

# INE5607 – Organização e Arquitetura de Computadores

Unidade Central de Processamento

Aula 15: Processadores monociclo

Prof. Laércio Lima Pilla

laercio.pilla@ufsc.br



# Sumário

- Revisão de desempenho
- Componentes de um processador
- Instruções em um processador
- Processador monociclo
- Considerações finais

# REVISÃO DE DESEMPENHO

# Revisão de desempenho

- Desempenho relativo

— **X** é ***n* vezes mais rápido** do que **Y**

$$n = \frac{\text{Tempo de execução}_y}{\text{Tempo de execução}_x} = \frac{\text{Desempenho}_x}{\text{Desempenho}_y}$$

- Tempo

$$\text{tempo} = \frac{\text{segundos}}{\text{programa}} = \frac{\text{instruções}}{\text{programa}} \times \frac{\text{ciclos}}{\text{instrução}} \times \frac{\text{segundos}}{\text{ciclo}}$$

# Revisão de desempenho

- Exemplo
  - Programa compilado para uma certa ISA
  - **Computador Super Proc**
    - Tempo de ciclo de 30 ns, CPI de 3
  - **Computador Master Comp**
    - Tempo de ciclo de 40 ns, CPI de 2.5
  - **Computador Über Calc**
    - Tempo de ciclo de 25 ns, CPI de 3.5
  - Qual é o mais rápido? Por quanto?

# Revisão de desempenho

- Dadas duas sequências de código abaixo, qual executa mais rápido e o quanto mais rápido?

Sequências de código	Instruções tipo A	Instruções tipo B	Instruções tipo C
Código X	10	5	20
Código Y	15	10	15

	Instruções tipo A	Instruções tipo B	Instruções tipo C
CPI	3	5	4

# COMPONENTES DE UM PROCESSADOR

# Componentes de um processador

- Principais componentes dentro de um processador

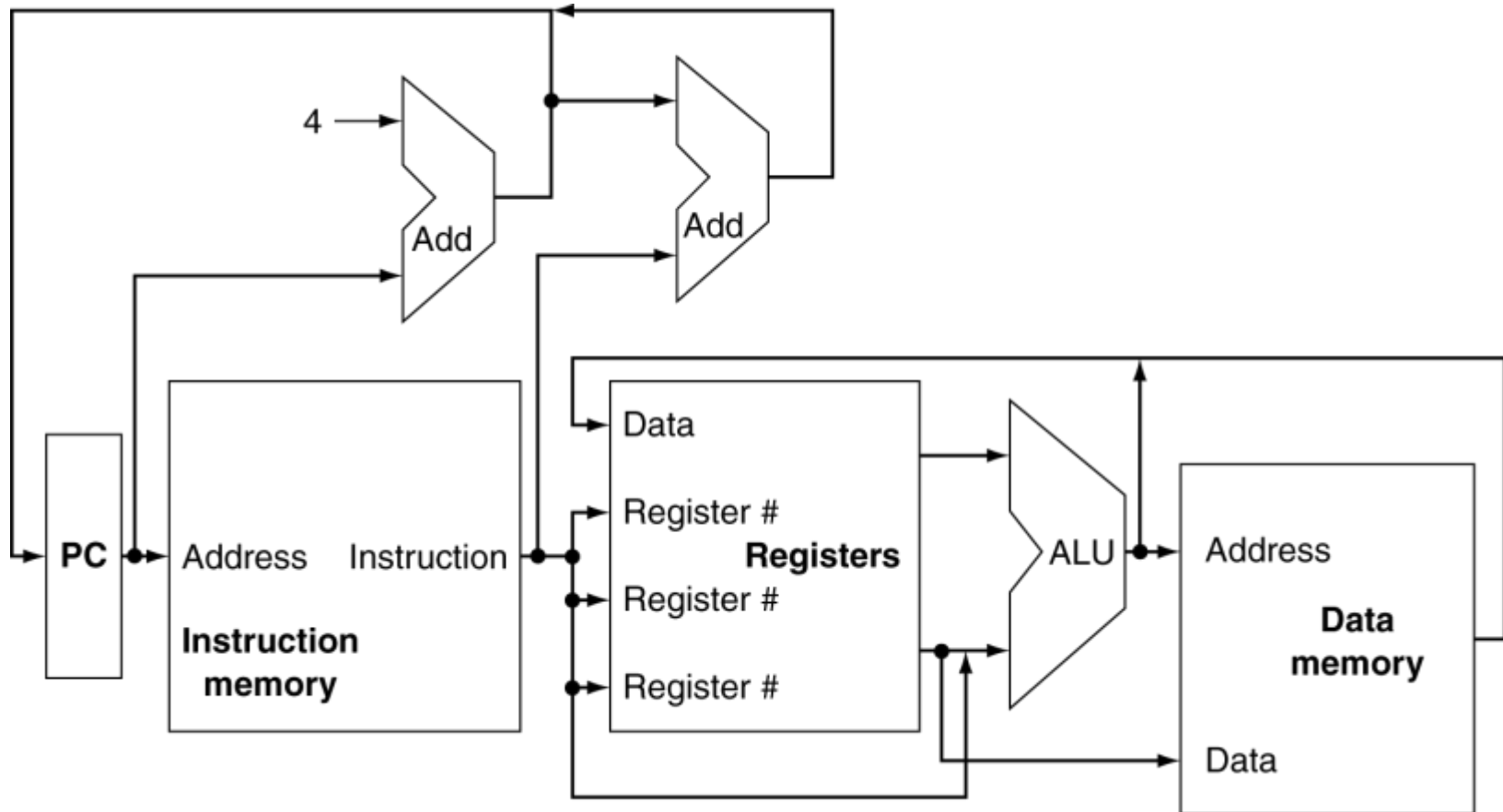


Figura 4.1 do livro *Computer Organization and Design 4th ed.*



# Componentes de um processador

- **Principais componentes dentro de um processador**
  - **Datapath** + Controle

PC

Memória de  
Instruções

Banco de  
Registradores

ULA

Memória de  
Dados

# Componentes de um processador

- **PC:** *Program counter*
  - Necessário para dizer qual é a próxima instrução a ser executada
    - **Execução de todas instruções dependem e influenciam o PC!**
  - MIPS32
    - $PC = PC + 4$  para instruções simples
    - $PC = \text{endereço de desvio}$  para desvios



# Componentes de um processador

- **Memória de instruções**

- Armazena as instruções do programa
  - Endereçada pelo PC
- **Memória de instruções precisa ser lida para cada instrução**
- MIPS32
  - Leitura de uma palavra = uma instrução



Memória de  
Instruções

A dark blue square box with the text "Memória de Instruções" in white, representing the instruction memory component of a processor.

