

# O Hardware

## Dentro da Unidade do Sistema

### # Memória RAM

#### ▣ Random Access Memory

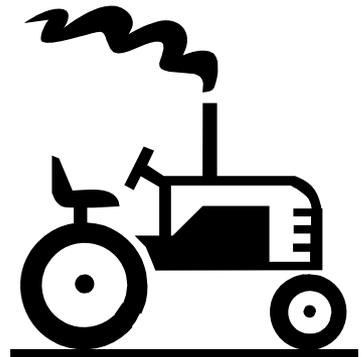
- ▣ Memória de Acesso Aleatório

#### ▣ Armazena:

- ▣ programas em execução
- ▣ dados dos programas em execução
- ▣ alguns programas do sistema operacional

#### ▣ O objetivo é tornar mais rápido o acesso aos programas e dados

- ▣ o disco rígido e outros meios de armazenamento são muito lentos

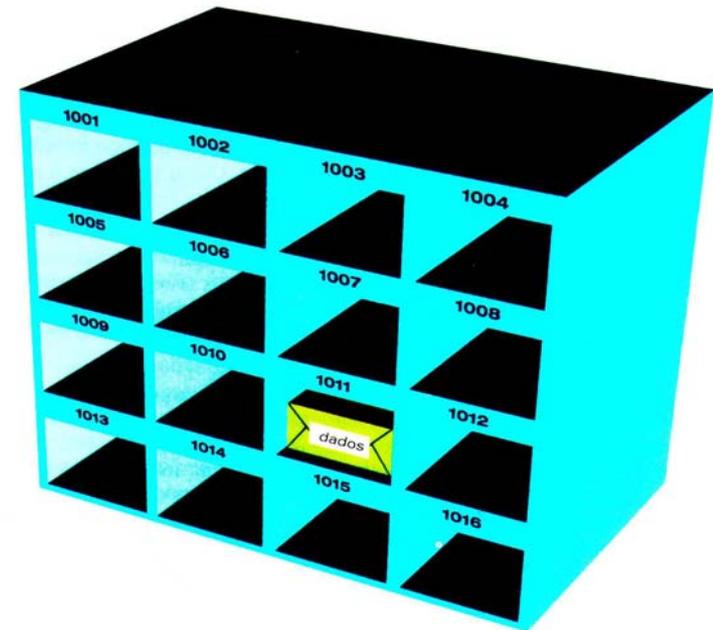


# O Hardware

## Dentro da Unidade do Sistema

### # Memória RAM

- Comparação rápida (velocidade):
  - disco rígido  $\approx 5$  ms
  - memória RAM  $\approx 7$  ns
- Cada posição da memória é referenciada por um endereço
- O processador armazena e lê os dados através do endereço



# O Hardware

## Dentro da Unidade do Sistema

### # Memória RAM

- Quanto de memória o computador deve ter?
  - Depende do objetivo, mas um valor mínimo atualmente é 512 MB
    - # Windows Vista → 1 GB
  - para programas gráficos ou multimídia, esse valor deve ser maior
  - mesmo com valores dessa ordem, os programas são tão grandes ou em tal quantidade que obrigam o sistema operacional a utilizar a **MEMÓRIA VIRTUAL**



# O Hardware

## Dentro da Unidade do Sistema

---

### # Memória RAM

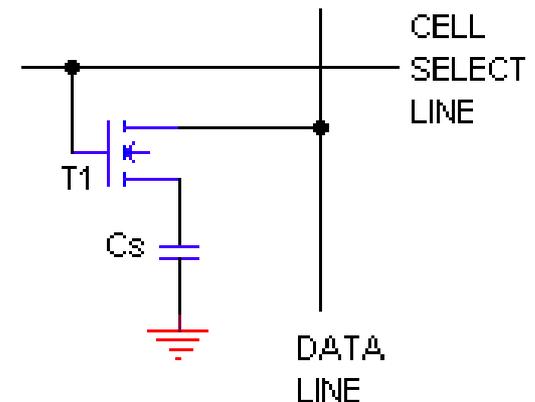
- ...Mas se a memória RAM é tão rápida, por que precisamos do disco rígido (HD)?
  - Porque a RAM é volátil: ao desligar a energia, todos os dados guardados nela são perdidos
- Dois tipos básicos
  - Dinâmica (SDRAM, EDO RAM, Rambus, etc.)
  - Estática (cache)

