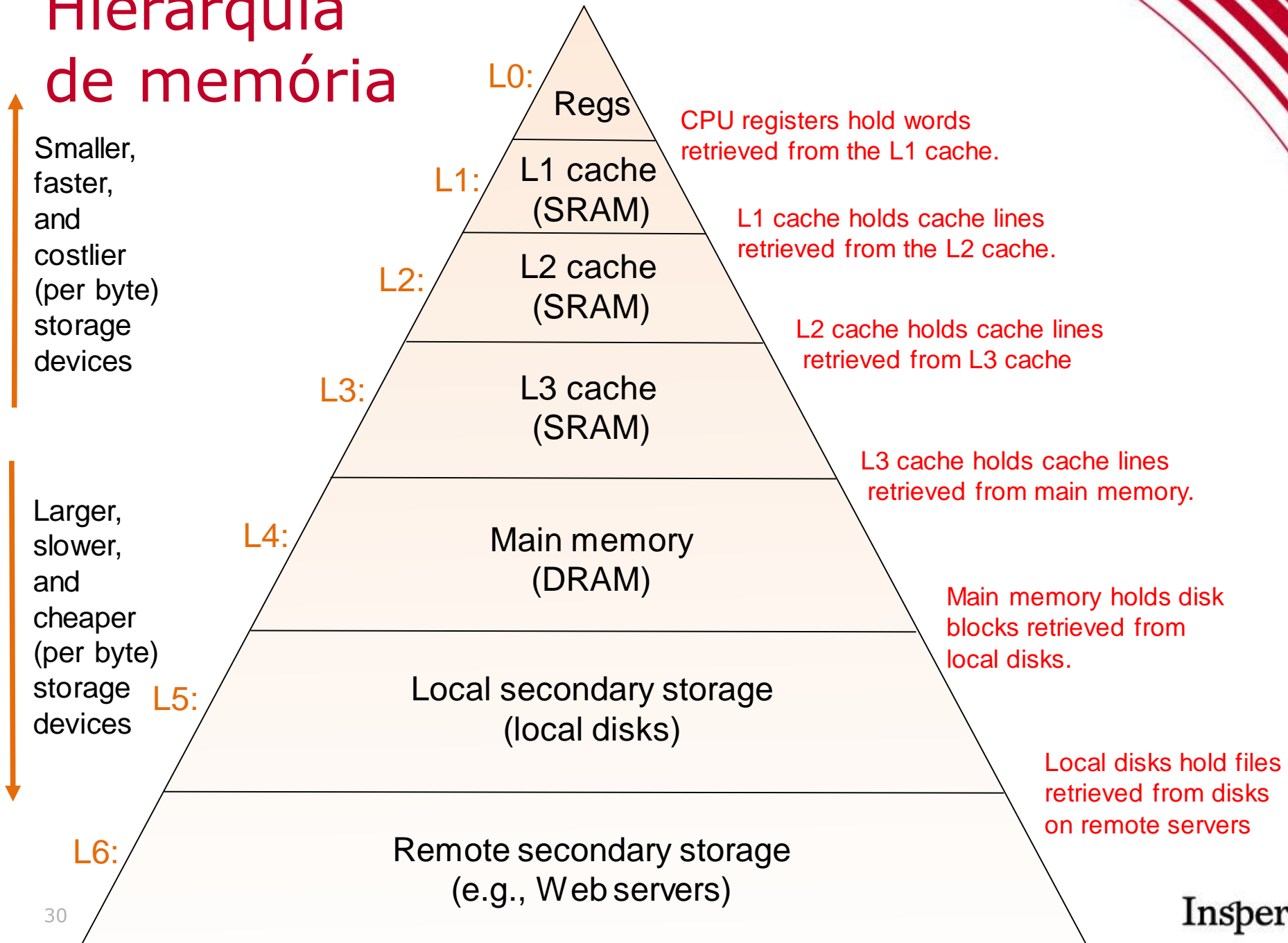


Bom, bonito e barato!

Como ter alto desempenho sem pagar tão caro?

