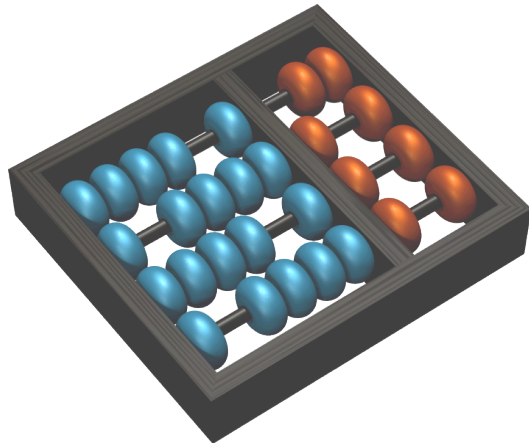


# Técnicas para desenvolvimento e aceleração de códigos científicos



**Prof. Edson Borin**  
edson@ic.unicamp.br  
Instituto de Computação  
UNICAMP

**Minicurso  
LNCC 2014**

---

# Agenda

- Apresentação
- Processadores Modernos
- Laboratório

# Sobre o minicurso

<b>Segunda</b>	<b>Terça</b>	<b>Quarta</b>	<b>Quinta</b>	<b>Sexta</b>
Introdução	Perfilamento - Contagem de tempo	Otimizações simples / compilação	Perfilamento - Detecção de código quente	GDB
Organização de processadores modernos	Otimização de acesso a dados	Bibliotecas otimizadas	SVN + CMake	Valgrind
Introdução ao laboratório		Vetorização de código		

# Sobre o minicurso

<b>Segunda</b>	<b>Terça</b>	<b>Quarta</b>	<b>Quinta</b>	<b>Sexta</b>
Introdução	Perfilamento - Contagem de tempo	Otimizações simples / compilação	Perfilamento - Detecção de código quente	GDB
Organização de processadores modernos	Otimização de acesso a dados	Bibliotecas otimizadas	SVN + CMake	Valgrind
Introdução ao laboratório		Vetorização de código		

# Sobre o minicurso

<b>Segunda</b>	<b>Terça</b>	<b>Quarta</b>	<b>Quinta</b>	<b>Sexta</b>
Introdução	Perfilamento - Contagem de tempo	Otimizações simples / compilação	Perfilamento - Detecção de código quente	GDB
Organização de processadores modernos	Otimização de acesso a dados	Bibliotecas otimizadas	SVN + CMake	Valgrind
Introdução ao laboratório		Vetorização de código		

# Sobre o minicurso

<b>Segunda</b>	<b>Terça</b>	<b>Quarta</b>	<b>Quinta</b>	<b>Sexta</b>
Introdução	Perfilamento - Contagem de tempo	Otimizações simples / compilação	Perfilamento - Detecção de código quente	GDB
Organização de processadores modernos	Otimização de acesso a dados	Bibliotecas otimizadas	SVN + CMake	Valgrind
Introdução ao laboratório		Vetorização de código		

# Sobre o minicurso

<b>Segunda</b>	<b>Terça</b>	<b>Quarta</b>	<b>Quinta</b>	<b>Sexta</b>
Introdução	Perfilamento - Contagem de tempo	Otimizações simples / compilação	Perfilamento - Detecção de código quente	GDB
Organização de processadores modernos	Otimização de acesso a dados	Bibliotecas otimizadas	SVN + CMake	Valgrind
Introdução ao laboratório		Vetorização de código		

# Experiência

- Arquitetura e microarquitetura de processadores modernos
- Compiladores
- Implementação de Máquinas Virtuais
- Computação de Alto Desempenho



# Pesquisa em CAD

## Aceleração de métodos numéricos

- Devloo: MEF para simulações de poços e reservatórios de petróleo. (Petrobrás)
- Tygel: Métodos para Imageamento Sísmico. (Petrobrás)
- Bittencourt: MEF de alta ordem. (Fapesp)

## Outras colaborações

- Torres: Aceleração de algoritmos para CBIR. (AMD)

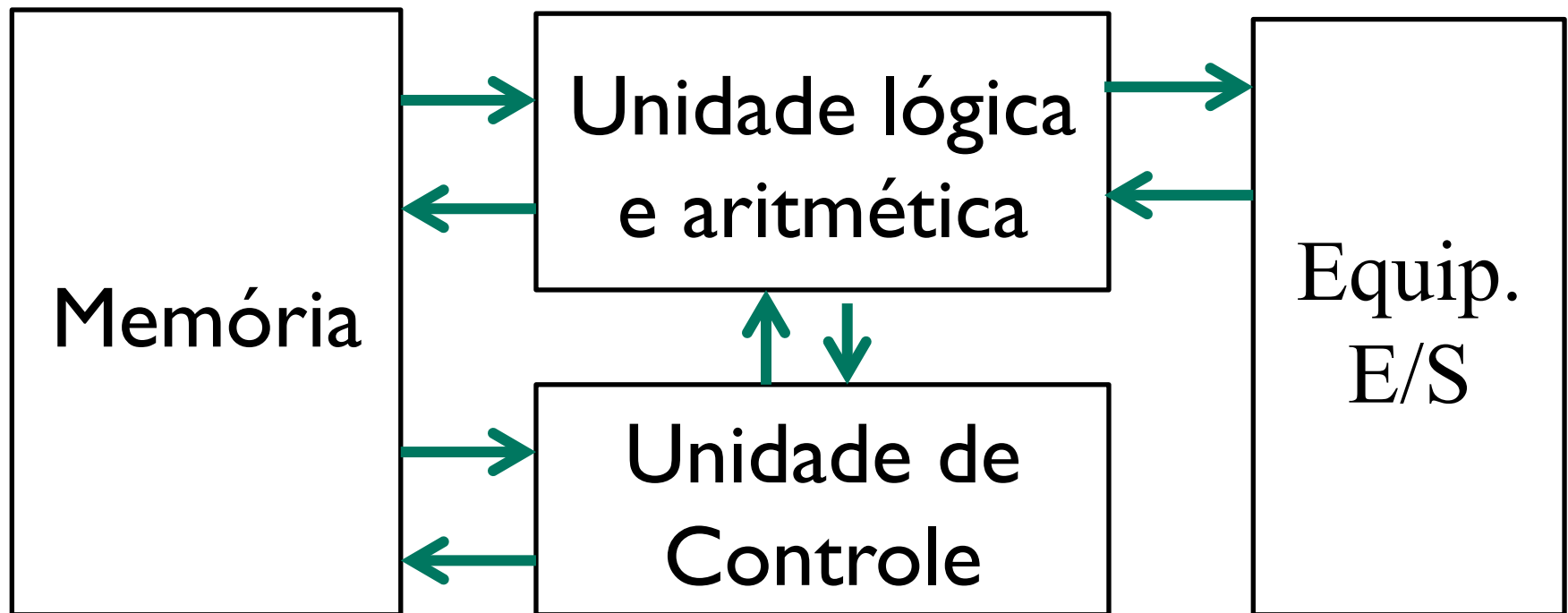
# Agenda

- Apresentação
- **Processadores Modernos**
- Laboratório

# Processadores Modernos

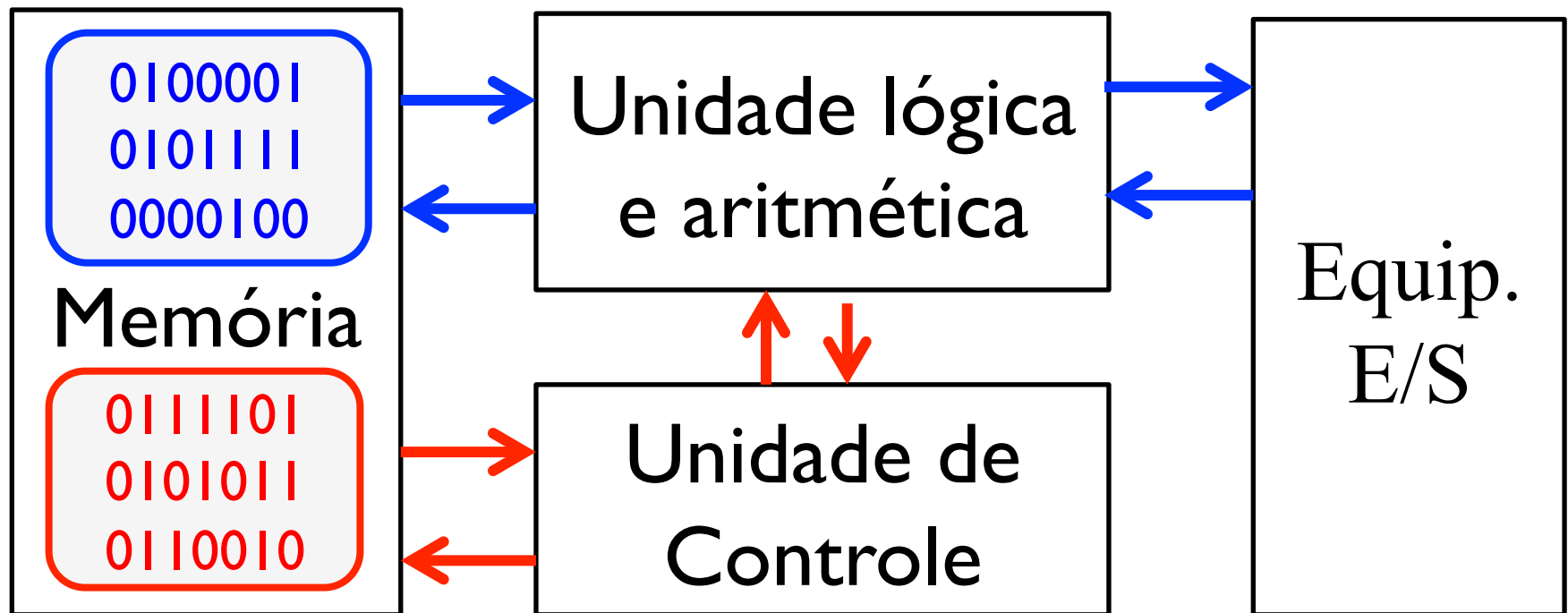
# Processadores Modernos

Baseados em conceitos desenvolvidos no final da década de 40 (EDVAC, IAS, ...)