# O Hardware

## Dentro da Unidade do Sistema

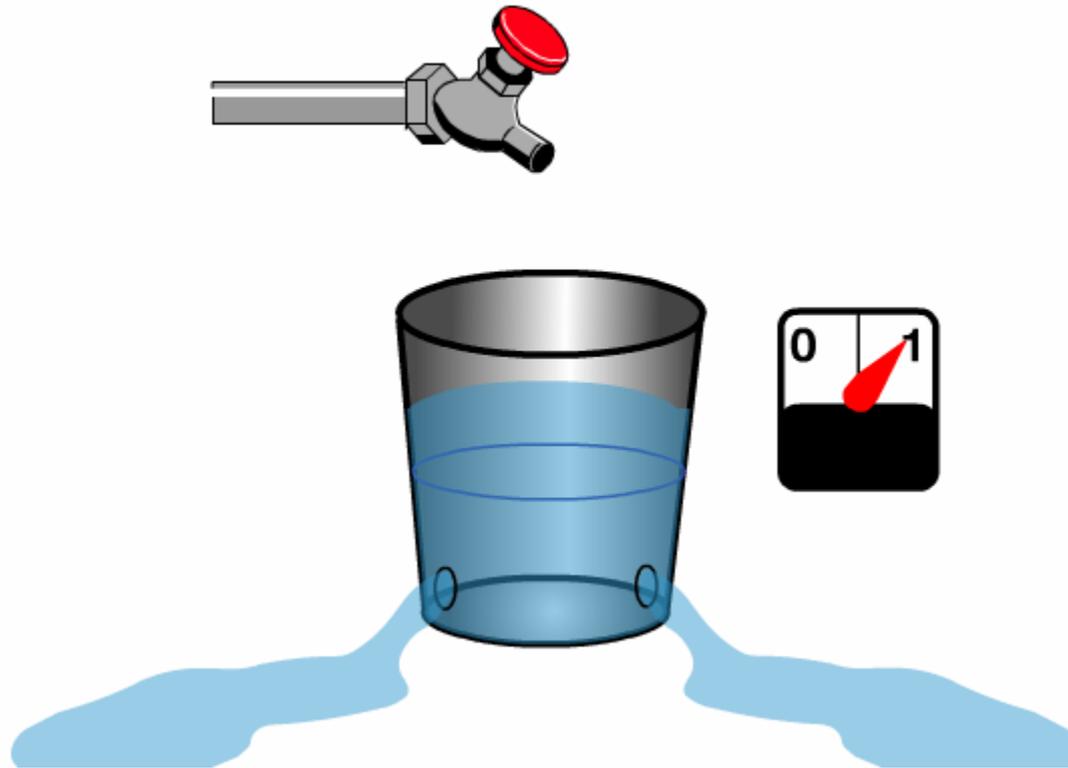
### # RAM Dinâmica (DRAM)

- ▣ Cada endereço ou posição de memória é construído com capacitores: descarrega com o tempo
- ▣ É barata
  - ▣ menos componentes
- ▣ Precisa de *refresh*