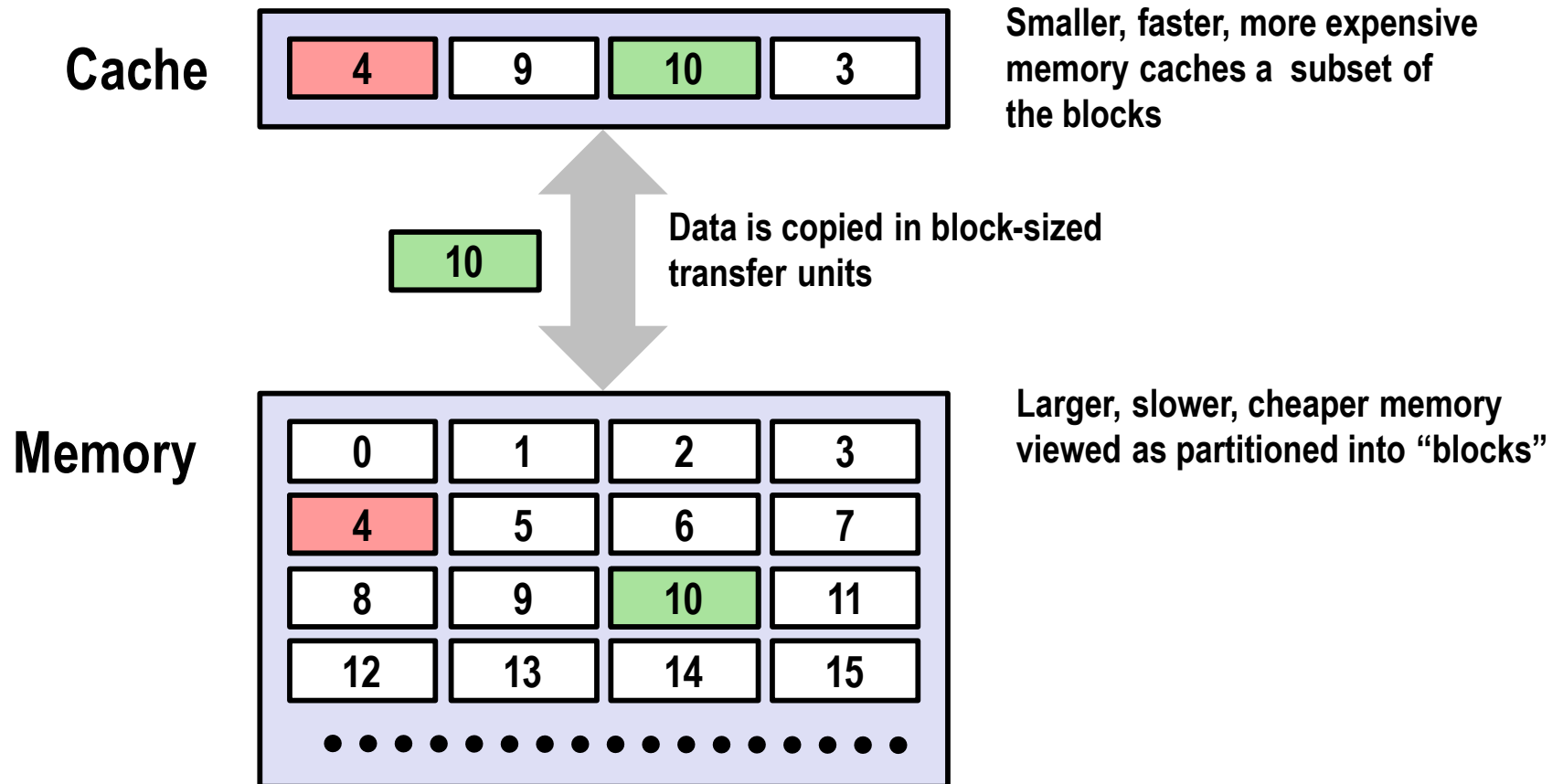
- Um sistema teórico de alto desempenho seria feito inteiramente de memória cache SRAM e armazenamento em SSD!
- Um sistema teórico de baixo custo seria feito de memória DRAM, sem cache, e com discos rígidos mecânicos!
- Nenhum dos dois extremos presta!