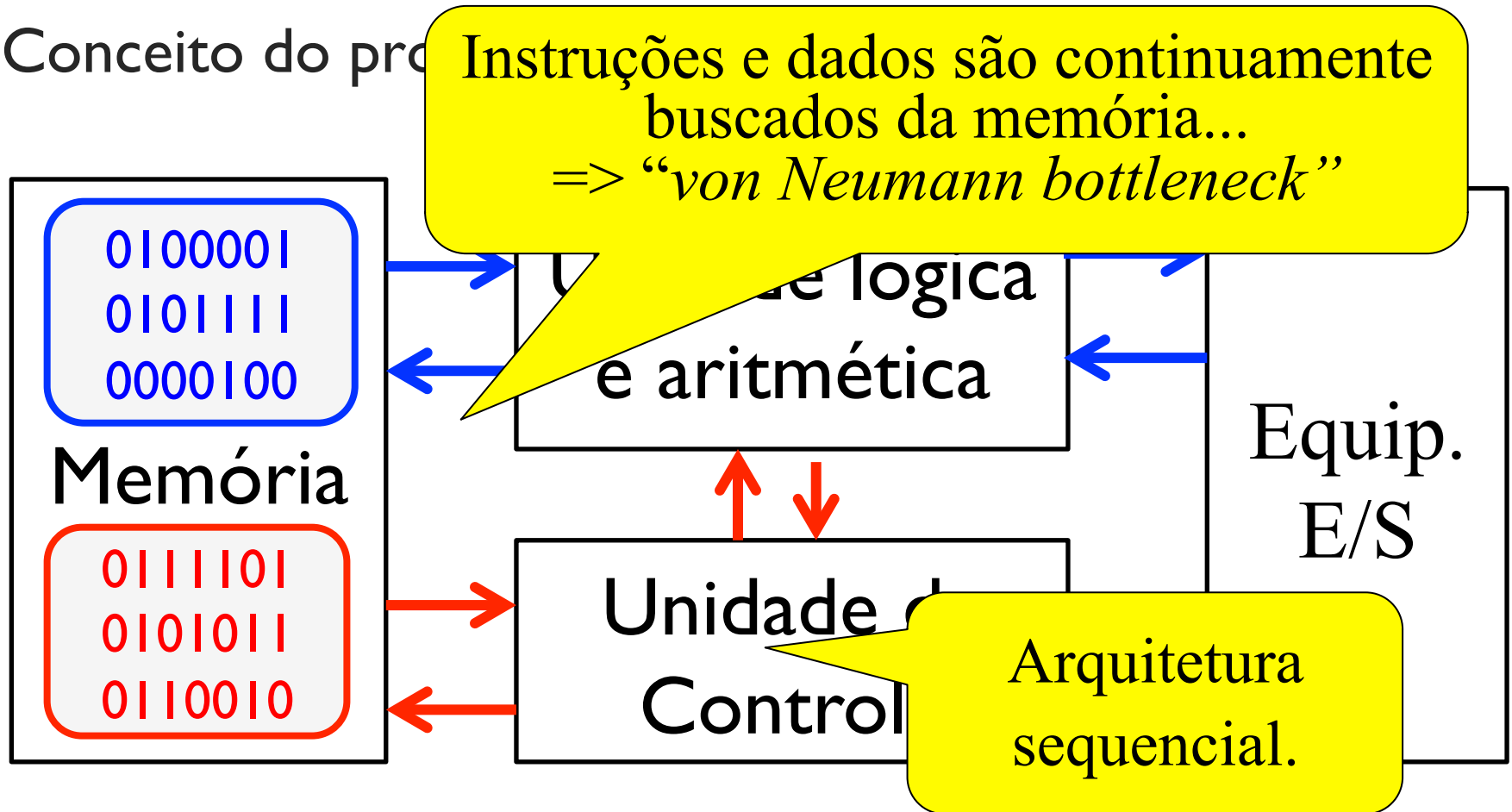
# Processadores Modernos

Conceito do programa armazenado



# Processadores Modernos

Conceito do pro



# Processadores Modernos

**Arquitetura/Modelo de Programação** ainda são parecidos com o que foi proposto no final da década de 40 (conceito do programa armazenado).

- Basicamente instruções para movimentar dados entre a memória e registradores e realizar operações lógicas e aritméticas.
- Instruções executadas sequencialmente. Este é o modelo visto pelo programador! Na prática instruções podem ser executadas em paralelo.