DYNAMIC RAM

# O Capacitor como um Balde Furado



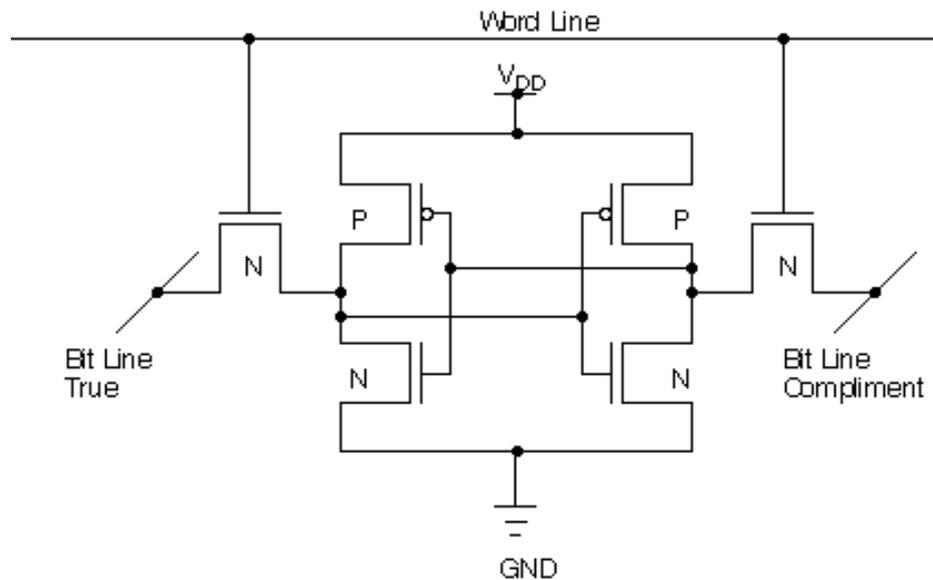
# O Hardware

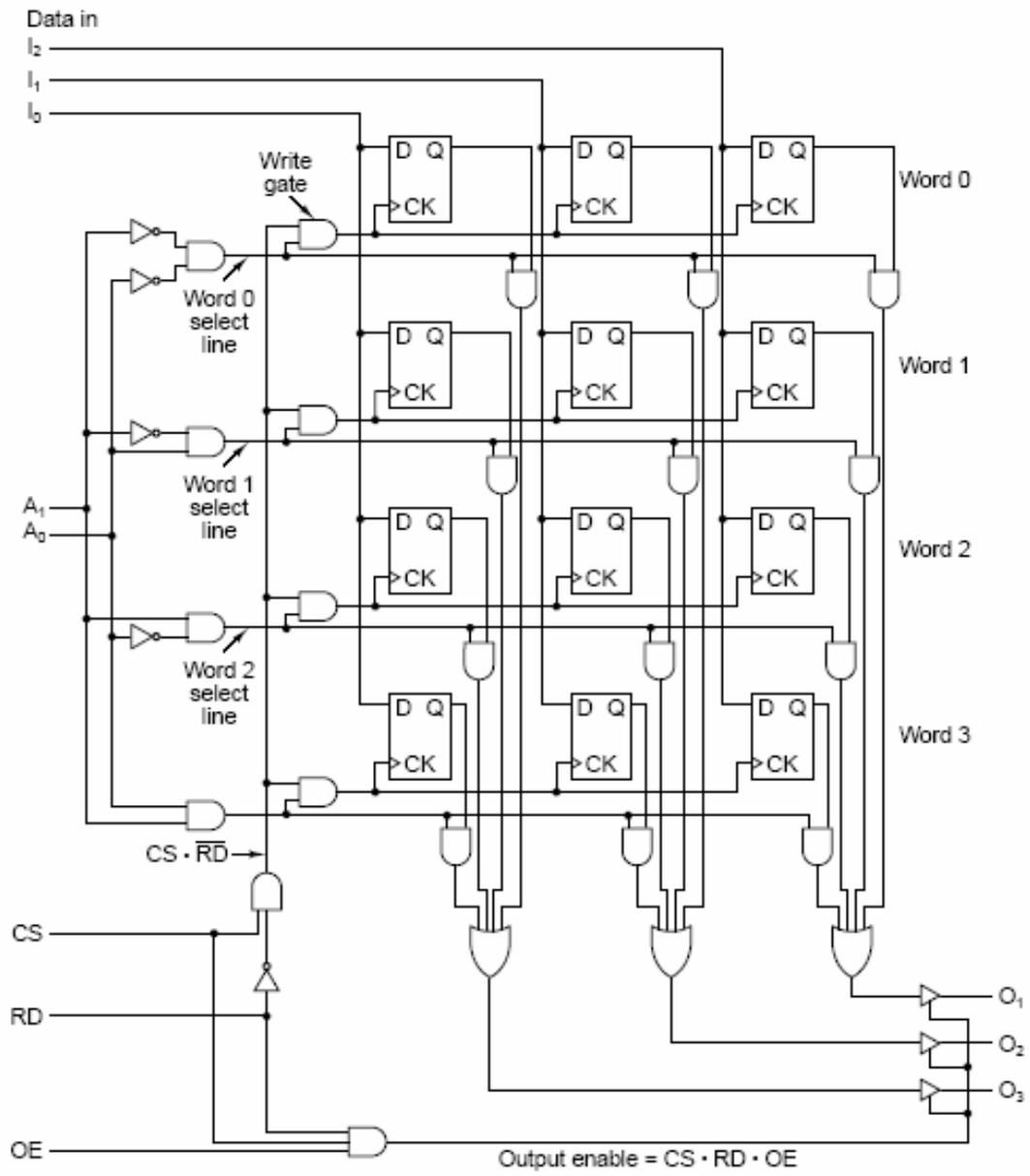
## Dentro da Unidade do Sistema

### # RAM Estática (SRAM)

- É construída com *flip-flops*, que mantêm seu estado até ser mudado ou até a alimentação ser desligada
