

# 02 – Avaliação do Desempenho

Luís Paulo Santos

Arquitectura de Computadores

Universidade do Minho

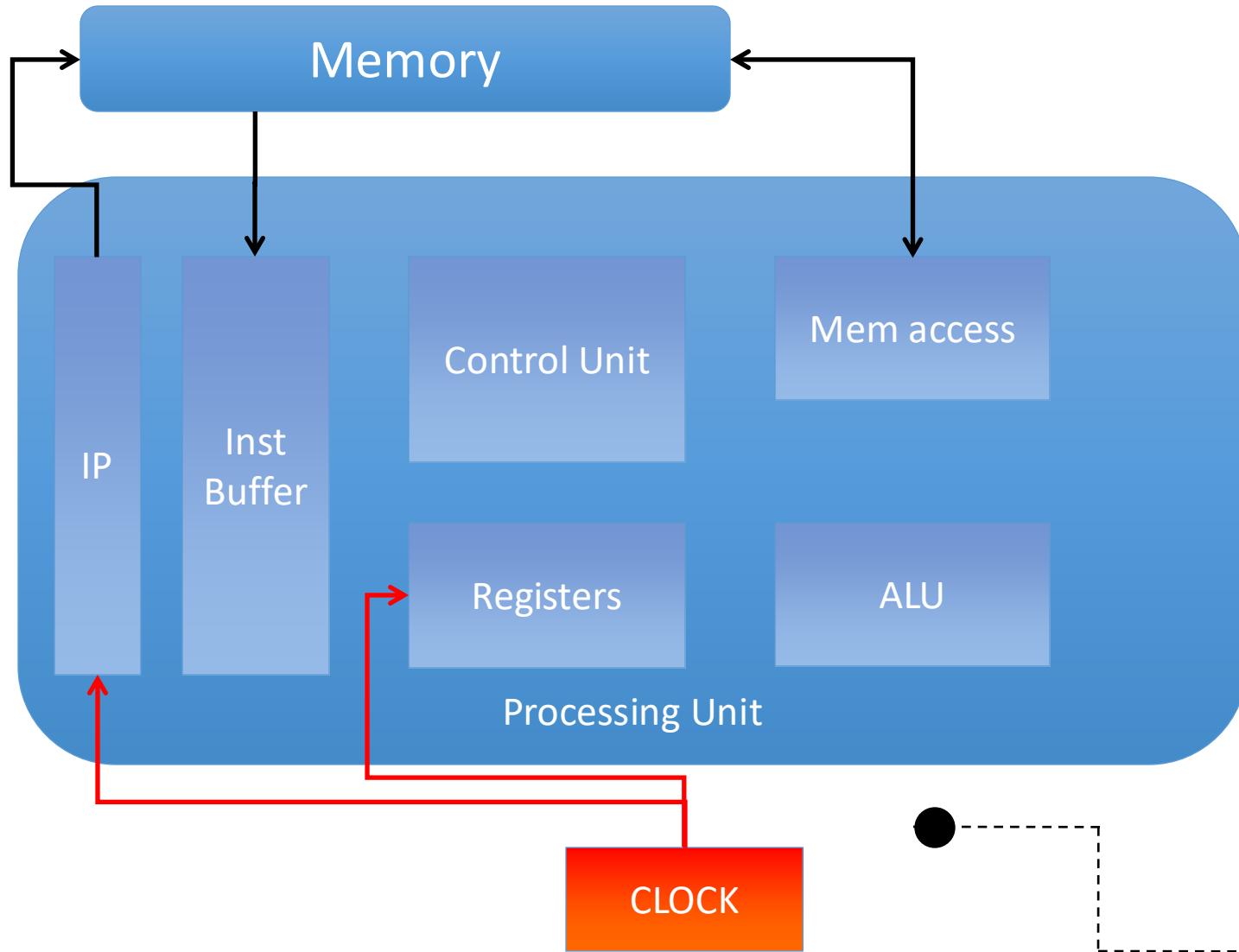
# Material de apoio

- “*Computer Organization and Design: The Hardware / Software Interface*”  
David A. Patterson, John L. Hennessy  
5th Edition, 2013
  - Secção 1.6 (pags. 28 .. 40) – Performance
  - Secção 1.10 (pags. 59 .. 51) – Fallacies and Pitfalls