# Componentes de um processador

- **Banco de registradores**

- Contém os registradores usados para a execução de instruções
  - \$s0-7, \$t0-9, \$zero, \$sp, etc.
- Registradores podem ser lidos e escritos
  - add \$t0, \$t1, \$t2 lê \$t1 e \$t2 e escreve em \$t0
- MIPS32
  - Máximo leitura de 2 registradores
  - Máximo escrita em 1 registrador

**Banco de  
Registradores**

# Componentes de um processador

- **ULA:** Unidade Lógica e Aritmética
  - Realiza os cálculos exigidos pelas instruções
    - Soma (valores ou endereços), subtração, deslocamento, operações bit a bit, etc.
  - MIPS32
    - Dois valores são usados como operandos
    - Um valor é gerado como resultado



# Componentes de um processador

- **Memória de dados**

- Mantém todos os dados de um programa
  - **Variáveis, vetores, estruturas de dados, etc.**
- Valores podem ser lidos e escritos
  - Não escrevemos na memória de instruções!
- MIPS32
  - Load: usa endereço para ler valor
  - Store: usa endereço para escrever valor

**Memória de  
Dados**

# INSTRUÇÕES EM UM PROCESSADOR

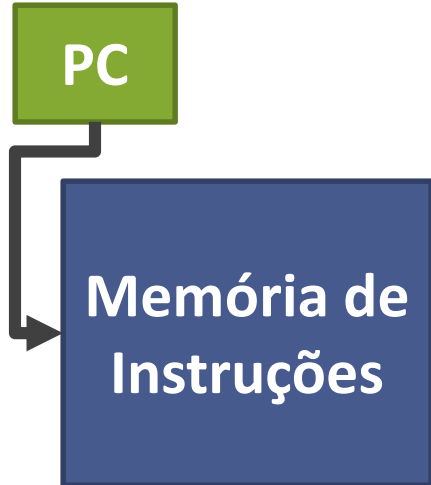
# Instruções em um processador

- **Como instruções são executadas?**
  - Exemplos
    - `add $s0, $s1, $s2`



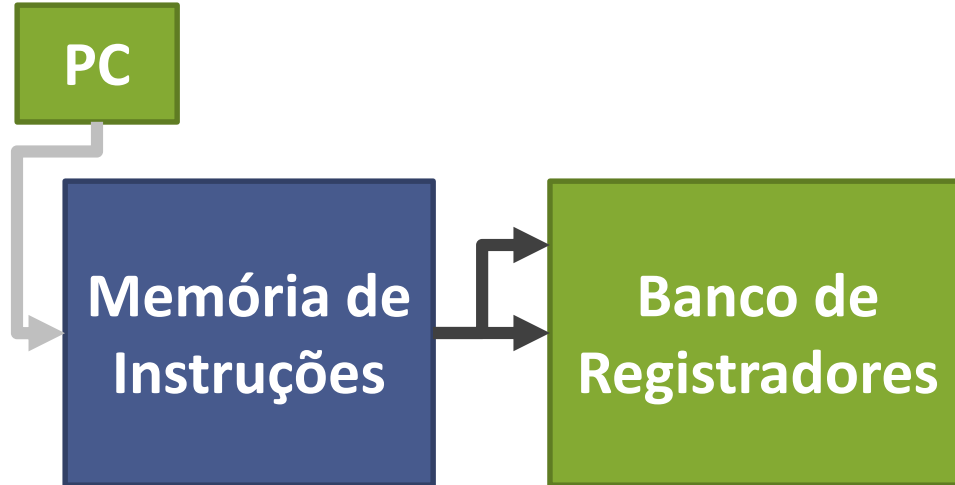
# Instruções em um processador

- **add \$s0, \$s1, \$s2**
  - Busca da instrução
    - Ler da memória e interpretar o que deve ser feito