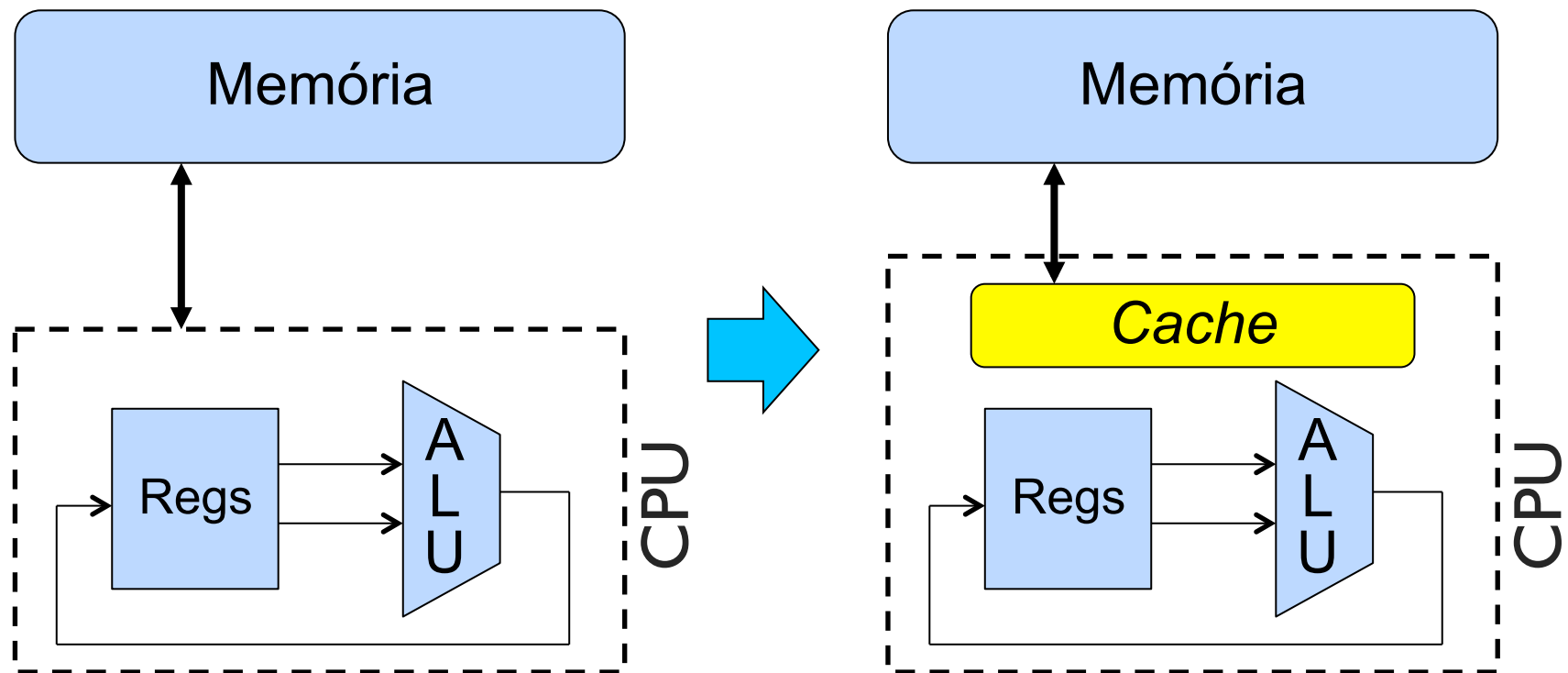
# Processadores Modernos

**Microarquitetura** evoluiu consideravelmente e atualmente incorpora diversos mecanismos sofisticados!



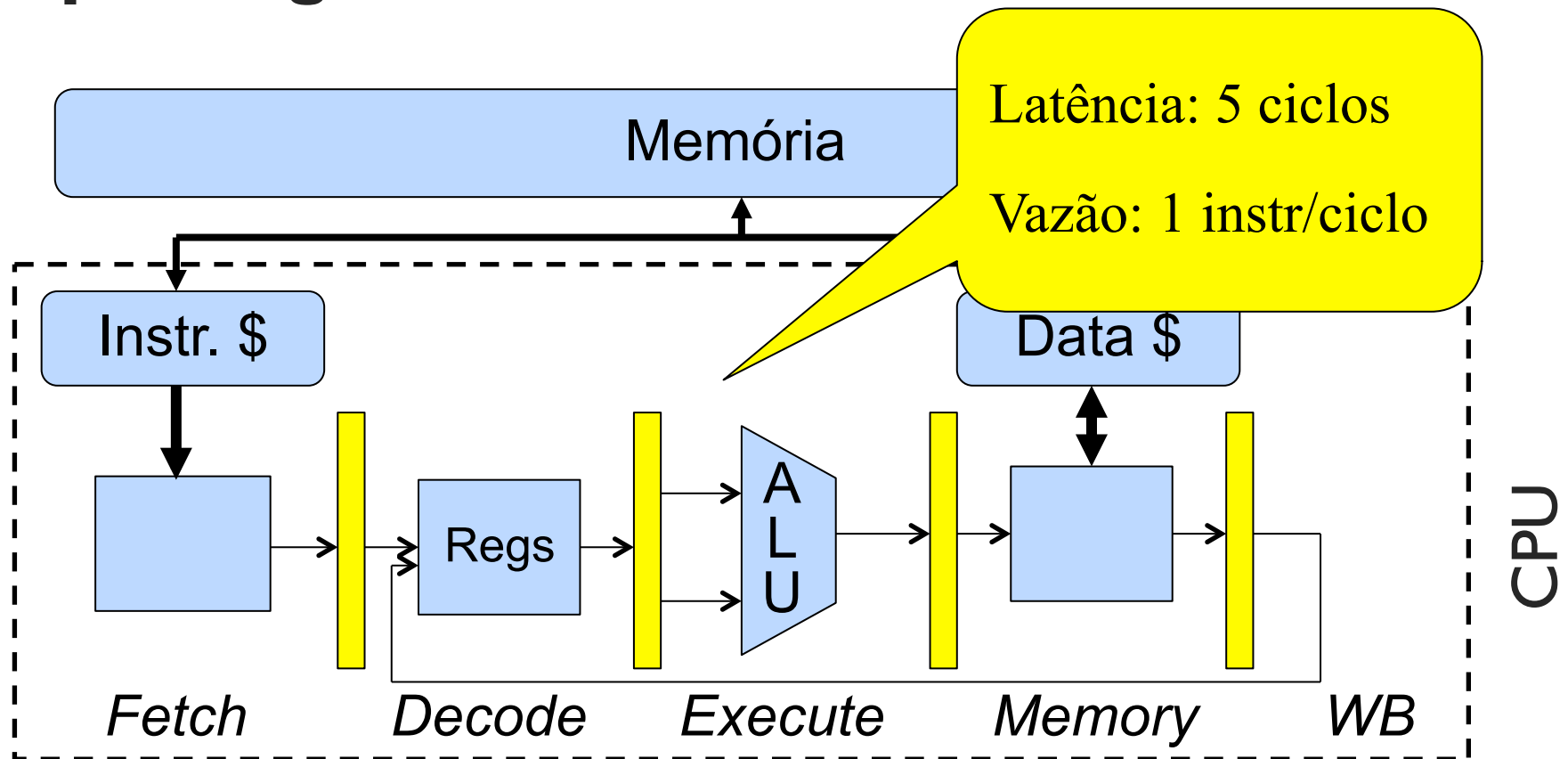
# Processadores Modernos

## Caches



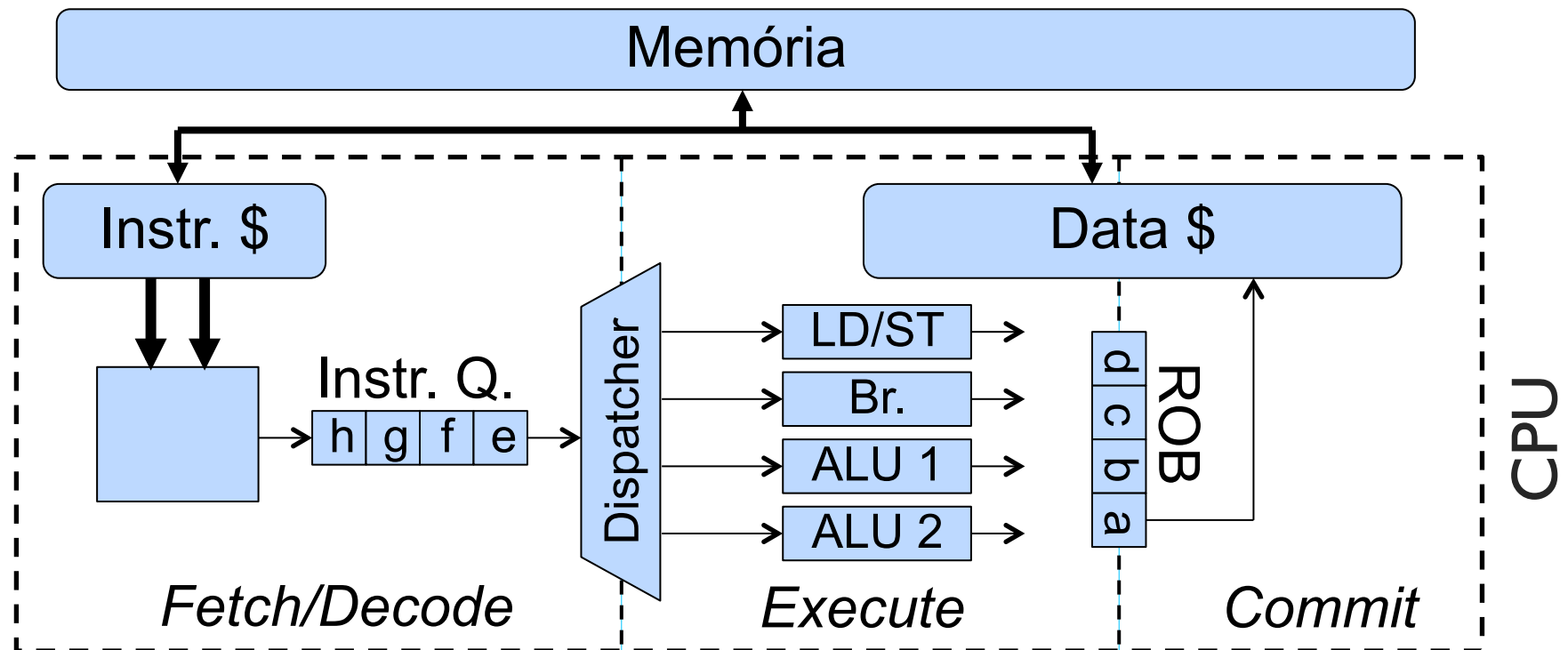
# Processadores Modernos

## Pipelining



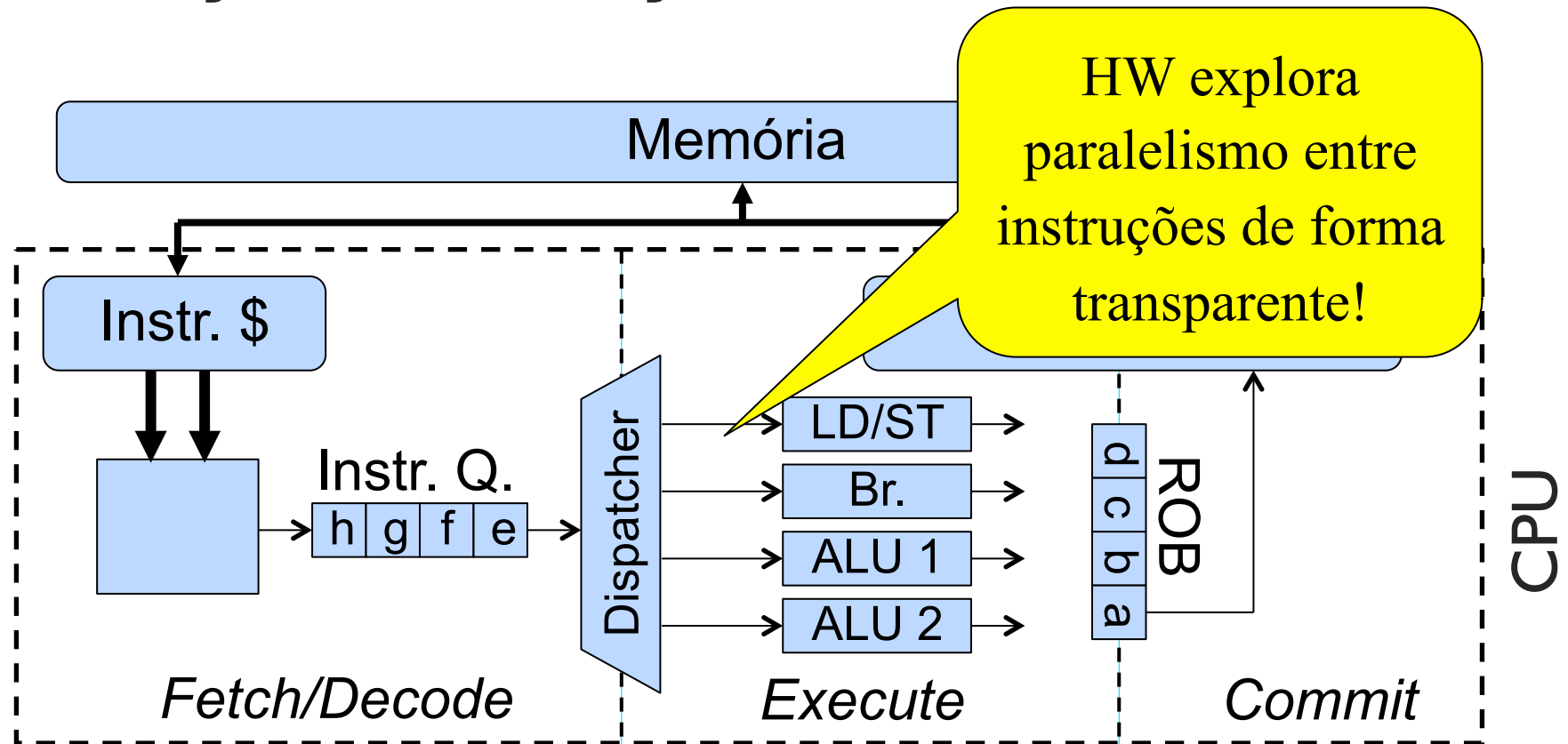
# Processadores Modernos

## Execução de Instruções Fora de Ordem



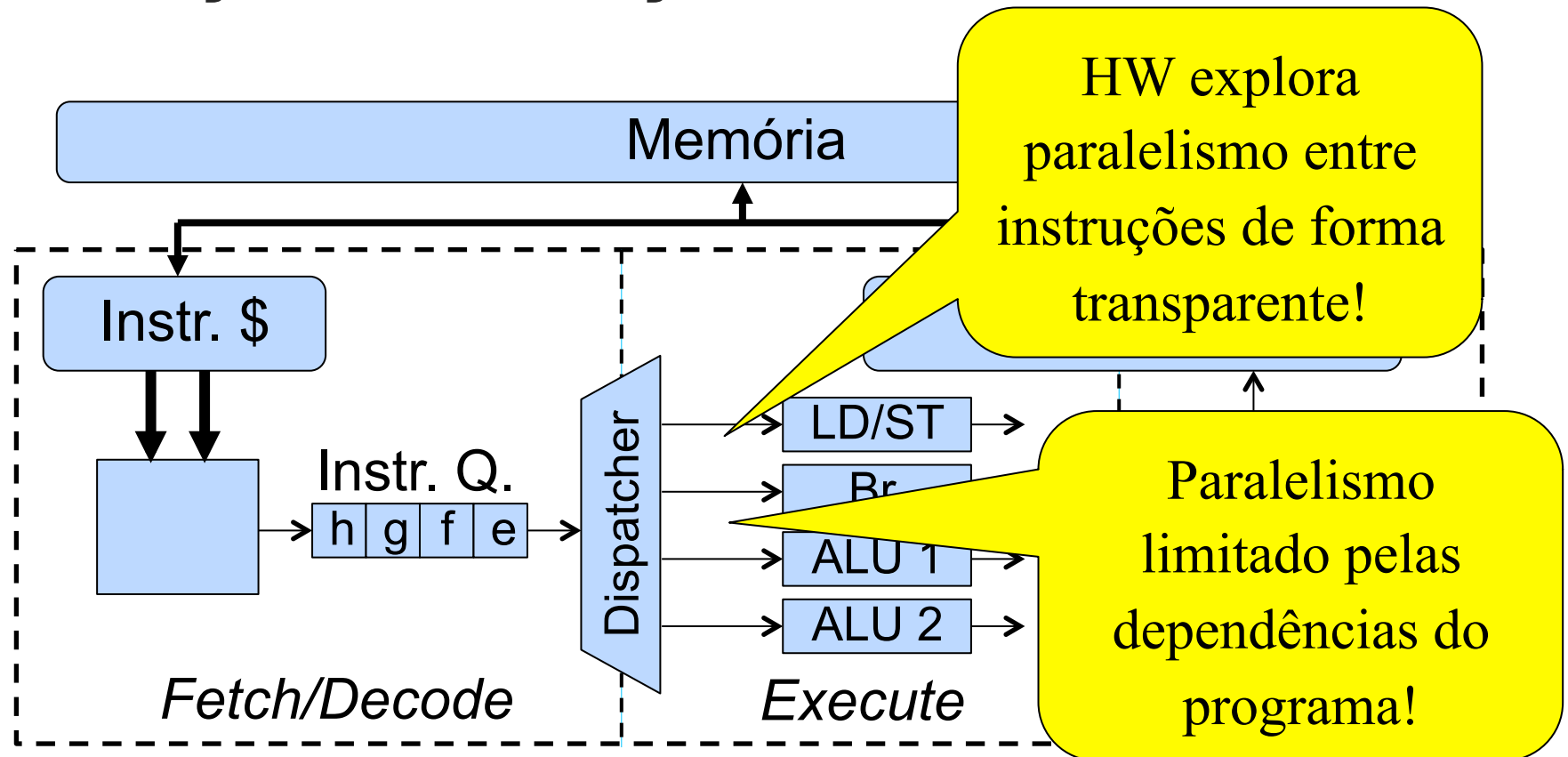