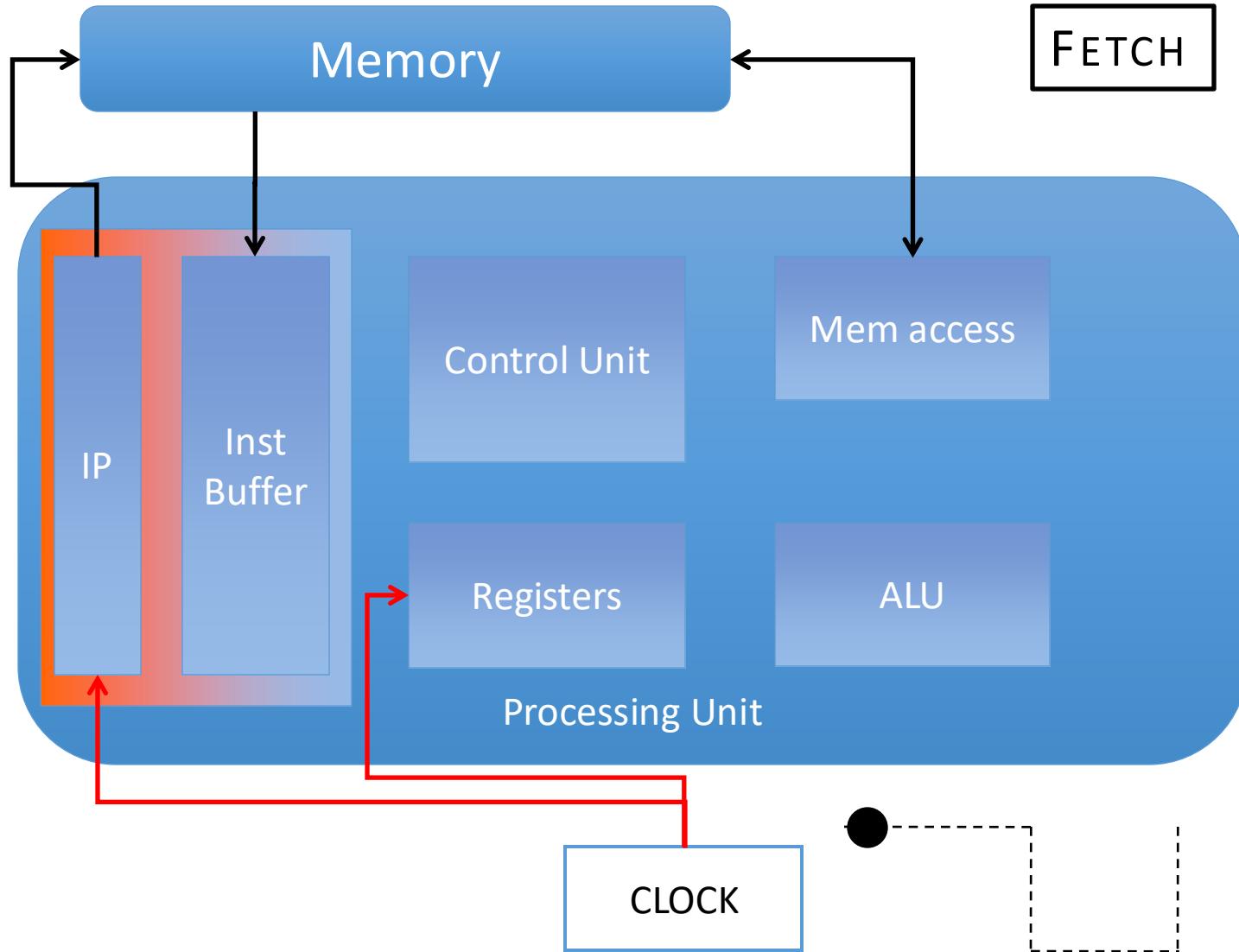
# Questão

- Como podemos modelar o desempenho de um programa, permitindo:
  - prever o seu tempo de execução num determinado sistema de computação
  - formular hipóteses sobre o tempo de execução observado