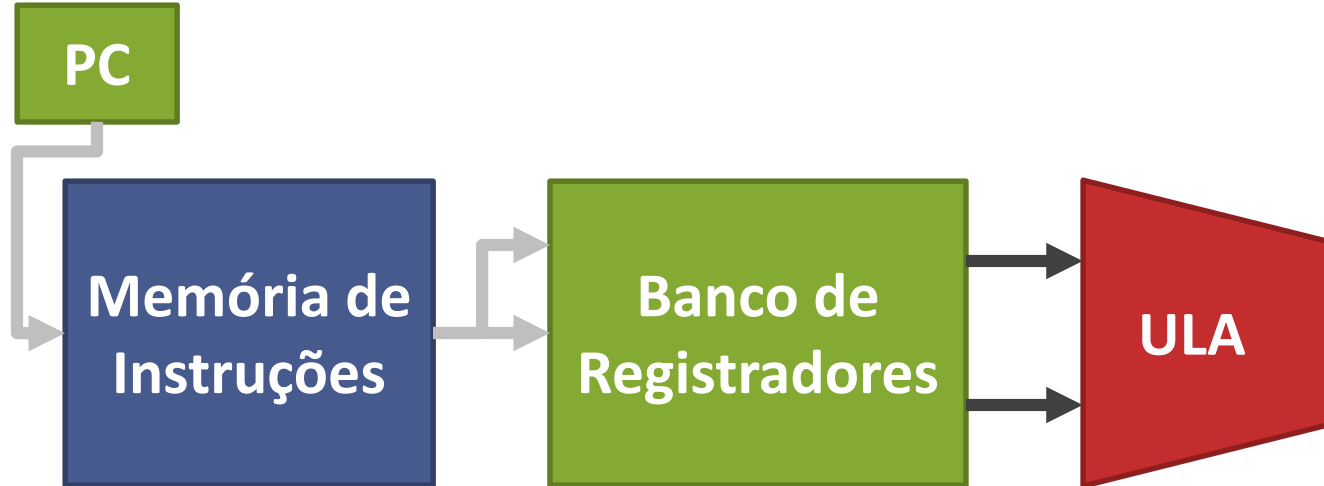
# Instruções em um processador

- add \$s0, \$s1, \$s2
  - Leitura dos registradores
    - Ler \$s1 e \$s2 para poder somá-los



# Instruções em um processador

- add \$s0, \$s1, \$s2
  - Operação: Unidade Lógica e Aritmética (ULA)
    - Somar \$s1 e \$s2



# Instruções em um processador

- add \$s0, \$s1, \$s2
  - Escrita do registrador
    - Salvar o resultado da soma no registrador \$s0



# Instruções em um processador

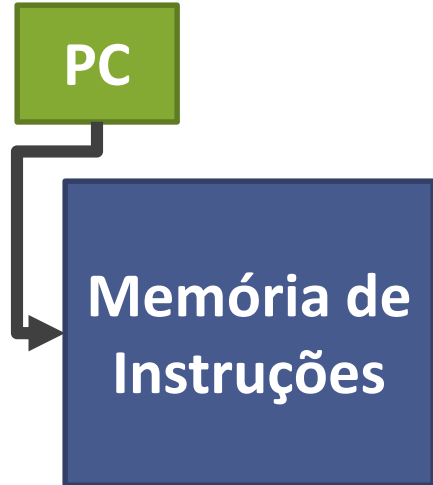
- Como instruções são executadas?
  - Exemplos
    - `add $s0, $s1, $s2`
    - `sw $t1, 24($s0)`