# Processadores Modernos

## Execução de Instruções Fora de Ordem



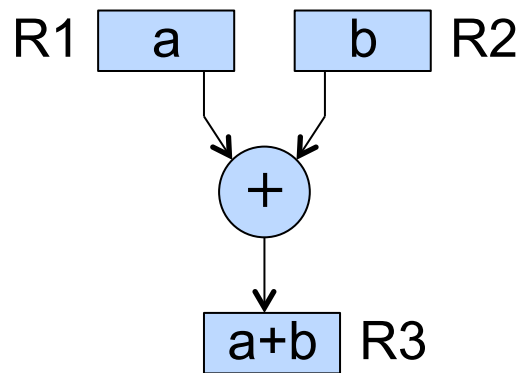
# Processadores Modernos

## Execução de Instruções Fora de Ordem

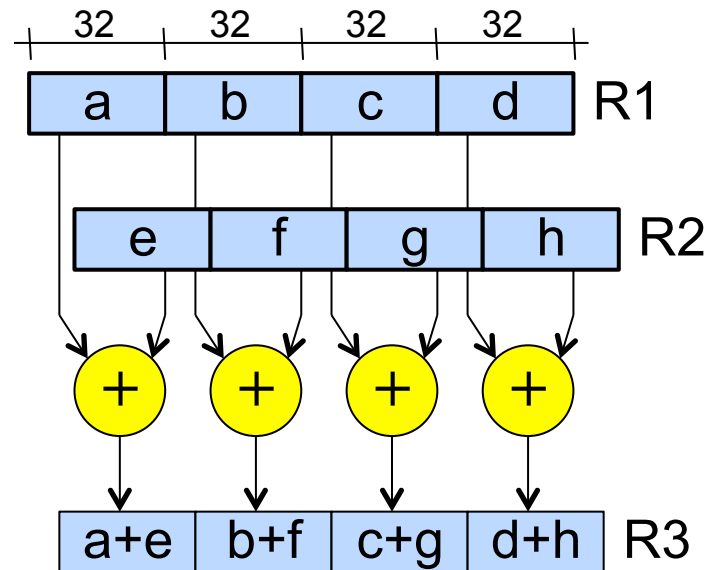
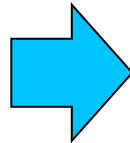


# Processadores Modernos

## SIMD



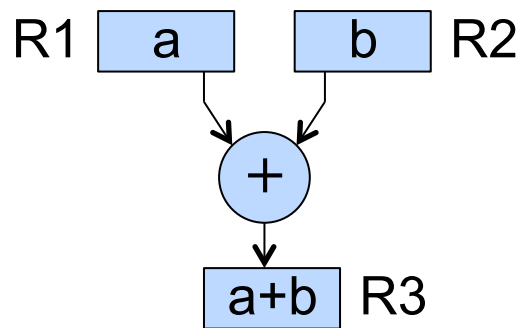
Escalar



SIMD

# Processadores Modernos

## SIMD

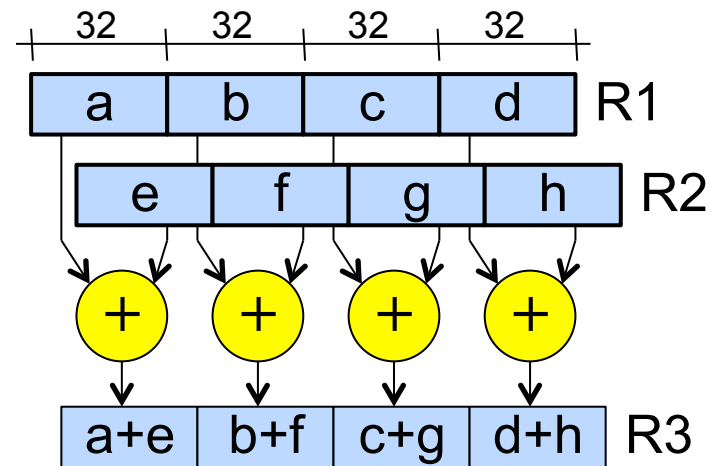
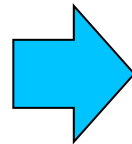


