

# Laboratório de Arquitetura de Computadores



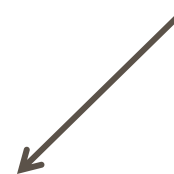
Profa. Yeda

Aula 2 – Introdução à Arquitetura de Processadores

# Introdução

- Linguagem de alto nível x linguagem assembly x linguagem de máquina: (P1.2\*)
  - $C = A + B$     linguagem C
  - **add** C, A, B    linguagem assembly
  - 000000000011000010001000000100000    ling.Máquina
- Princípio da simplicidade e regularidade: operações simples, básicas e padronizadas (P2.2\*)
  - Ex.: soma e subtração sempre possuem 3 operandos
  - Então como faz  $D = A + B + C$  ?
    - add D, A, B
    - add D, D, C

00611020 Hexa



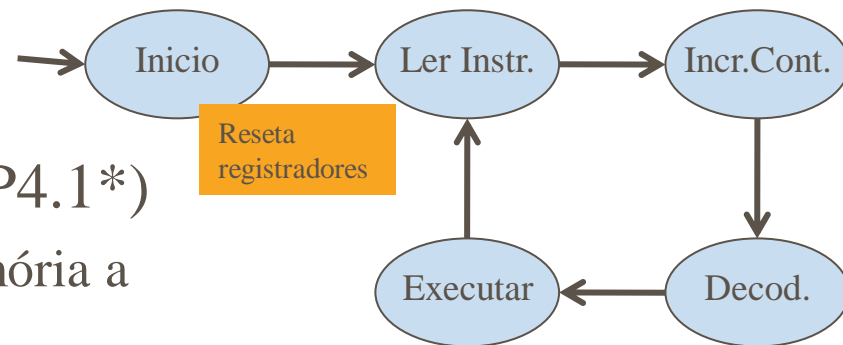
# Introdução

## ■ Organização básica de computadores

- processador (ULA + controle), memória e I/O (P4.1\*)

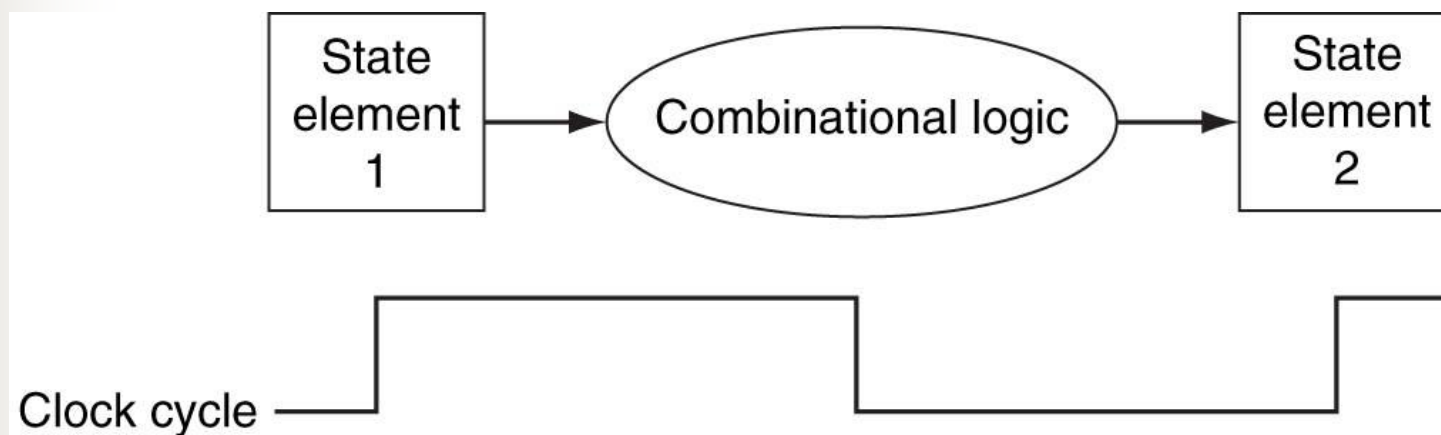
## ■ Como funciona?

- instruções em sequência (P4.1\*)
  - Ler instrução: busca na memória a próxima instrução
  - Decodificar: interpretá-la e gerar os sinais de controle necessários para sua execução
  - Executar: efetivamente concluir sua execução.



Combinacional  
X  
Sequencial?

# Introdução



Circuitos sequenciais síncronos possuem elementos de estado (registradores/FF), acionados pelo clock, e lógica combinacional.