# Instruções em um processador

- **sw \$t1, 24(\$s0)**

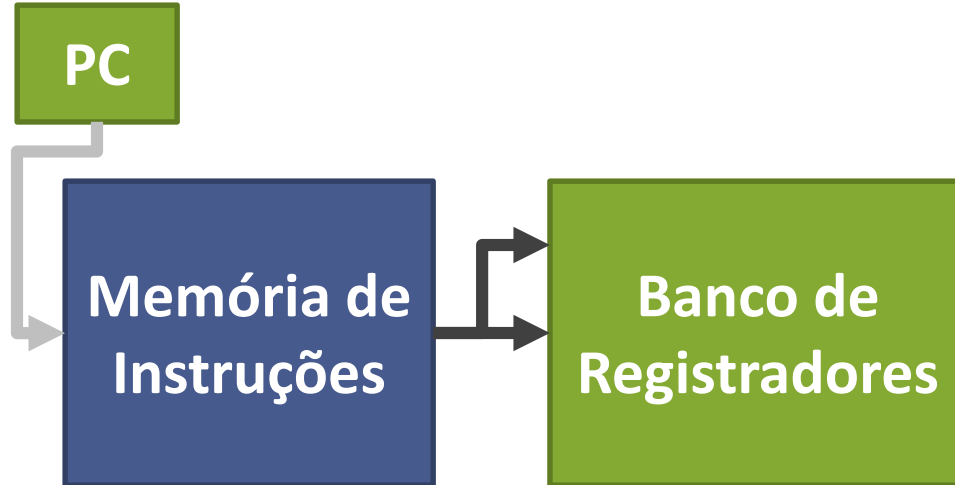
- Busca da instrução

- Ler da memória e interpretar o que deve ser feito



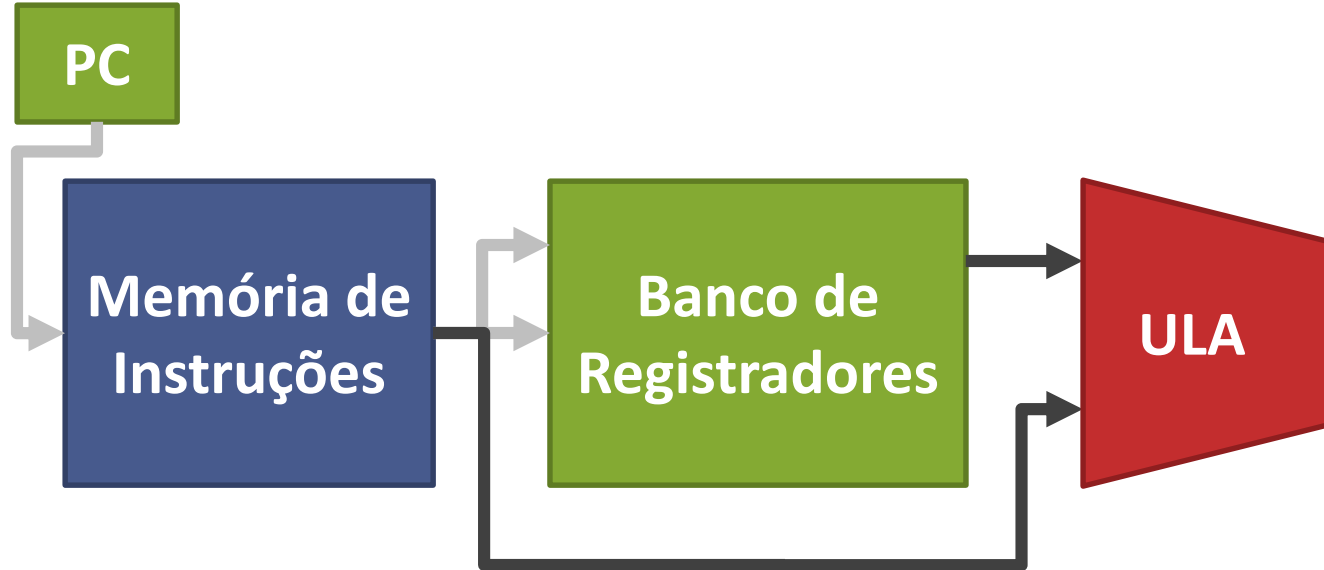
# Instruções em um processador

- `sw $t1, 24($s0)`
  - Leitura dos registradores
    - Ler `$s0` para somar com 24, `$t1` para armazenar



# Instruções em um processador

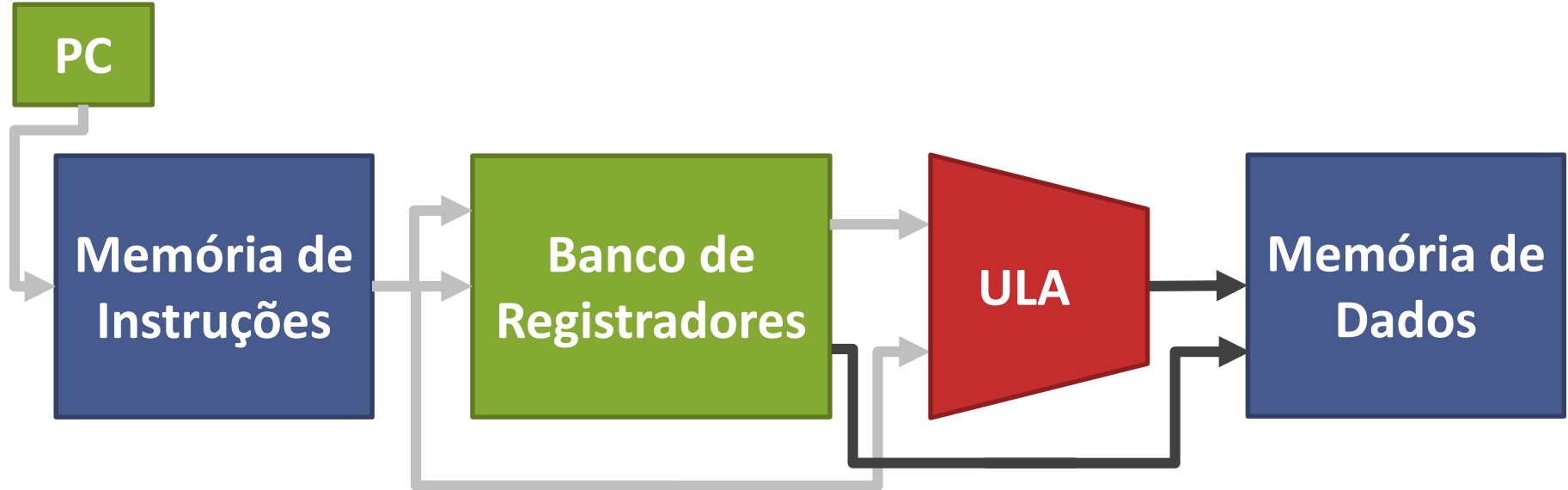
- `sw $t1, 24($s0)`
  - Operação: Unidade Lógica e Aritmética (ULA)
    - Somar `$s0` e 24 para calcular o endereço





# Instruções em um processador

- `sw $t1, 24($s0)`
  - Acesso à memória de dados
    - Armazenar o valor de `$t1` em `$s0+24`

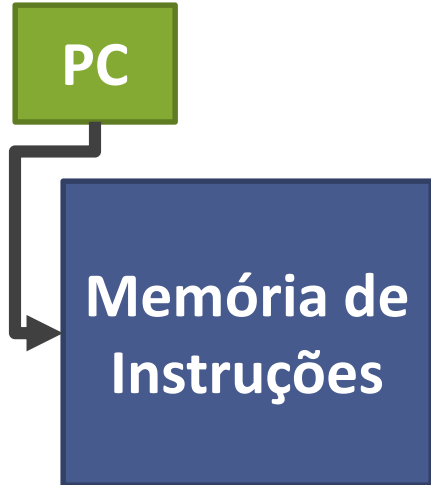


# Instruções em um processador

- Como instruções são executadas?
  - Exemplos
    - add \$s0, \$s1, \$s2
    - sw \$t1, 24(\$s0)
    - lw \$t0, 20(\$s0)

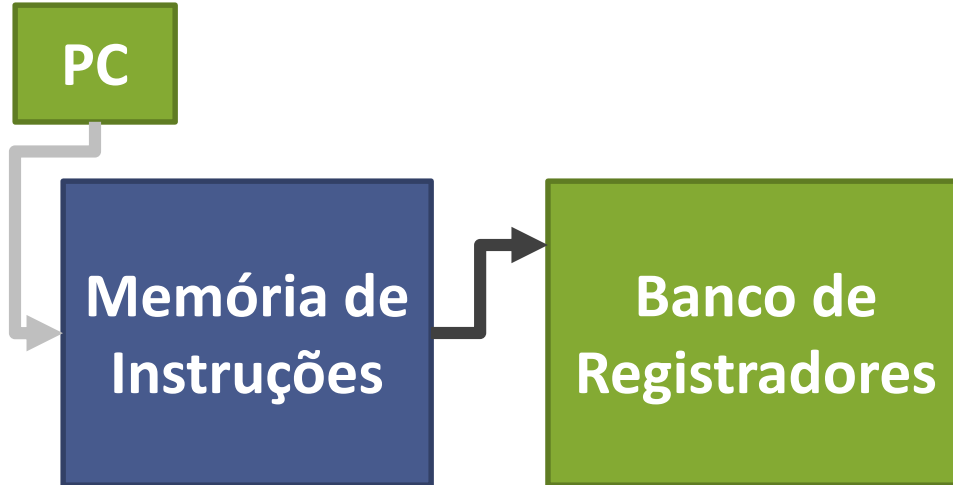
# Instruções em um processador

- **lw \$t0, 20(\$s0)**
  - Busca da instrução
    - Ler da memória e interpretar o que deve ser feito