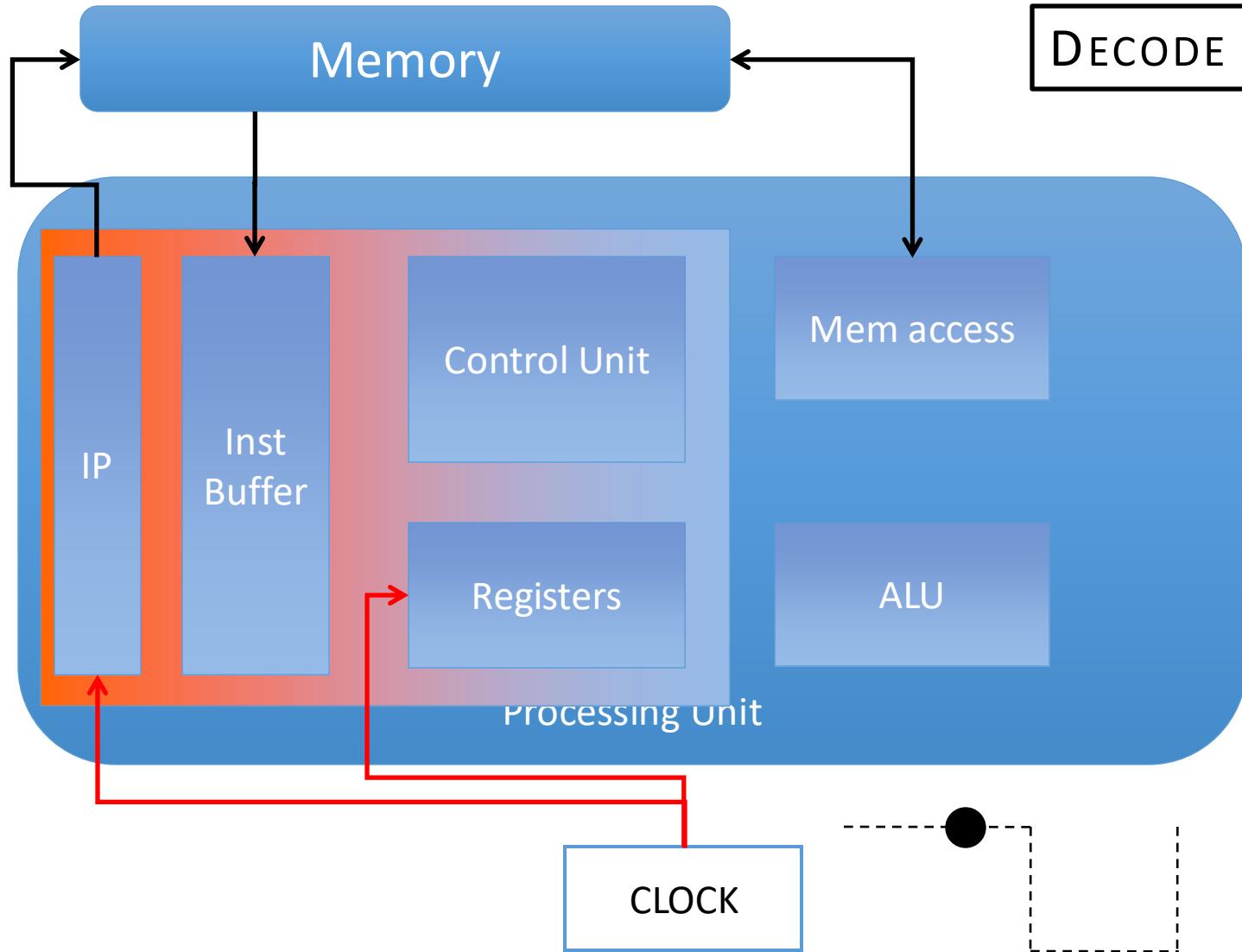
# Avaliação Desempenho



# Avaliação Desempenho

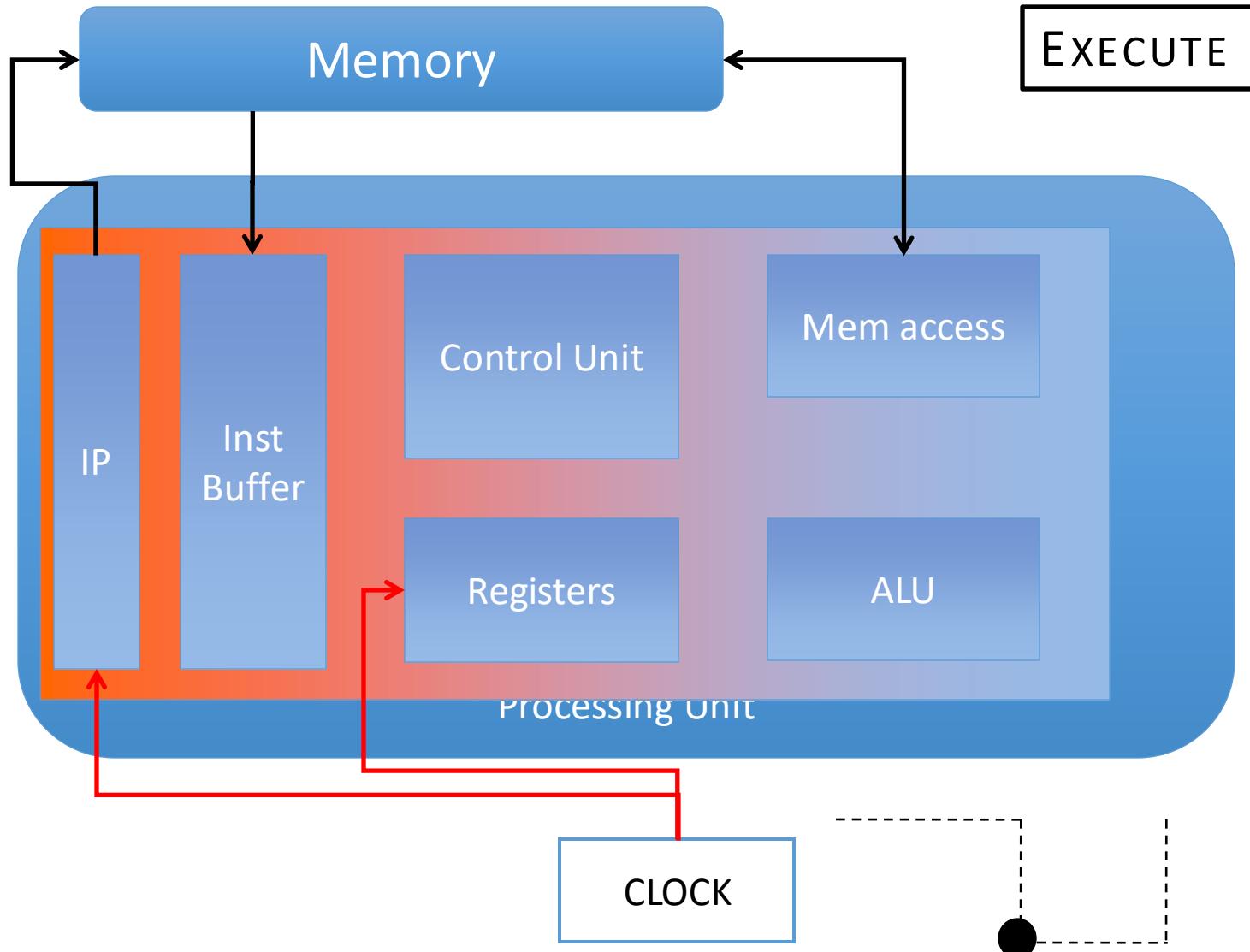


# Avaliação Desempenho

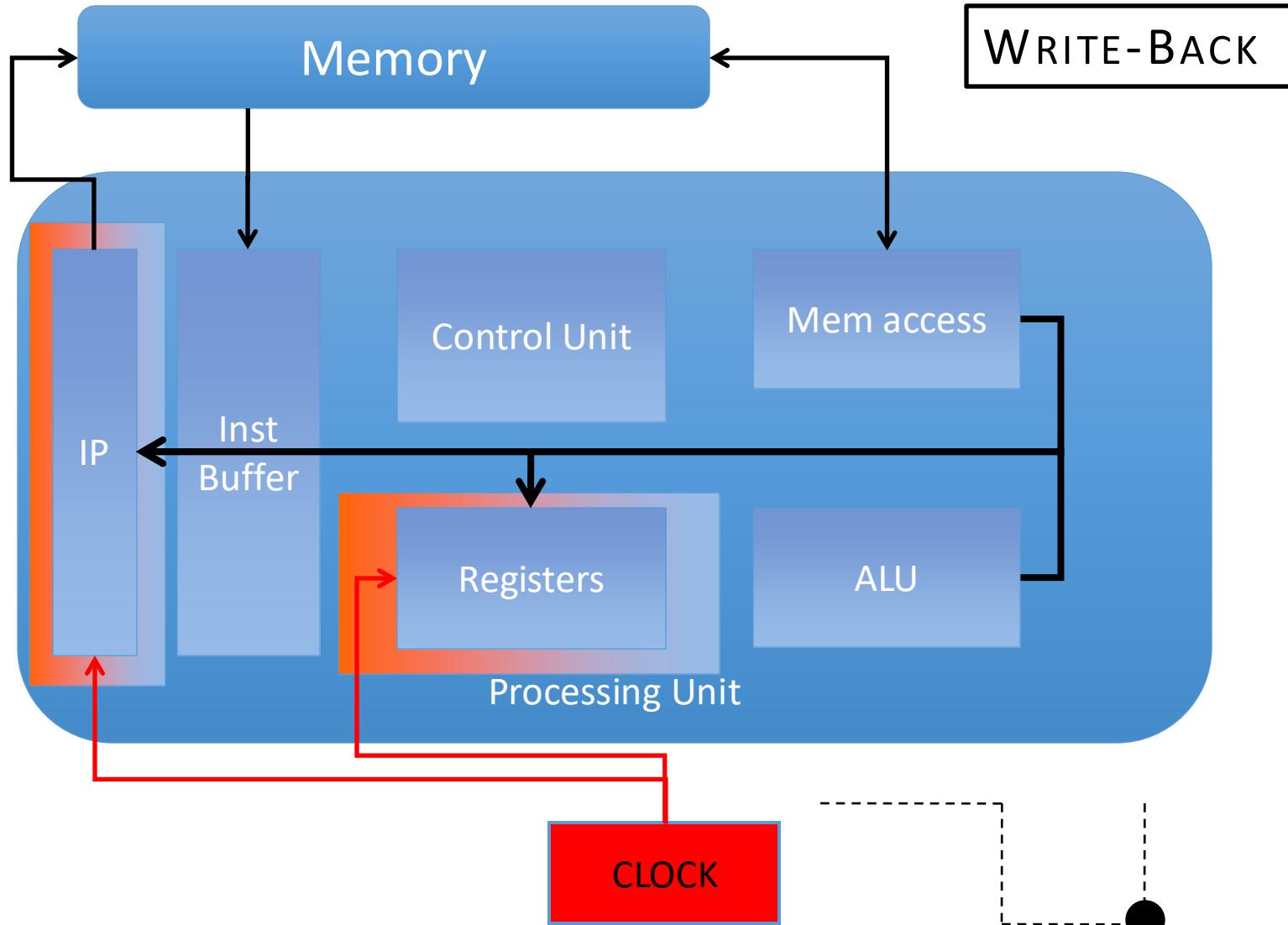


AC - Avaliação do Desempenho

# Avaliação Desempenho

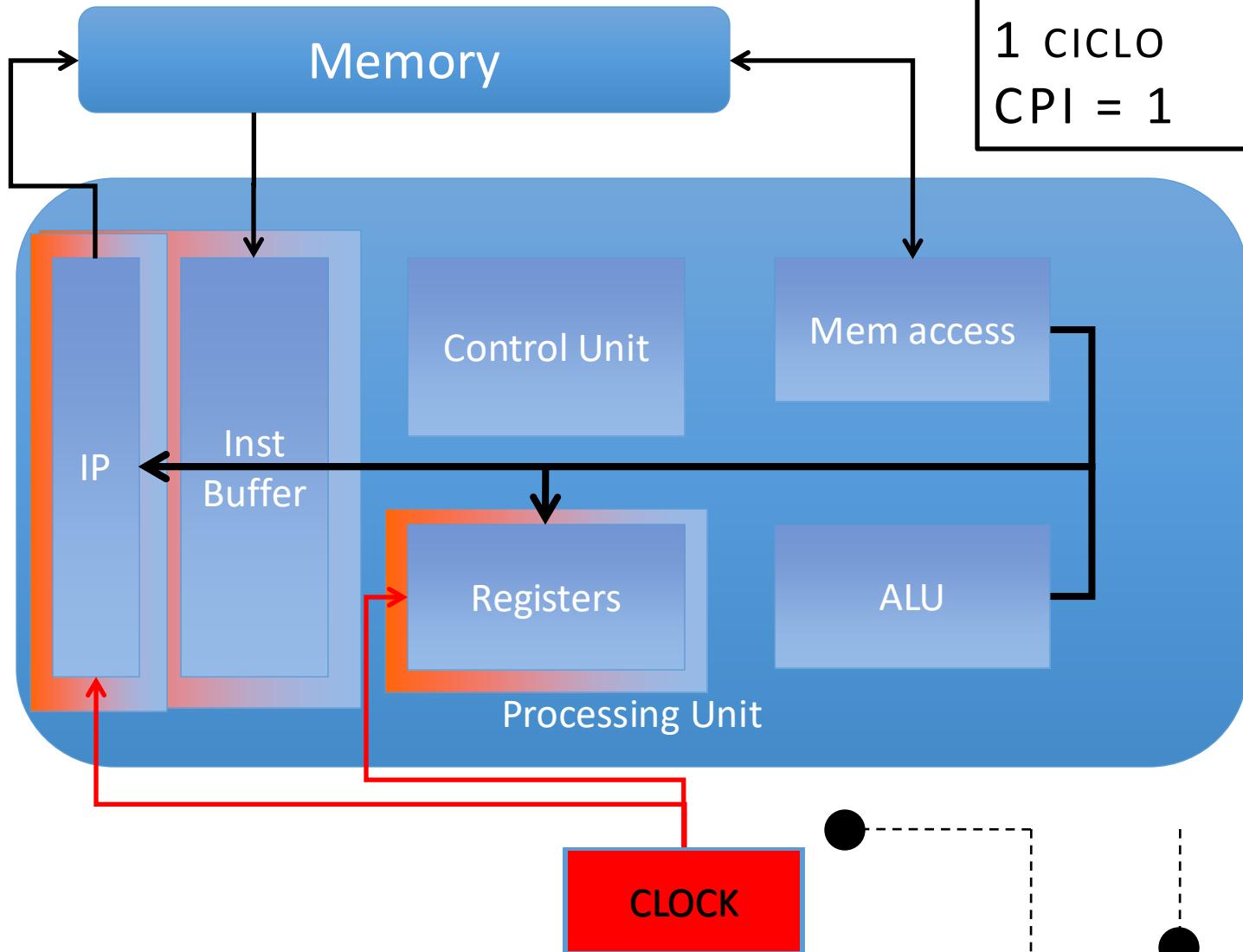


# Avaliação Desempenho



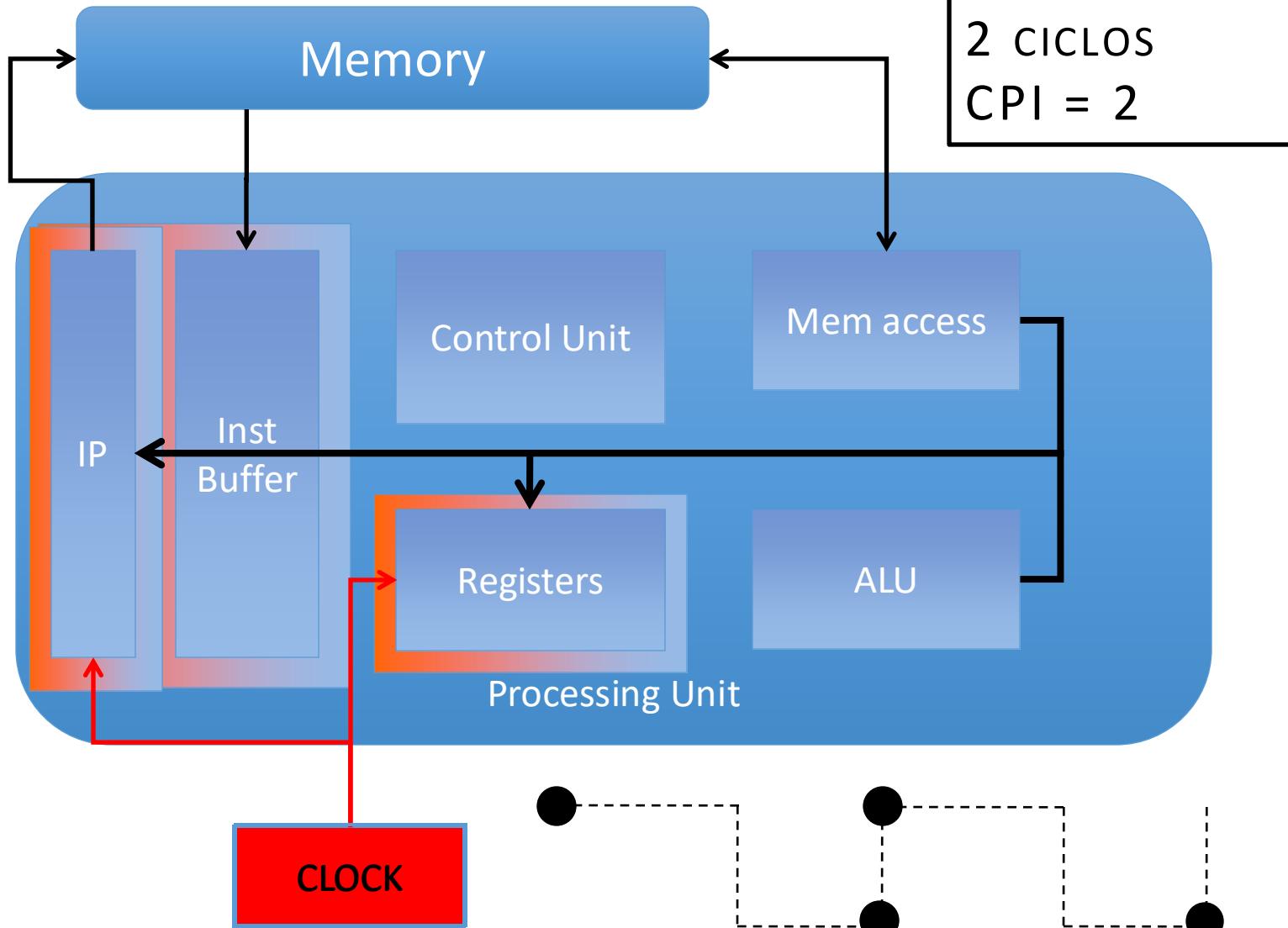
AC - Avaliação do Desempenho

# Avaliação Desempenho



AC - Avaliação do Desempenho

# Avaliação Desempenho



AC - Avaliação do Desempenho

# CPI – cycles per instruction

- Diferentes tipos de instruções exibem diferentes CPI:
  - CPI divisões > CPI adições
  - CPI acessos à memória > CPI acessos a registos
  - CPI vírgula flutuante  $\geq$  CPI inteiros
- A mesma instrução pode requerer um número de ciclos diferente para diferentes estados da máquina
- CPI é um valor médio

$$CPI = \frac{\#cc}{\#I}$$

- Pode ser medido com diferentes precisões.

# Desempenho do CPU

Previsão do tempo de execução ( $T_{EXEC}$ ) de um programa numa máquina - requer um **modelo** que relacione o desempenho com as características do sistema de computação ( $hw+sw$ )

Um programa numa máquina executa num determinado número médio de ciclos de relógio:  
**# clock cycles**

O período do relógio do CPU é constante:  
 $T_{cc} = 1 / f$

$$T_{EXEC} = \# \text{ clock cycles} * T_{cc}$$

# Desempenho do CPU

- De que depende o número médio de ciclos necessários para executar um programa?