Escalar

`addps %xmm0, %xmm1`

`addpd %xmm0, %xmm1`

`vaddpd %ymm1, %ymm0, %ymm0`



SIMD

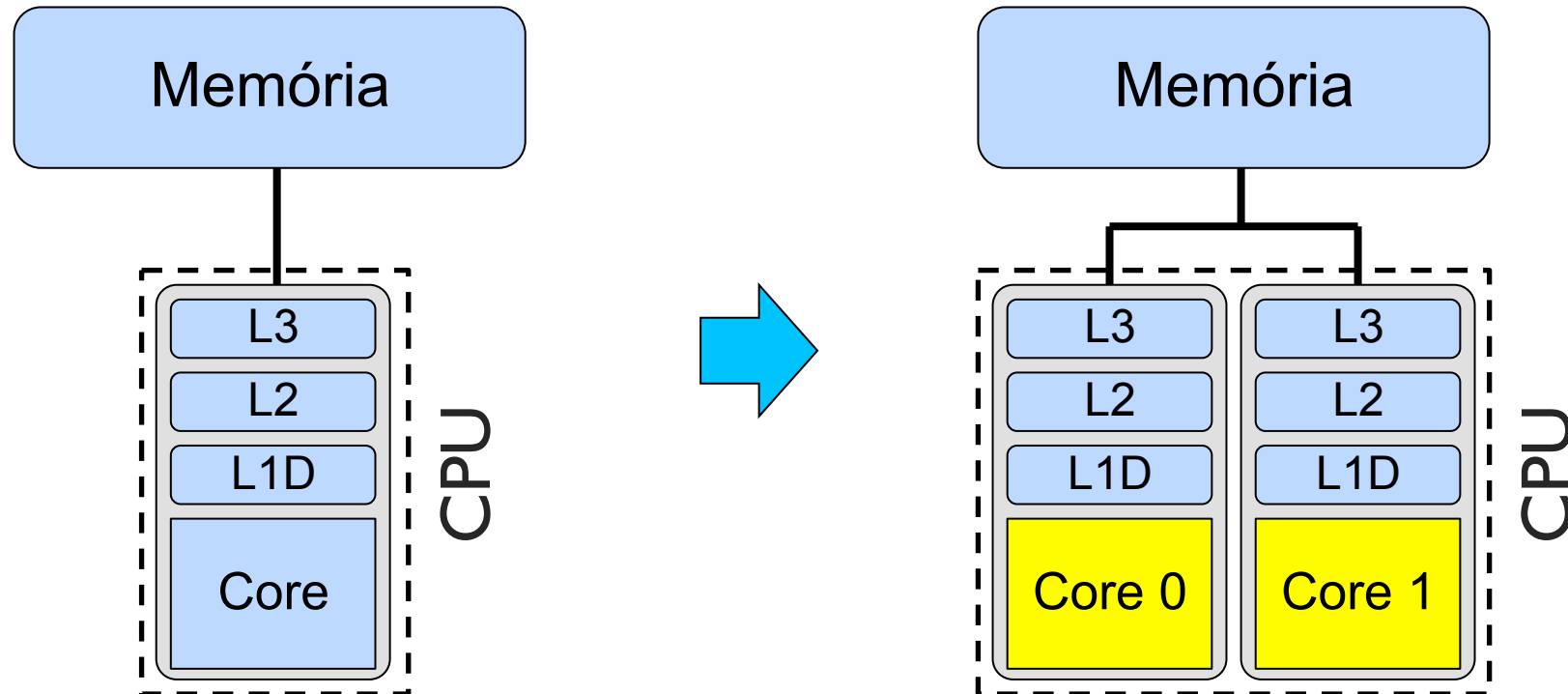
(SSE/128bits – 4 SP)

(SSE2/128bits – 2 DP)

(AVX/256bits – 4 DP)

# Processadores Modernos

## Múltiplos núcleos (*Multicore*)

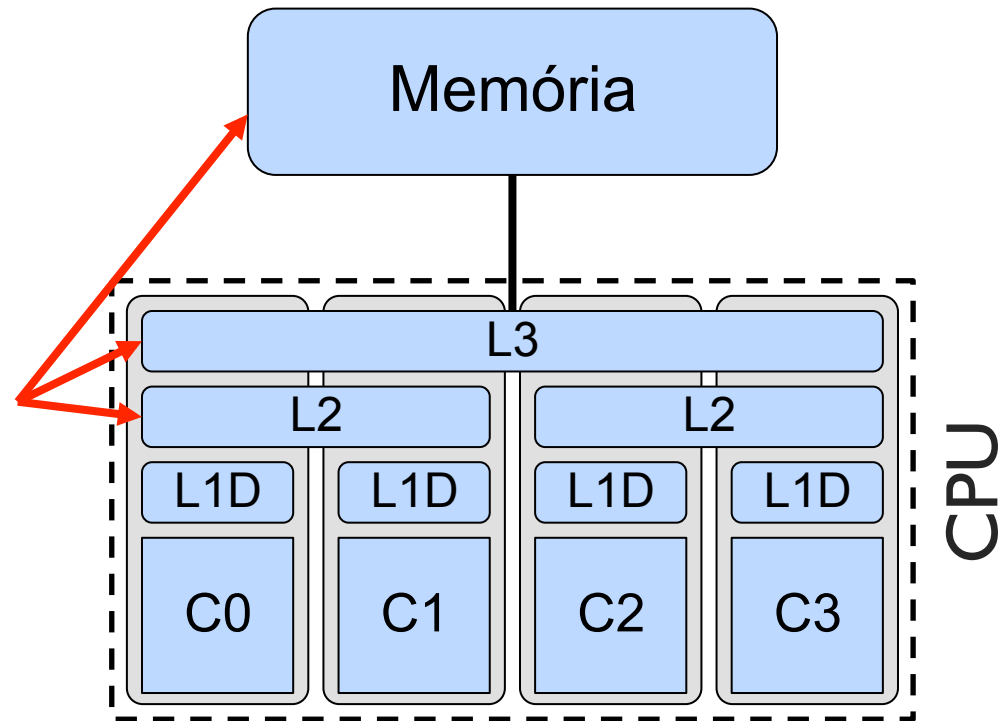




# Processadores Modernos

## Múltiplos núcleos (*Multicore*)

Além da memória,  
outros recursos podem  
ser compartilhados!



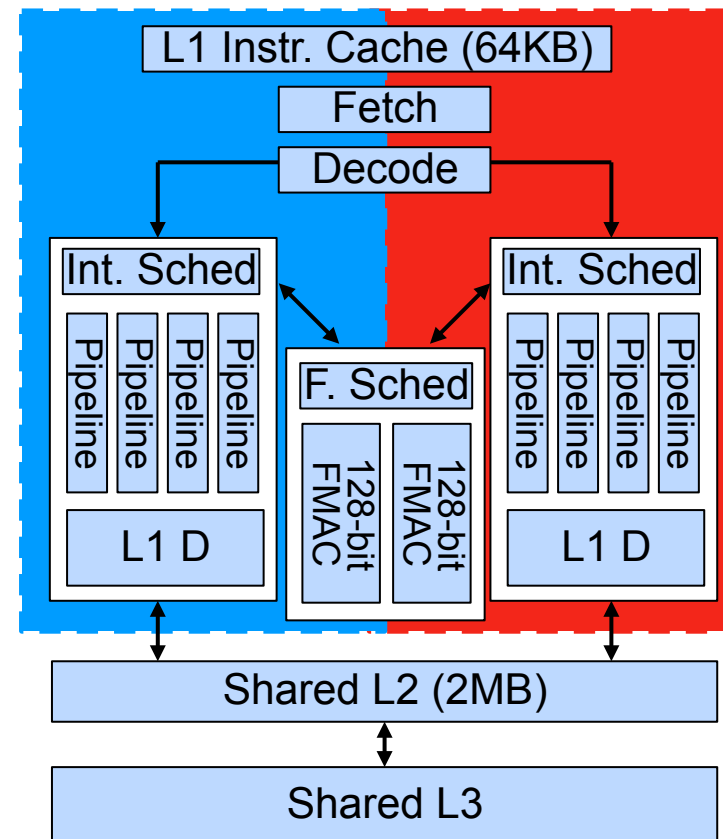
# Processadores Modernos

## Múltiplos núcleos (*Multicore*)

AMD Bulldozer

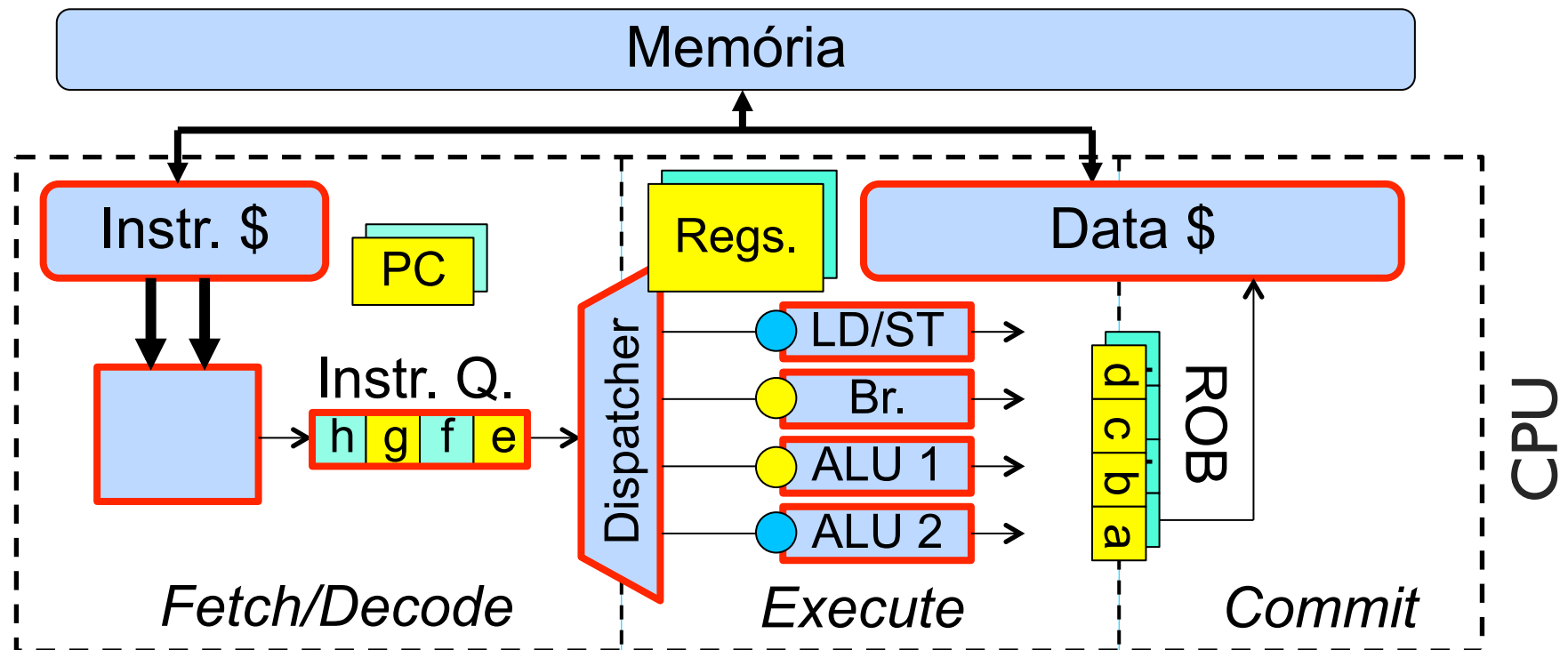
Vários recursos compartilhados:

- Cache L2 e L3
- Cache L1 de instruções
- Unidade de Fetch/Decode
- Unidades de Ponto Flutuante



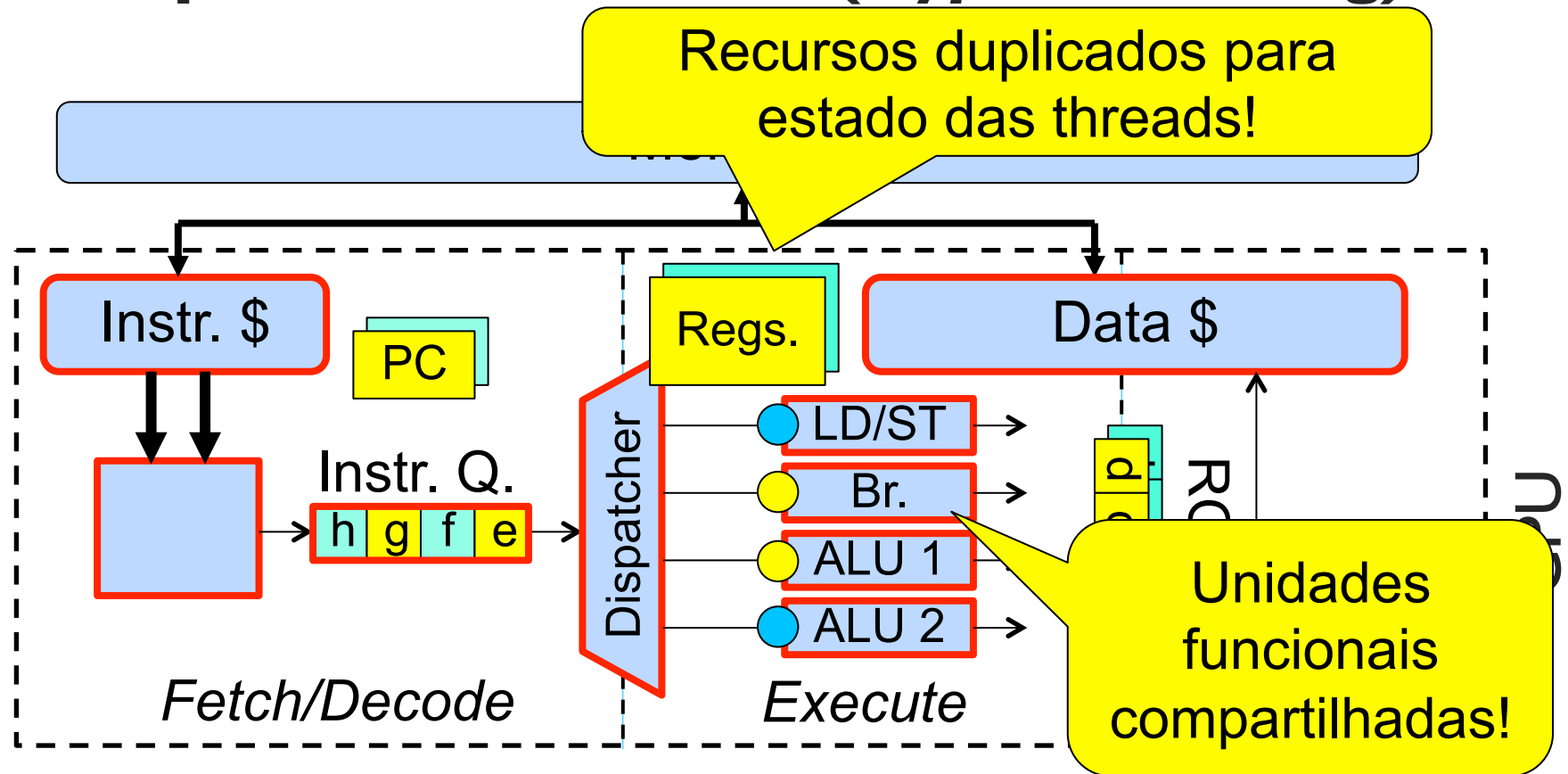
# Processadores Modernos

## Múltiplas *Threads* - SMT (*HyperThreading*)



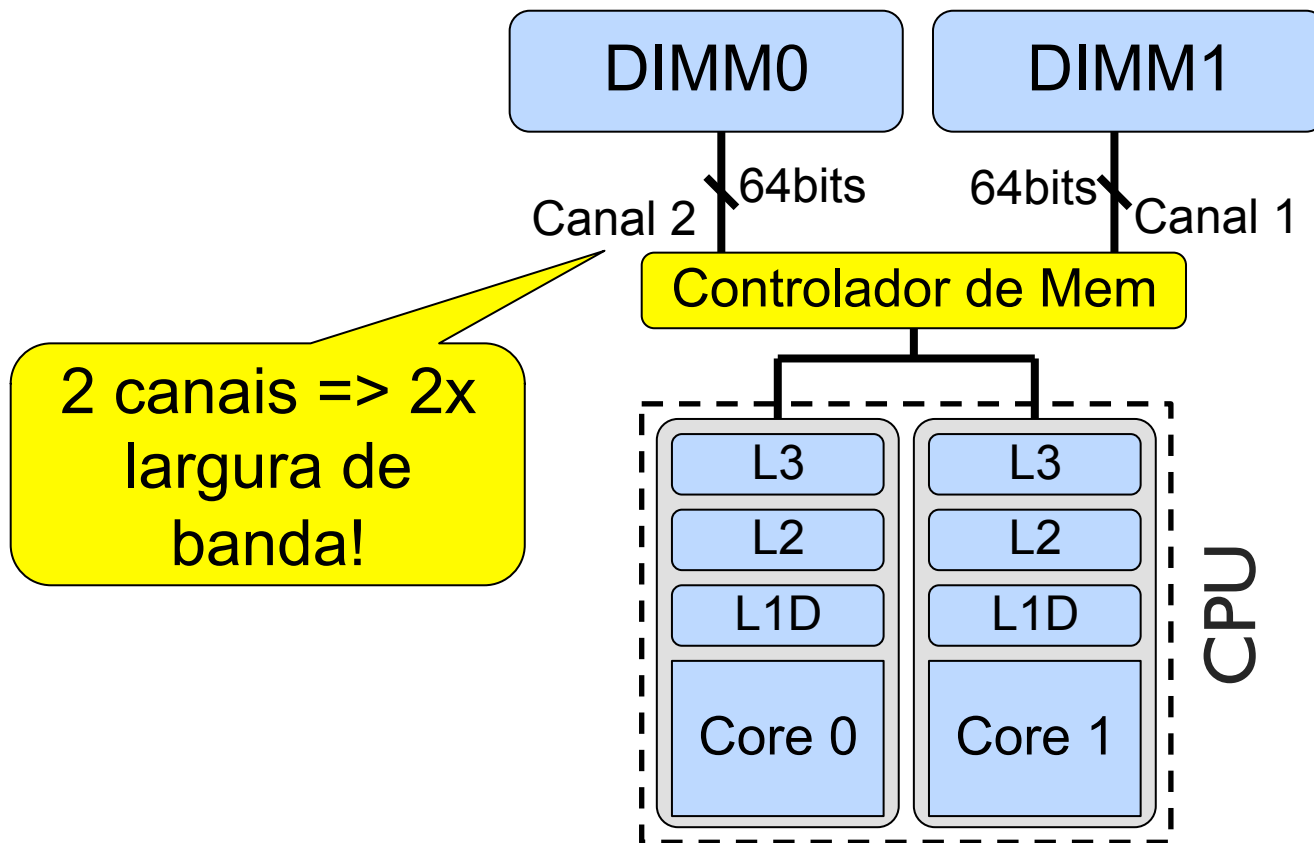
# Processadores Modernos

## Múltiplas *Threads* - SMT (*HyperThreading*)