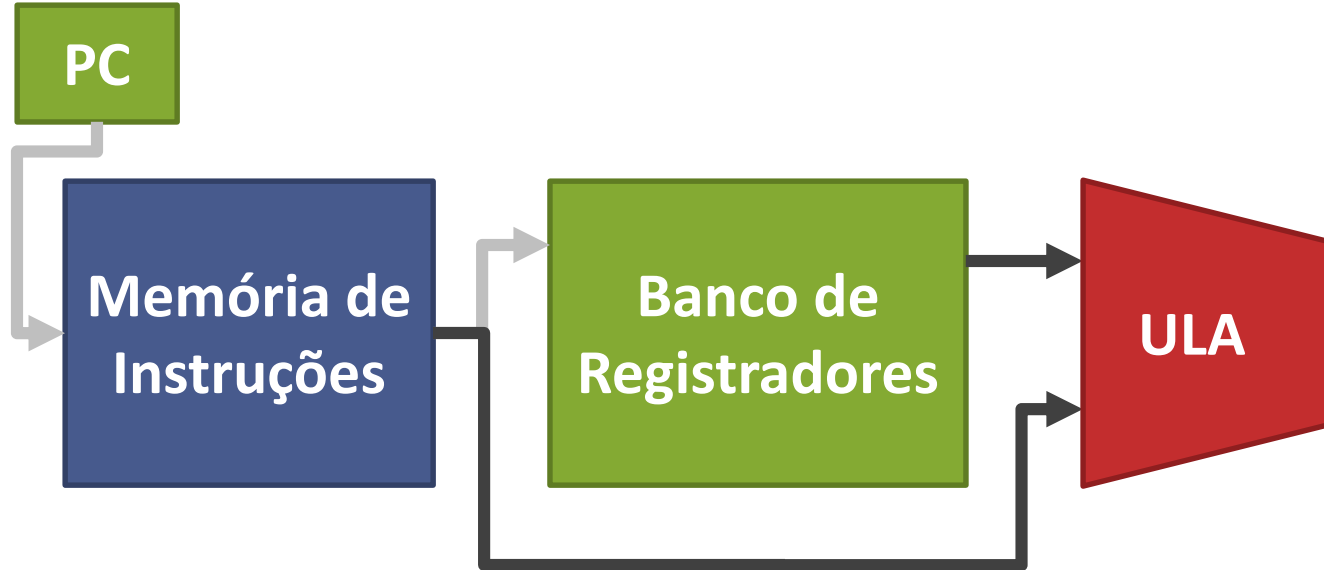
# Instruções em um processador

- `lw $t0, 20($s0)`
  - Leitura dos registradores
    - Ler `$s0` para somar com 20



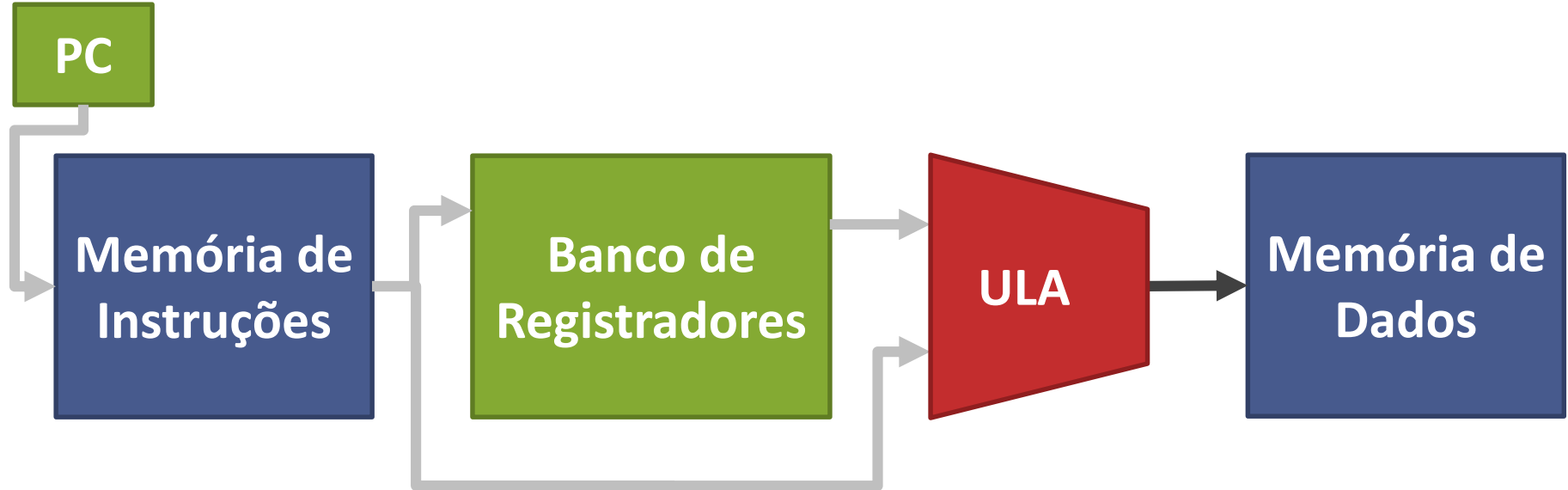
# Instruções em um processador

- `lw $t0, 20($s0)`
  - Operação: Unidade Lógica e Aritmética (ULA)
    - Somar `$s0` e 20 para calcular o endereço