Hierarquia de memória



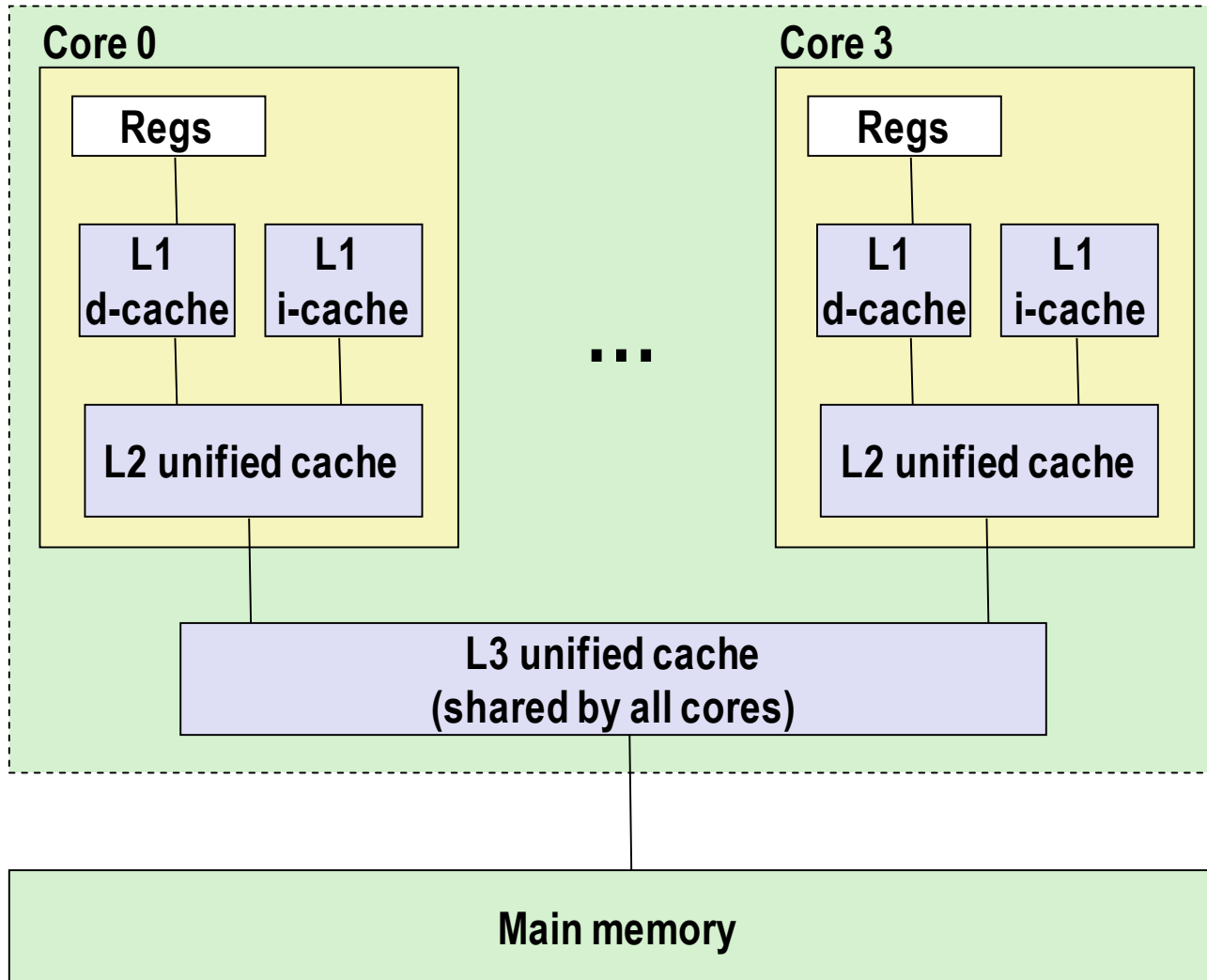
ápido.

Guarda os itens mais recentes no cache, para reuso mais rápido.



Arquitetura Intel i7

Processor package



L1 i-cache and d-cache:

32 KB, 8-way,
Access: 4 cycles

L2 unified cache:

256 KB, 8-way,
Access: 10 cycles

L3 unified cache:

8 MB, 16-way,
Access: 40-75 cycles

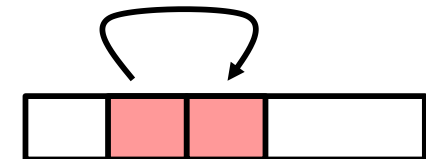
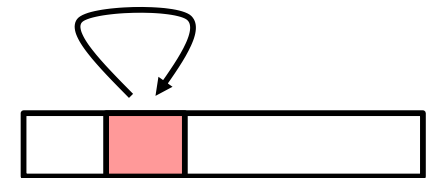
Block size: 64 bytes
for all caches.