# Processadores Modernos

## Múltiplos Controladores e Canais de Memória

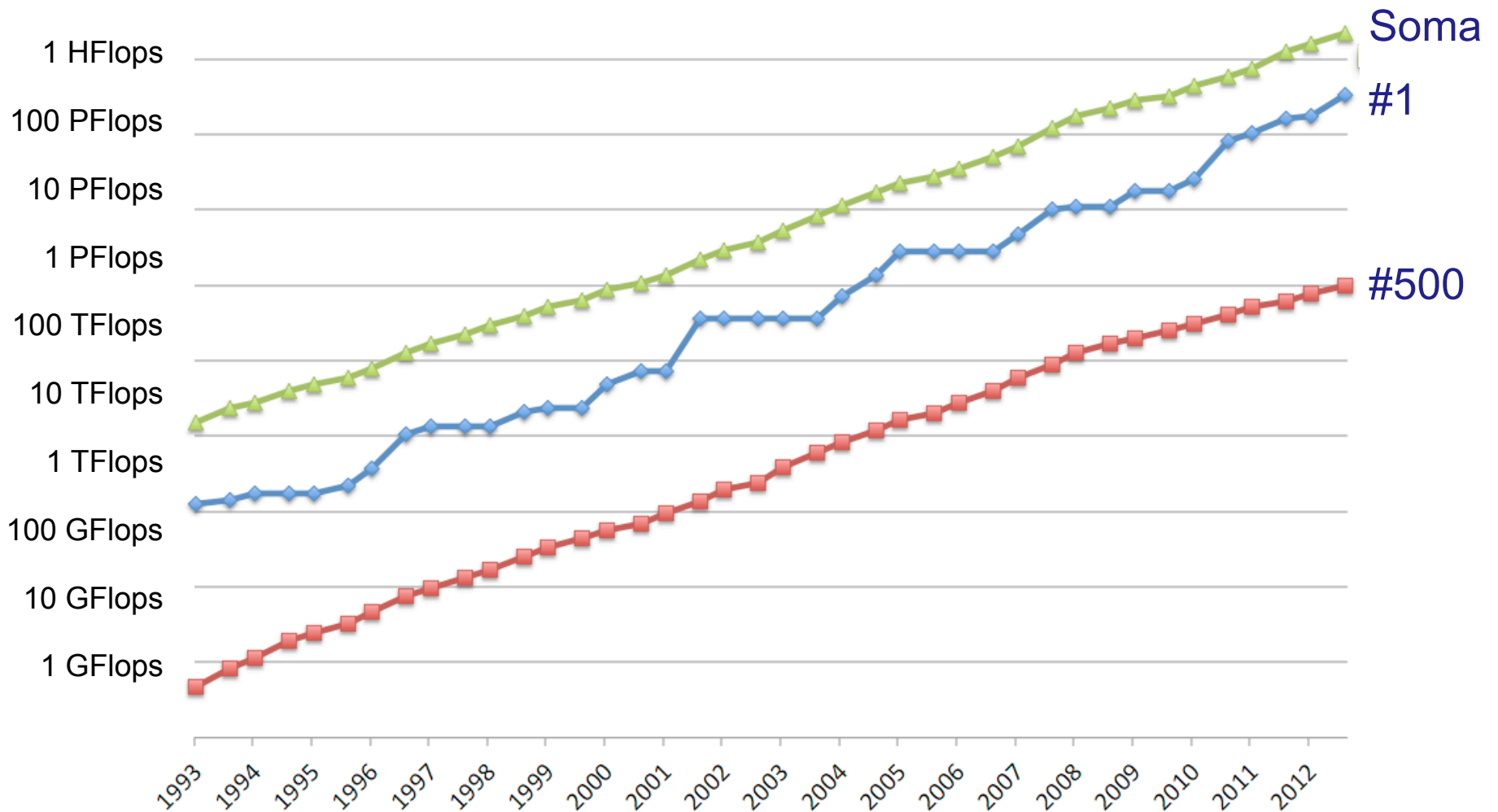


# Processadores Modernos

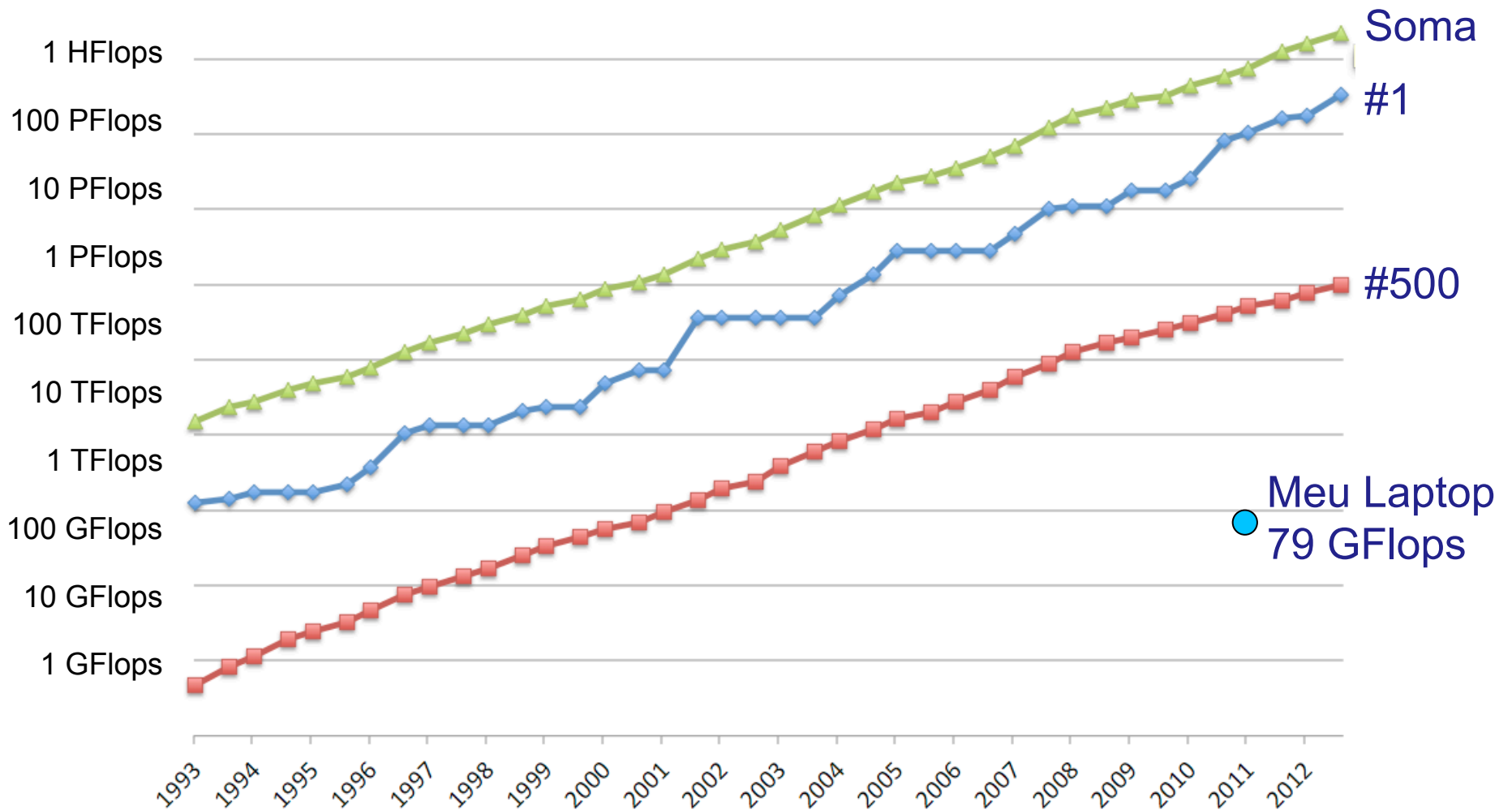
Desempenho de **processadores** evoluía exponencialmente.

Desempenho de **super-computadores** ainda evolui exponencialmente (adição de mais processadores/núcleos).