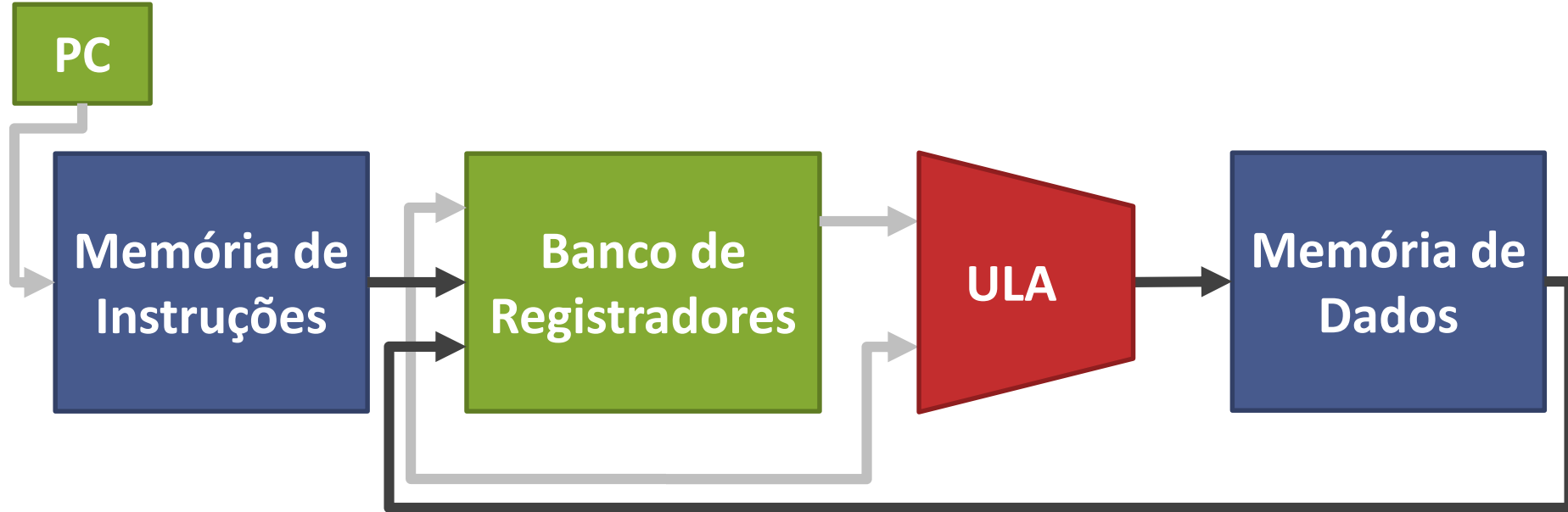
# Instruções em um processador

- `lw $t0, 20($s0)`
  - Acesso à memória de dados
    - Ler o valor em  $\$s0+20$



# Instruções em um processador

- `lw $t0, 20($s0)`
  - Escrita do registrador
    - Armazenar em `$t0` o valor lido em `$s0+20`



# Instruções em um processador

- Como instruções são executadas?

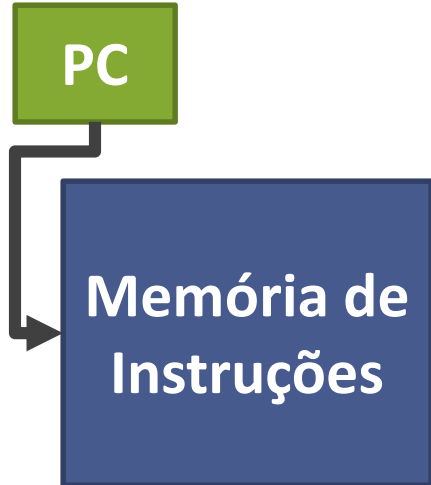
- Exemplos

- add \$s0, \$s1, \$s2
    - sw \$t1, 24(\$s0)
    - lw \$t0, 20(\$s0)
    - beq \$t0, \$zero, BLA



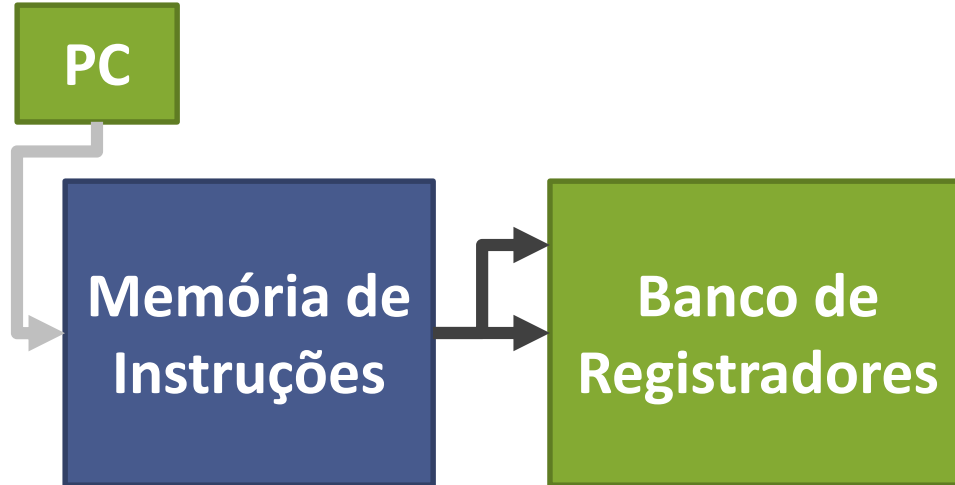
# Instruções em um processador

- **beq \$t0, \$zero, BLA**
  - Busca da instrução
    - Ler da memória e interpretar o que deve ser feito