Número médio de ciclos necessário para executar uma instrução:

CPI

número de instruções executadas de um programa:  
 $\#I$

$$\# \text{ clock cycles} = \text{CPI} * \#I$$

$$T_{EXEC} = \# \text{ clock cycles} * Tcc = \text{CPI} * \#I * Tcc = \text{CPI} * \#I / f$$

# Desempenho do CPU

$$T_{EXEC} = CPI * \#I / f$$

# Desempenho do CPU

- Um programador quer escolher entre dois segmentos de código diferentes para um mesmo algoritmo. Qual o mais rápido?

Tipo de Instrução	CPI
A	1
B	2
C	3

Código	Número de Instruções		
	A	B	C
1	2000	1000	100
2	100	1000	1000

$$\#I_1 = 2000 + 1000 + 100 = 3100$$

$$\#I_2 = 100 + 1000 + 1000 = 2100$$

$$T_{EXEC1} = \frac{(1*2000 + 2*1000 + 3*100)}{f} = 4300/f$$

$$T_{EXEC2} = \frac{(1*100 + 2*1000 + 3*1000)}{f} = 5100/f$$

$$Ganho = \frac{T_{EXEC2}}{T_{EXEC1}} = \frac{5100}{4300} = 1,186$$

# Desempenho do CPU

- Calcule o tempo de execução do programa abaixo numa máquina com um relógio de 2 GHz e CPI=1.5

```
    movl 10, %eax
    movl 0, %ecx
ciclo:
    addl %eax, %ecx
    decl %eax
    jnz ciclo
```

#I = 32

**NOTA:** número de instruções **executadas**.

$$T_{exec} = 32 * 1.5 / 2 E9 = 24 E-9 s = 24 ns$$

# Relação entre as métricas

$$T_{EXEC} = CPI * \#I / f$$

- $\#I$  – depende do algoritmo, do compilador e da arquitectura (ISA)