- Mais cara
  - Quatro a seis transistores
  - Mais espaço físico
- Mais rápida que DRAM
  - Tempo de acesso menor
    - # (SRAM  $\approx$  10-30ns, DRAM  $\approx$  60ns)

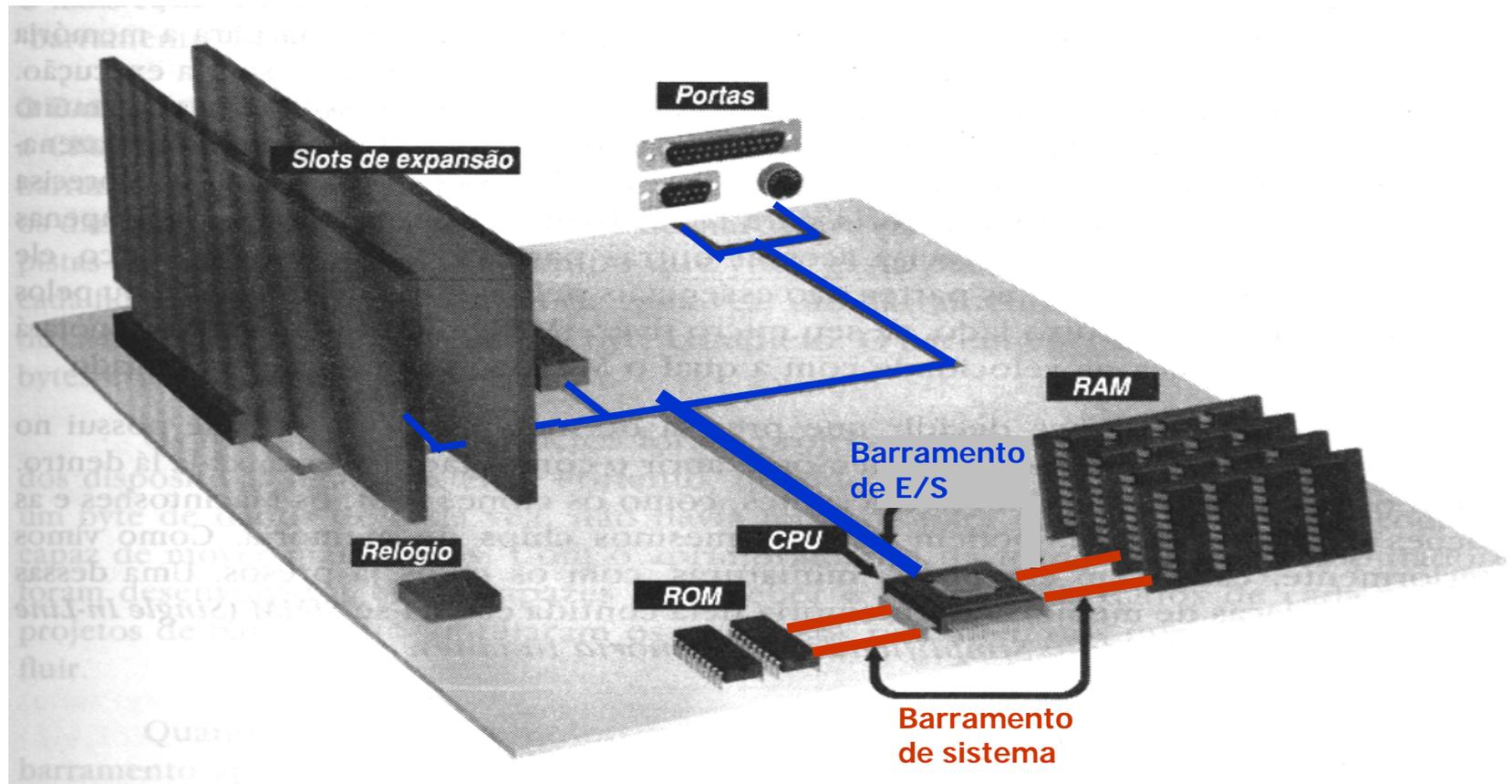
# Célula de RAM Estática (SRAM)