# Instruções em um processador

- beq \$t0, \$zero, BLA
  - Leitura dos registradores
    - Ler \$t0 e \$zero para fazer a comparação



# Instruções em um processador

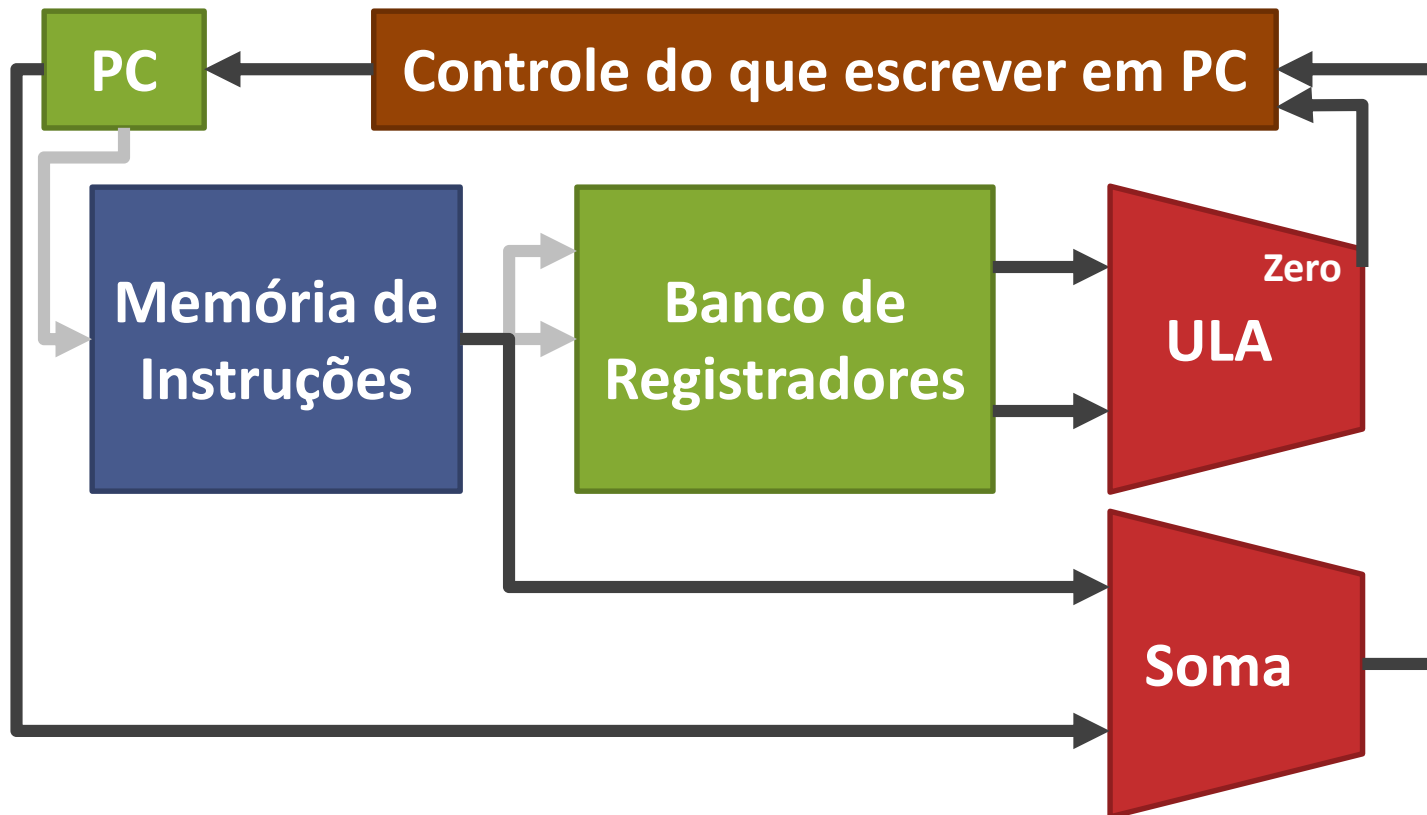
- beq \$t0, \$zero, BLA
  - Operação: Unidade Lógica e Aritmética (ULA)
    - Comparar \$t0 e \$zero



- **Como é feito? \$t0 - \$zero**
  - Se iguais: Sinal Zero ligado
  - Se diferentes: Sinal Zero desligado

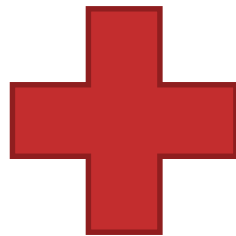
# Instruções em um processador

- beq \$t0, \$zero, BLA
  - Operação na ULA + Escrita em PC
    - Se os valores são iguais, altera PC usando BLA



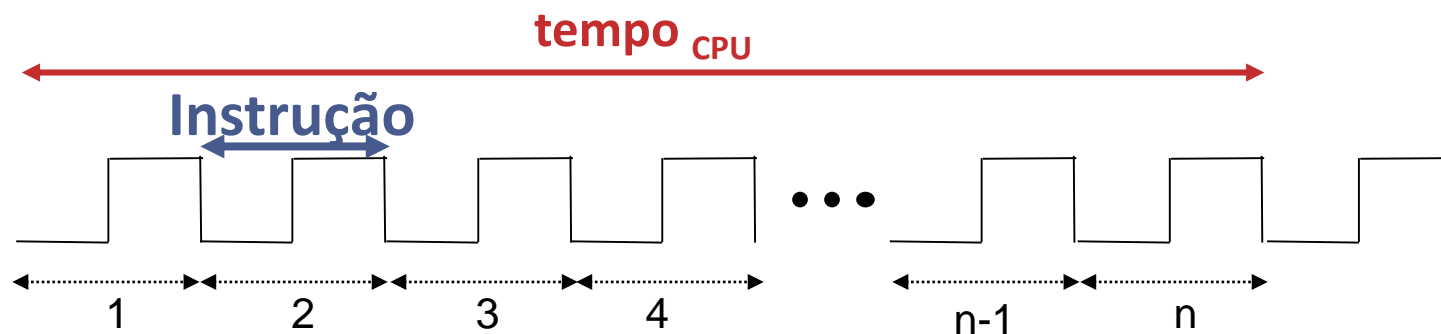
# PROCESSADOR MONOCICLO

# Processador monociclo



# Processador monociclo

- Principal característica
  - Instruções levam **um ciclo de relógio** para serem executadas
    - **Tudo** tem que ser feito **dentro do ciclo**
    - **Tempo** de ciclo depende da **instrução mais lenta**
    - O **ciclo** de relógio tem **duração fixa**