- CPI – depende da arquitectura (ISA), da mistura de instruções efectivamente utilizadas, da organização do processador e da organização dos restantes componentes do sistema (ex., memória)
- $f$  – depende da organização do processador e da tecnologia utilizada

**“A única métrica completa e fiável para avaliar o desempenho de um computador é o tempo de execução”**

As métricas CPI,  $f$  e  $\#I$  não podem ser avaliadas isoladamente, devendo ser sempre consideradas em conjunto, pois dependem umas das outras.

# Relação entre as métricas

**Exemplo 1 :** Aumentar  $f$  (diminuir Tcc) implica frequentemente um aumento do CPI!

**Explicação:** Se Tcc diminui, mas o tempo de acesso à memória (Tmem) se mantém, são necessários mais ciclos para aceder à memória.

$$f_1 = 1\text{GHz}$$

$$T_{cc1} = 1\text{ns}$$

$$T_{mem} = 40\text{ns}$$

$$Ciclos_{mem1} = 40$$

$$f_2 = 2\text{GHz}$$

$$T_{cc2} = 0.5\text{ns}$$

$$T_{mem} = 40\text{ns}$$

$$Ciclos_{mem2} = 80$$

**Conclusão:** Apesar de **Tcc diminuir para metade**, Texec não diminui para metade, pois o número de ciclos de acesso à memória aumenta, logo o **CPI aumenta**

# Relação entre as métricas

**Exemplo 2 :** Diminuir o número de instruções (#I) recorrendo a instruções mais complexas resulta num aumento do CPI!

**Explicação:** As instruções mais complexas realizam o trabalho de várias instruções simples, mas podem necessitar de mais ciclos para o completar, resultando num aumento do CPI.