# Top 500: evolução do desempenho

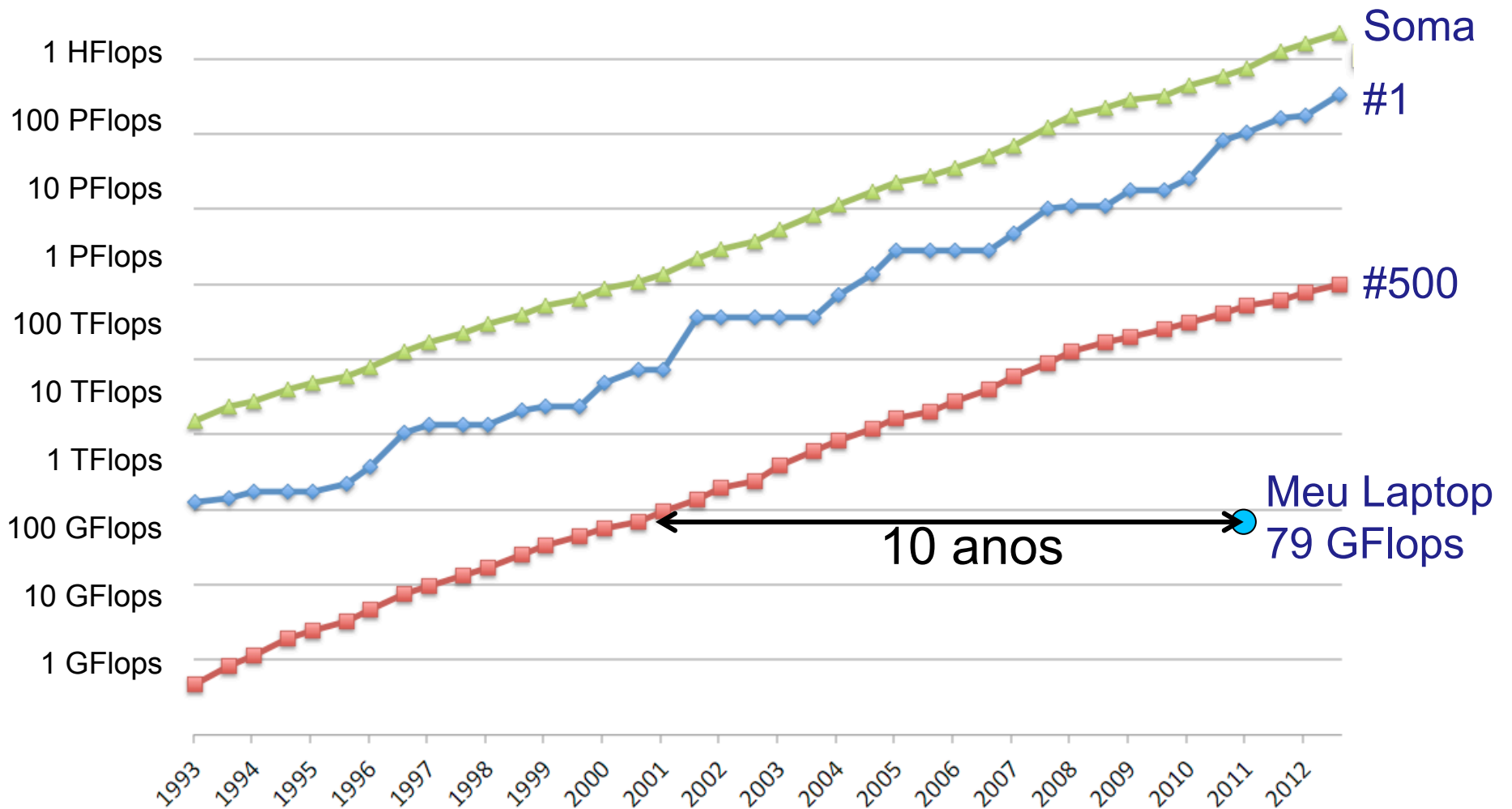


# Top 500: evolução do desempenho

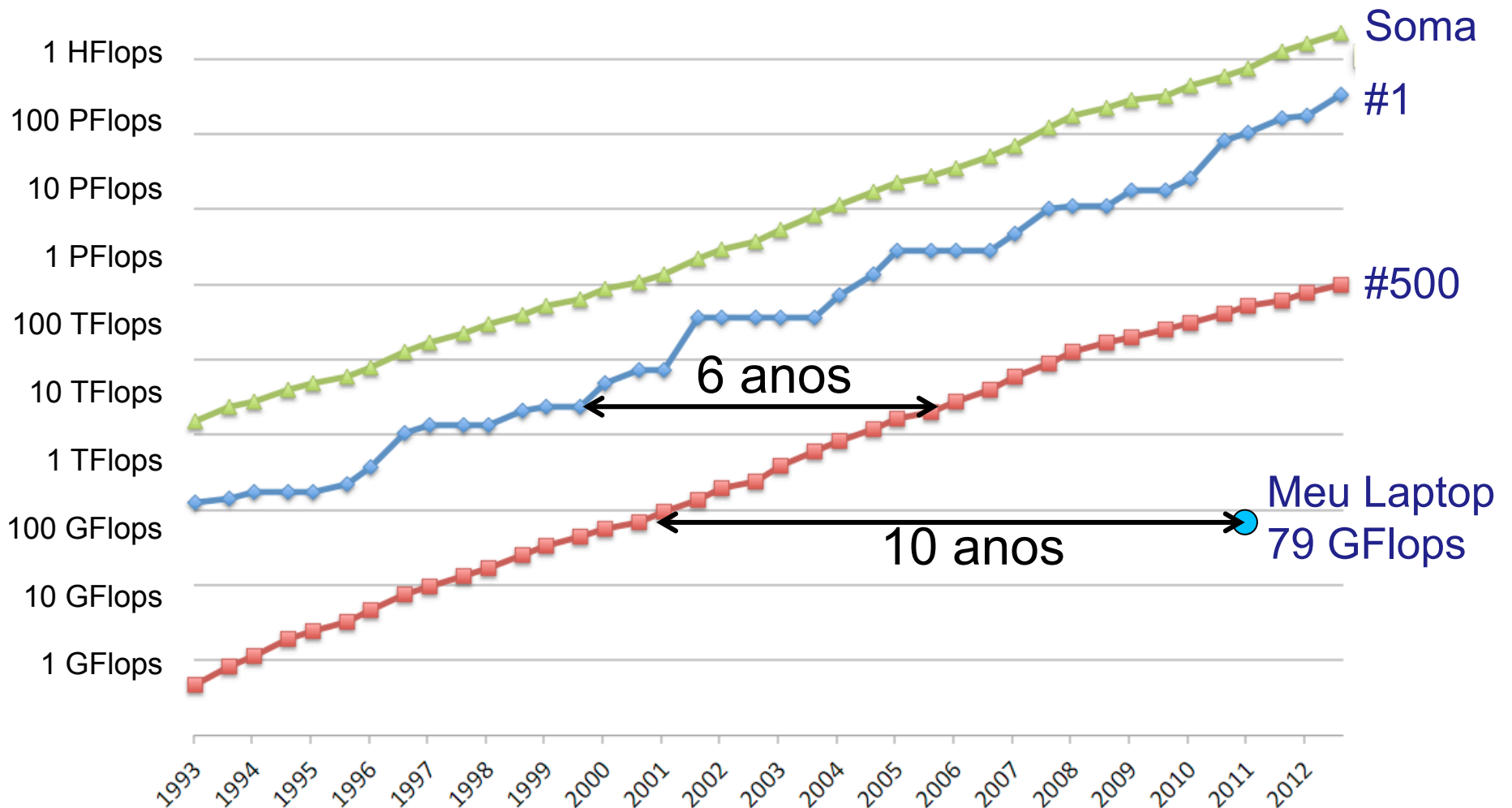




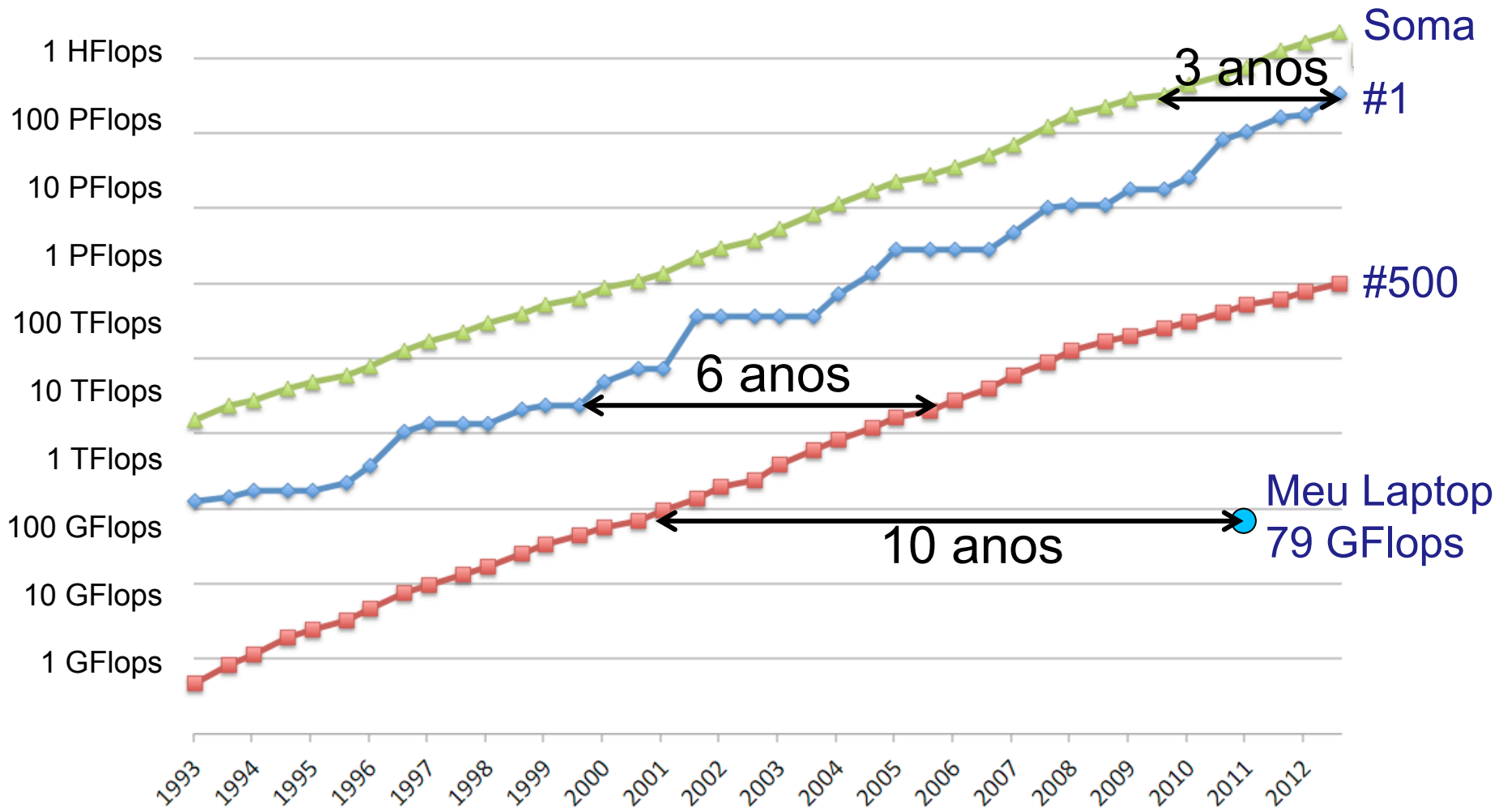
# Top 500: evolução do desempenho



# Top 500: evolução do desempenho



# Top 500: evolução do desempenho



# Agenda

- Apresentação
- Processadores Modernos
- **Laboratório**

# Atividade de Laboratório

- **Página:**