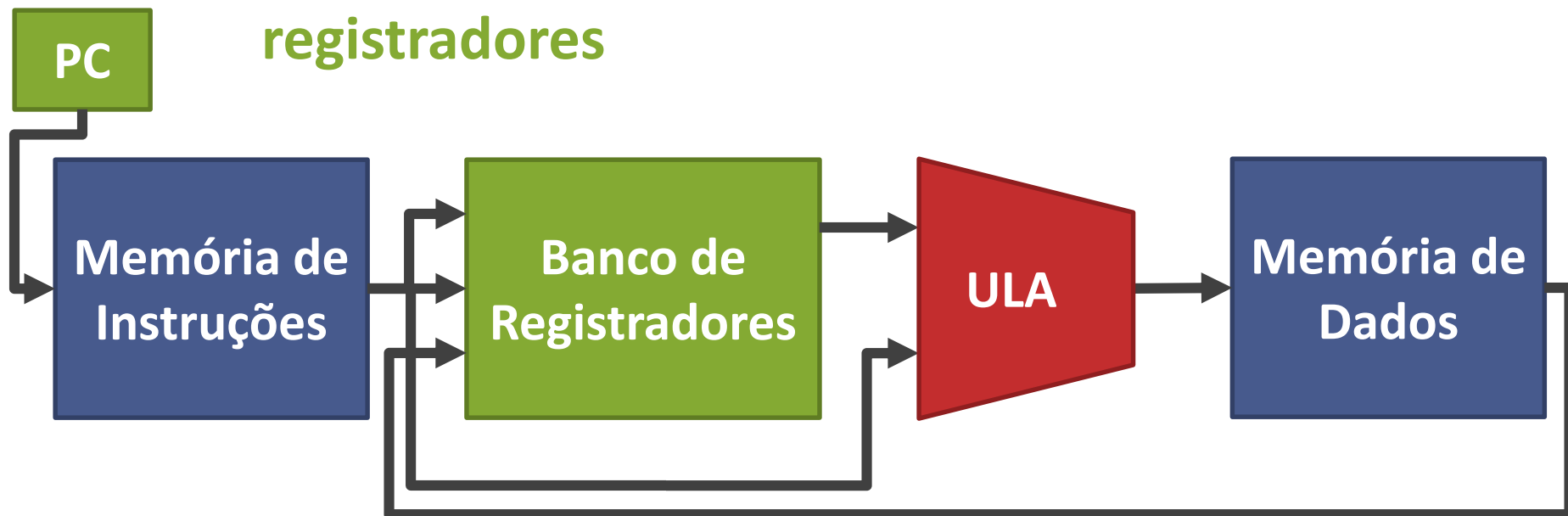
# Processador monociclo

- Qual das instruções é a mais lenta?
  - add \$s0, \$s1, \$s2
  - sw \$t1, 24(\$s0)
  - lw \$t0, 20(\$s0)
  - beq \$t0, \$zero, BLA



# Processador monociclo

- `lw $t0, 20($s0)`
  - Cinco estágios
    - **Memória**, **registradores**, **ULA**, **memória**, **registradores**



# Processador monociclo

- Exemplo
  - Dado um processador com os componentes abaixo, qual seria sua frequência de relógio?
    - Memórias de 200 ps
    - Banco de registradores de 50 ps
    - ULA de 100 ps
    - Outros componentes desprezíveis
  - Quanto tempo levaria uma soma nesse processador?

# Processador monociclo

- Exemplo

- Dado um processador monociclo de 100 MHz e um programa de 10.000.000 instruções, quanto **tempo** leva sua execução?
- Qual o **CPI** do processador?

# Processador monociclo

- Exemplo
  - Considerando os dois programas abaixo, qual deles executa mais rápido em um processador monociclo?
    - **Programa A**
      - $10^8$  instruções, sendo 40% aritméticas, 20% branches, 20% loads e 20% stores
    - **Programa B**
      - $10^8$  instruções, sendo 60% aritméticas, 10% branches, 20% loads e 10% stores

# Processador monociclo

- Exemplo
  - Um processador será construído com componentes que farão as seguintes instruções levarem os seguintes tempos:
    - **Soma: 10 ns**
    - **Subtração: 10 ns**
    - **Leitura de memória: 18 ns**
    - **Multiplicação: 50 ns**
  - Qual será a frequência de relógio do processador?

# Processador monociclo

- Problema
  - **Ineficiência**: a frequência de relógio depende da instrução mais lenta
    - E se a **memória** é muito **lenta**?
    - E se eu quero uma **operação de raiz quadrada**?
    - O que acontece com **instruções mais curtas**?
    - Onde fica “**tornar o caso comum mais rápido**”?

# CONSIDERAÇÕES FINAIS

# Considerações finais

- Resumo
  - Como instruções são executadas
  - Processador monociclo
    - Toda a instrução em apenas um ciclo
    - Ineficiente



# INE5607 – Organização e Arquitetura de Computadores

Unidade Central de Processamento

Aula 15: Processadores monociclo

Prof. Laércio Lima Pilla

laercio.pilla@ufsc.br