Este é um dos argumentos dos defensores de arquitecturas RISC.

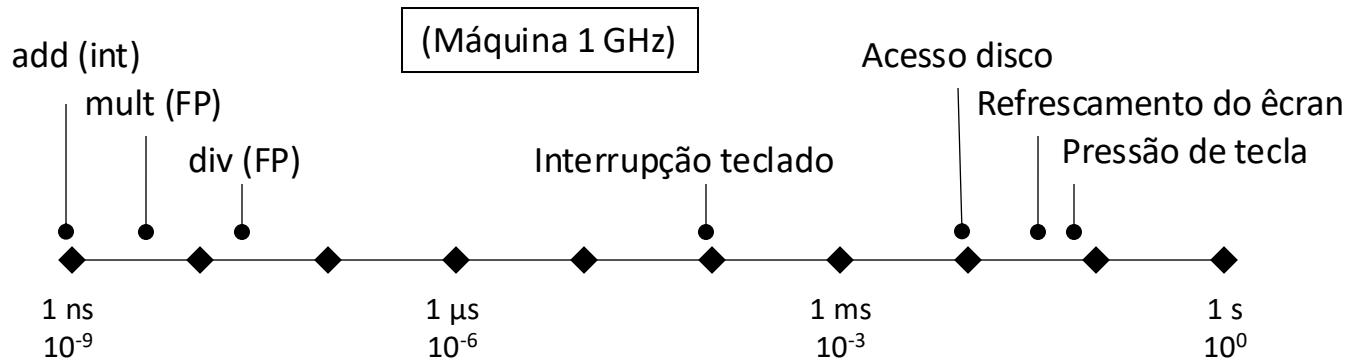
**Conclusão:** O número de instruções diminui, mas o ganho em tempo de execução não diminui na mesma proporção, devido ao aumento do CPI.

# Vox Vote

- Vá para:  
[live.voxvote.com](http://live.voxvote.com)
- Introduza o **PIN: 127845**

# Metodologia: Medição de Desempenho

```
before = ReadTimer();  
<<< Code Segment>>>  
after = ReadTimer();  
ElapsedTime = after - before;
```



- **Resolução** do relógio: unidade de tempo entre 2 incrementos do contador  
não é possível medir eventos com duração inferior à resolução
- **Precisão** do relógio: diferença entre o valor medido e o tempo efectivamente decorrido

# Metodologia: Medição de Desempenho

- Qual o tempo a medir?
  - *Wall Time*
    - Tempo decorrido desde o início até ao fim do programa
    - Depende da carga do sistema (E/S, outros processos,...)
  - Tempo de CPU
    - Tempo efectivamente dedicado a este processo
    - Menos sensível à carga do sistema

# Metodologia: Medição de Desempenho

Combinar o resultado de várias medições:

- Média das várias medições
  - Valores muito alto/baixos influenciam a média
  - Analisar também o desvio padrão (e.g., variações entre medições)
- Melhor medição
  - Valor obtido nas condições ideais
- Média das K-melhores medições
  - Média das k melhores execuções
- Mediana
  - Mais robusto a variações nas medições

# Metodologia: Medição de Desempenho

## Contadores de eventos

- Lógica incluída nos processadores (modernos) para contagem de eventos específicos
- Actualizados a cada ciclo de relógio
- Vantagens:
  - Não intrusivos / baixa sobrecarga (disponibilizados pelo hardware)
  - Elevada resolução (relógio do processador)
- Pontos fracos:
  - Específicos de cada processador / não existe um “standard”
  - Não são apropriados para serem usados por utilizador “comum”

# Metodologia: Medição de Desempenho

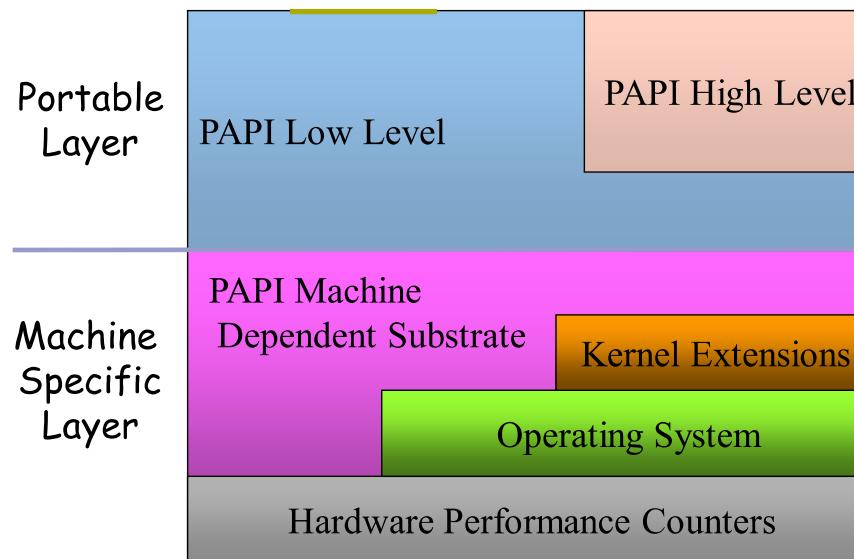
## Contadores de desempenho

- Eventos típicos
  - Ciclos de relógio
  - Número de instruções
  - Instruções de vírgula flutuante
  - Instruções sobre valores inteiros (add, sub, etc)
  - *Load/stores*
  - *Cache misses* (L1, L2, etc)

# Metodologia: Medição de Desempenho

## Performance Application Programming Interface

- Interface para acesso aos contadores de desempenho
- Inclui rotinas para contagem de tempo e para obter informação sobre o sistema
- <http://icl.cs.utk.edu/papi/>
- Arquitectura:



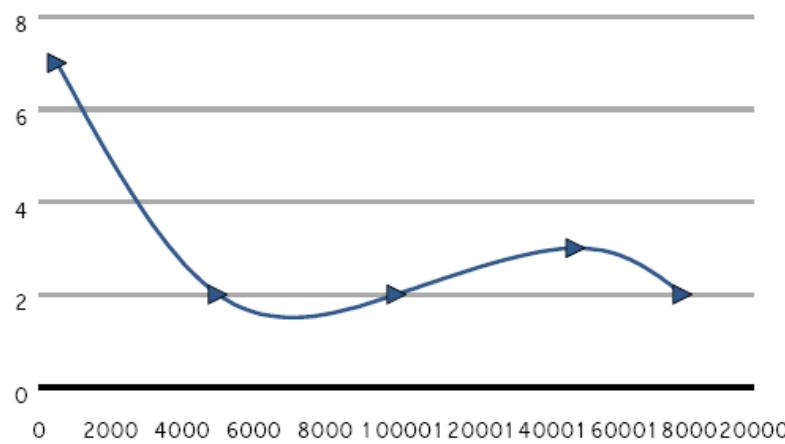
# Metodologia: Medição de Desempenho

## Apresentação dos resultados

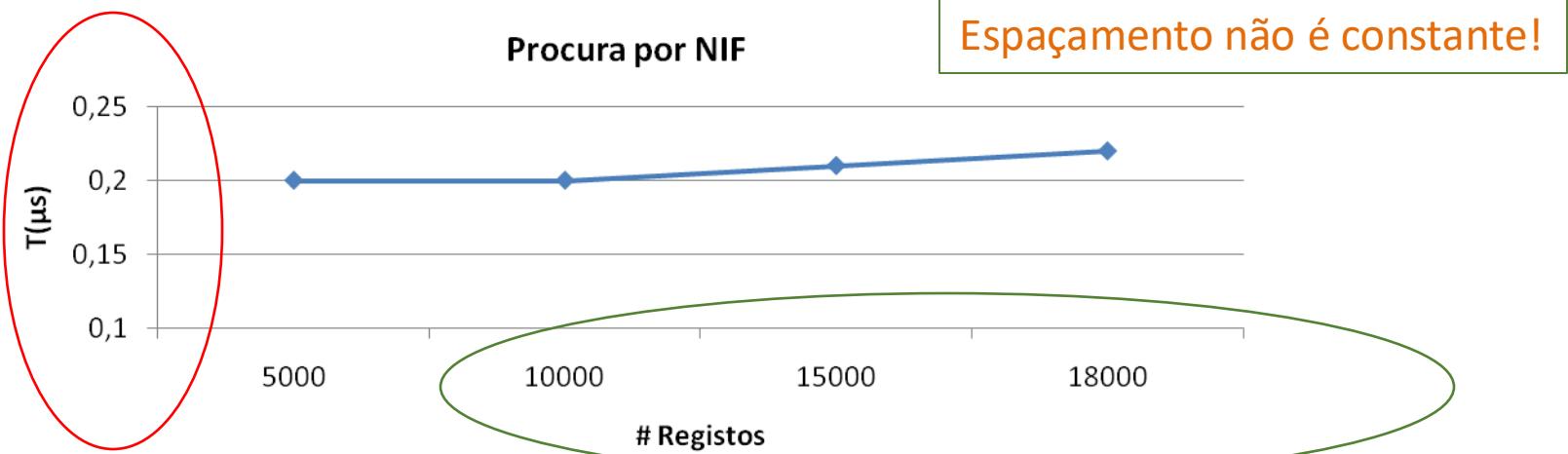
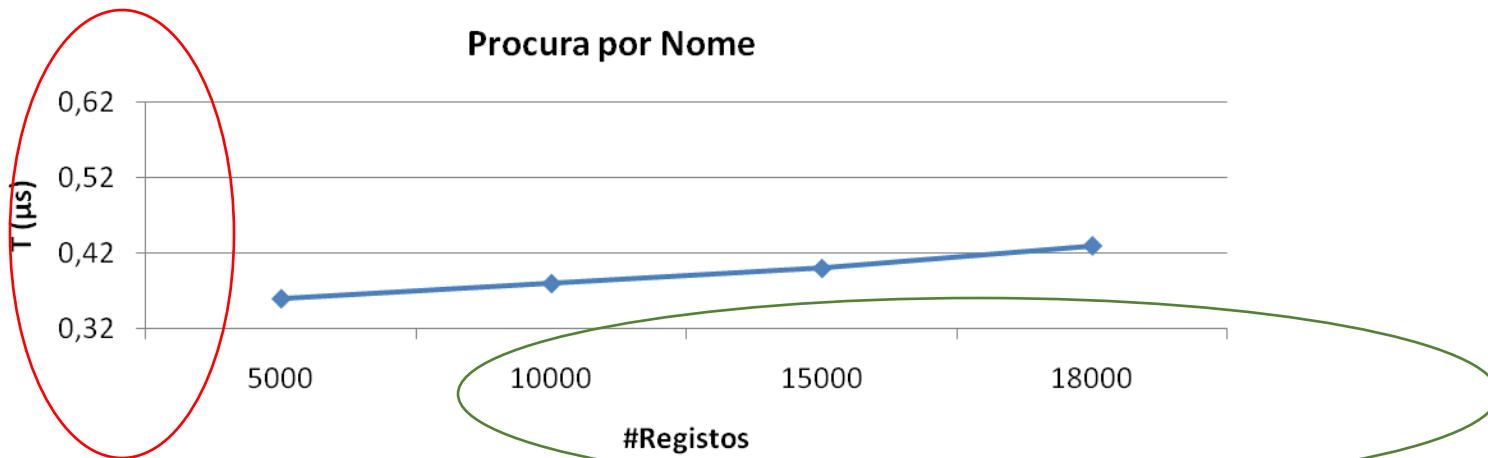
- Apresentar os resultados de forma compacta

Operações	Tempos de Execução			
	5000	10000	15000	18000
Carregar Dados	10.019 ms	20.881 ms	32.027 ms	40.992 ms
Inserir Cliente	7.100 $\mu$ s	7.400 $\mu$ s	8.800 $\mu$ s	9.500 $\mu$ s
Procura por Nome	0.360 $\mu$ s	0.380 $\mu$ s	0.400 $\mu$ s	0.430 $\mu$ s
Procura por Nif	0.020 $\mu$ s	0.020 $\mu$ s	0.020 $\mu$ s	0.020 $\mu$ s
Percorrer Estrutura	0.092 ms	0.232 ms	0.470 ms	0.673 ms