Localidade

A solução deste dilema é observar que os programas de computador tendem a apresentar **localidade**:

Programas tendem a usar dados e instruções cujos endereços são os mesmos ou próximos àqueles usados recentemente

- Localidade temporal:
Reutilizar o mesmo endereço
- Localidade espacial:
Utilizar endereços adjacentes



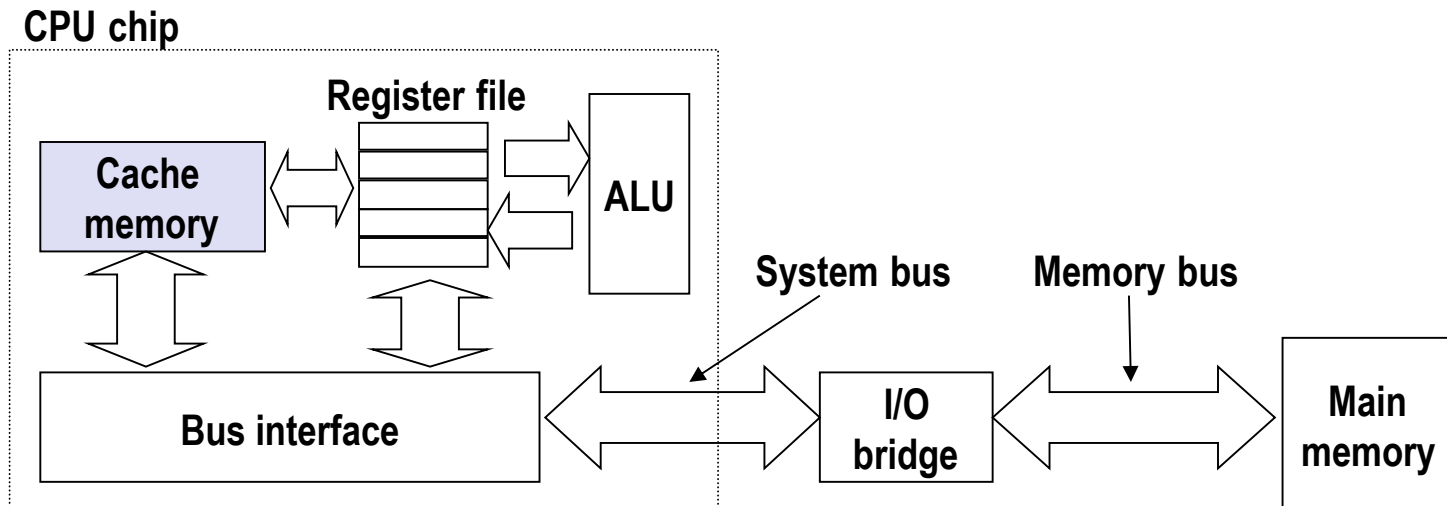
Tipos de cache miss

- ***Cold miss***: O cache estava vazio
- ***Capacity miss***: O *working set* (o conjunto de dados a ser trabalhado) é maior que o cache
- ***Conflict miss***: Apesar de o cache não estar cheio, dois blocos são alocados para exatamente a mesma posição do cache

Arquitetura típica

Cache:

- Mantém blocos de memória acessados frequentemente
- Memória rápida (SRAM), tamanho pequeno
- Gerenciada automaticamente por hardware
- CPU busca dados primeiramente no cache



Código cache-friendly

- Prioriza acessos de dados vizinhos
- Evita buscas desnecessárias a memória
- Reutiliza dados sempre que possível

Como medidos isto?

- Calculamos cache miss rate = cache miss / total de acessos
- Testamos com diferentes tamanhos de entrada
- Isto também pode ser feito usando o valgrind

Atividade

Escrita de código cache-friendly e análise de uso de cache.

Hoje:

1. Execução do código ingênuo e medidas de uso de cache
2. Criação da otimização 1

Para dia 09/10:

1. Otimizações 2 e 3
2. Perguntas do handout respondidas no arquivo .c

Insper

www.insper.edu.br