# Processador **MIPS** Simplificado

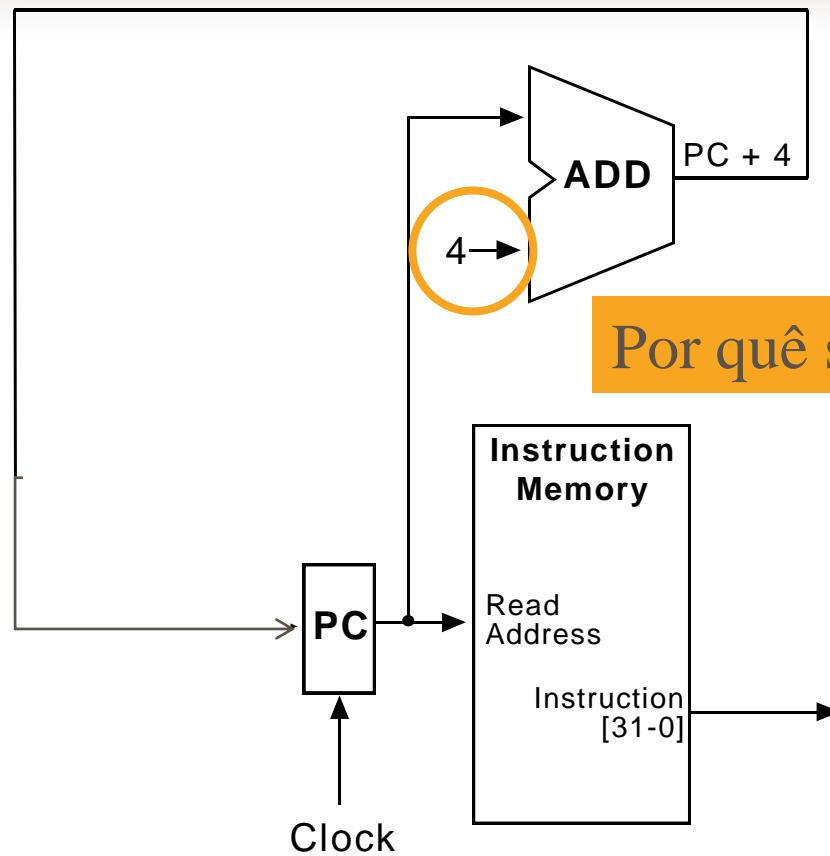
Ler Instr.

*Instruction Fetch*

Incr.Cont.

## ■ Blocos Operacionais

- Registrador Contador de Programa (PC)
- Endereça memória de programa com PC,
- Calcula endereço da próxima instrução:
  - PC+1: instrução na sequência (next **word**).



Por quê soma 4?

**Figure 14.4** Block Diagram of MIPS Fetch Unit.





## Revisão de VHDL e Conceitos novos necessários

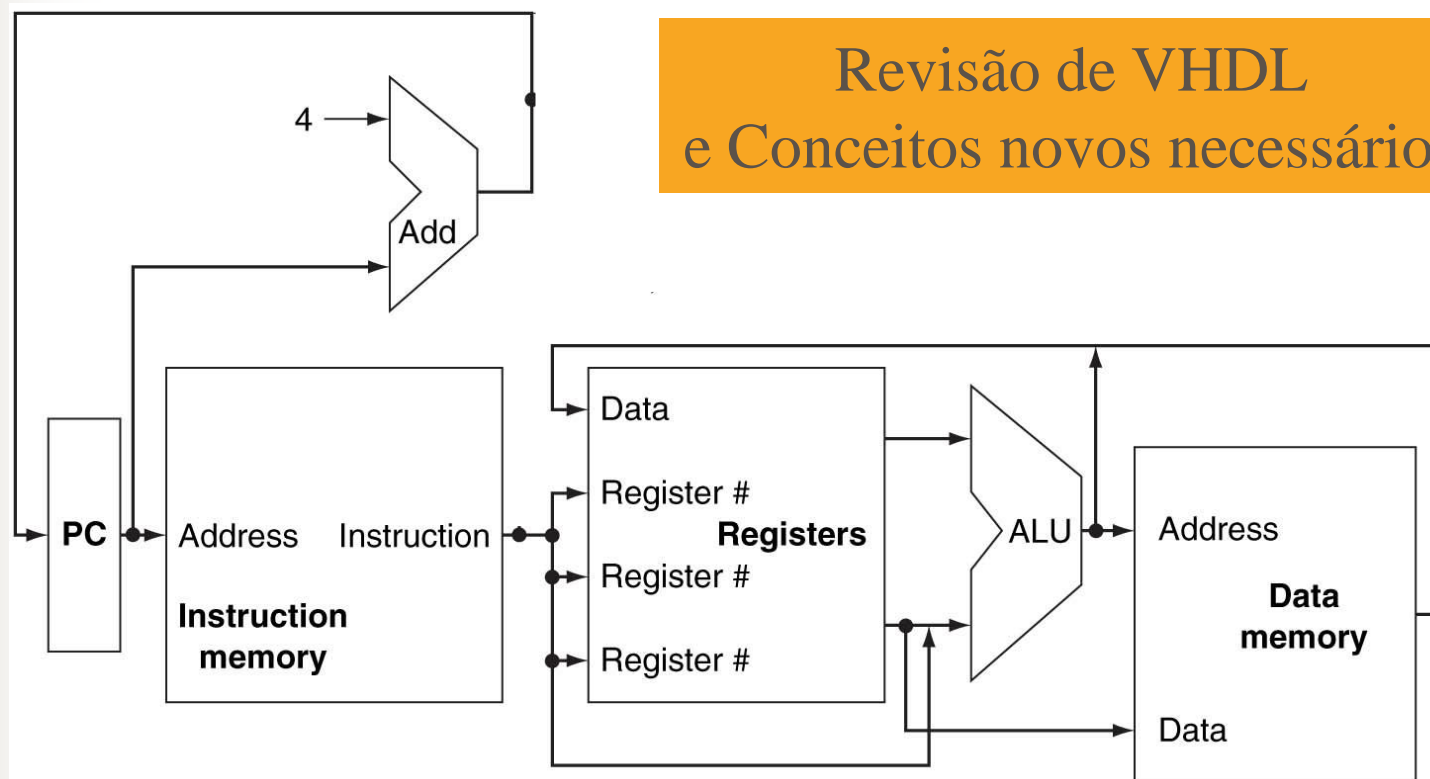


Diagrama de blocos das estruturas básicas do MIPS