- Colocar legendas nas tabelas e gráficos

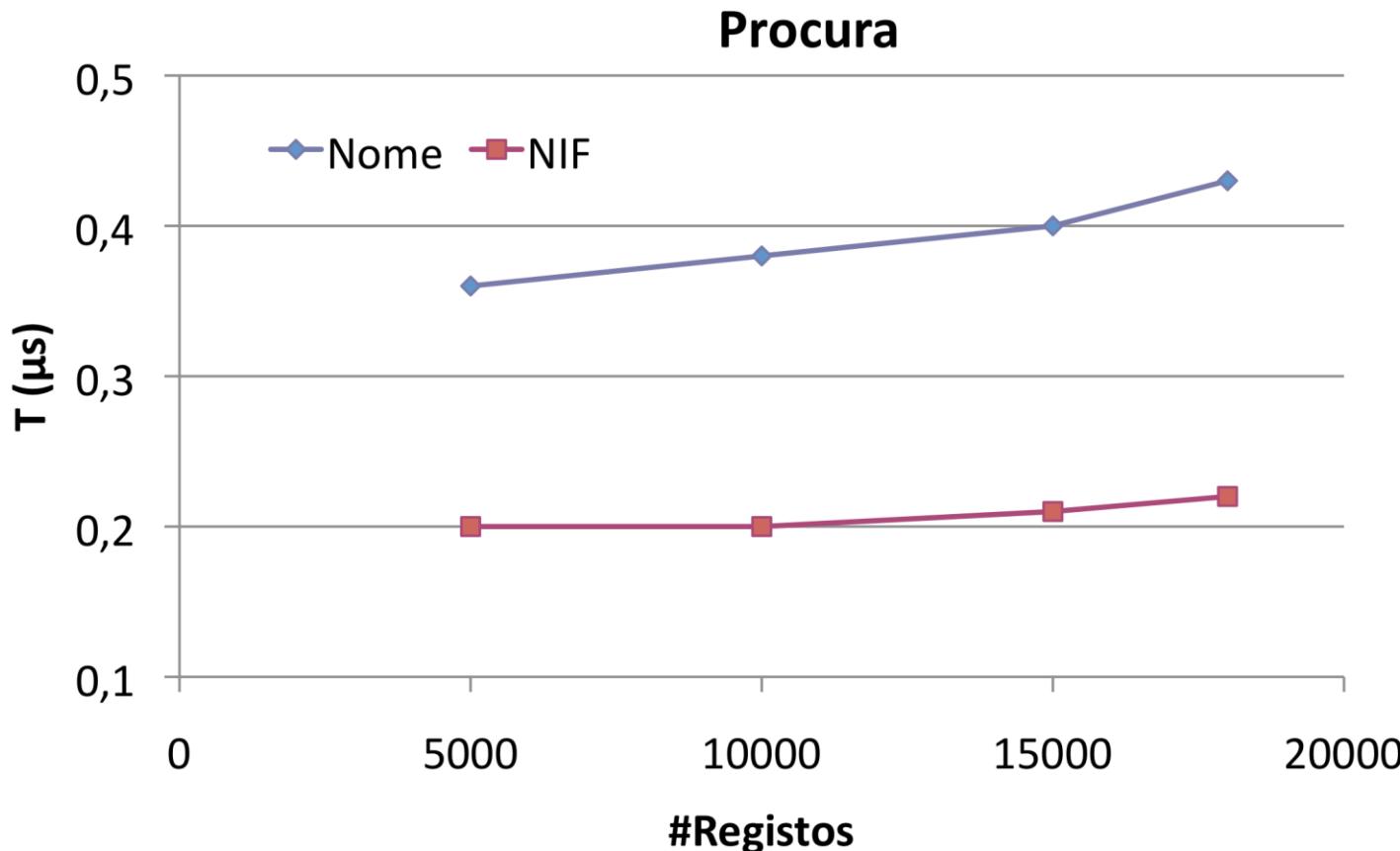


# Metodologia: Medição de Desempenho



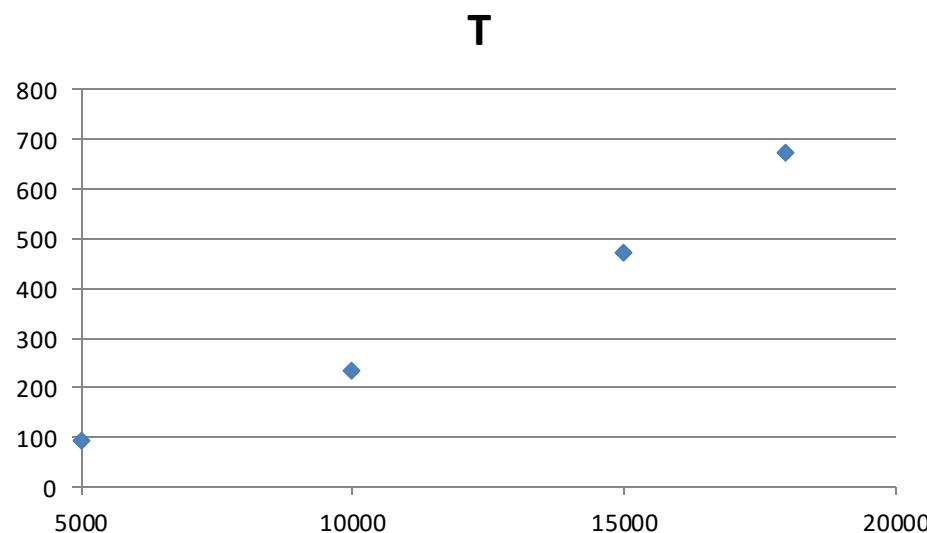
Escalas diferentes em comparações directas induzem em erro!

# Metodologia: Medição de Desempenho



# Complexidade – curve fitting

Tempo de Execução ( $\mu$ s)				
	Nº de registos			
Operação	5000	10000	15000	18000
Percorrer Estrutura	92.00	232.00	470.00	673.00



O processo de curve fitting permite determinar a equação da curva que melhor descreve os dados.

# Complexidade – curve fitting

A opção “Trendline” do MS EXCEL determina a equação da curva dado um modelo: linear, polinomial, logarítmico, exponencial, etc.

O parâmetro R2 descreve a qualidade do fitting.

Quanto mais perto de 1 melhor.