Matriz de RAM Estática (SRAM) 4x3

# O Hardware Dentro da Unidade do Sistema



# O Hardware

## Dentro da Unidade do Sistema

### # Barramentos da Placa Mãe

- ▣ para transferir bits entre o processador e os demais componentes da placa
  - ▣ conjunto de condutores, 1 bit por condutor
- ▣ Barramento do Sistema
  - ▣ liga o processador à memória RAM e outros chips da placa mãe
  - ▣ pode ou não ser da mesma largura do barramento interno do processador
- ▣ Barramento de E/S
  - ▣ liga processador e slots de expansão

# O Hardware

## Dentro da Unidade do Sistema

---

### # Questão importante:

- Em linhas gerais, como se processa a execução de um programa?

### # Veremos a seguir:

- como o processador executa um programa
  - ciclo de busca e execução
- como o processador efetua um cálculo

# O Hardware

## Dentro da Unidade do Sistema

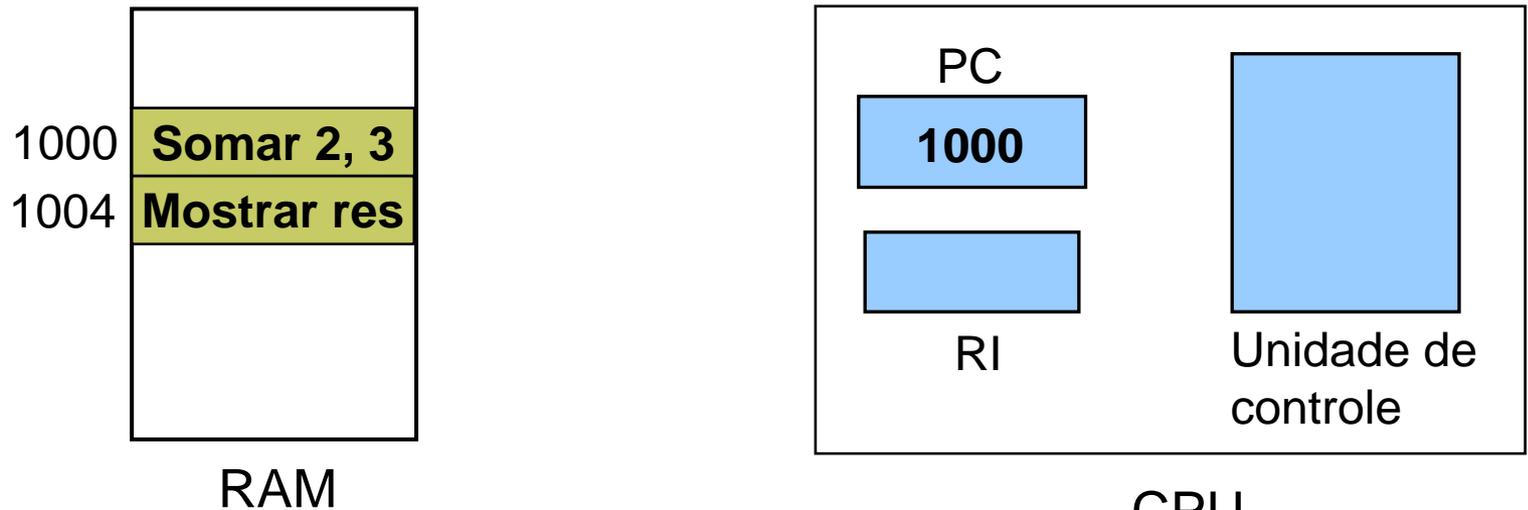
### # Ciclo de Busca-Execução

- ▣ um programa é composto de **instruções**
- ▣ para executar um programa, o processador
  - ▣ busca 1a instrução na memória RAM e executa
  - ▣ busca 2a instrução na RAM e executa
  - ▣ fica nesse ciclo até terminar a execução
- ▣ esse processo é denominado de **ciclo de busca e execução**
- ▣ é comandado pela **unidade de controle**
- ▣ com o auxílio de dois registradores principais **PC e RI**

# O Hardware

## Dentro da Unidade do Sistema

### # Ciclo de Busca-Execução

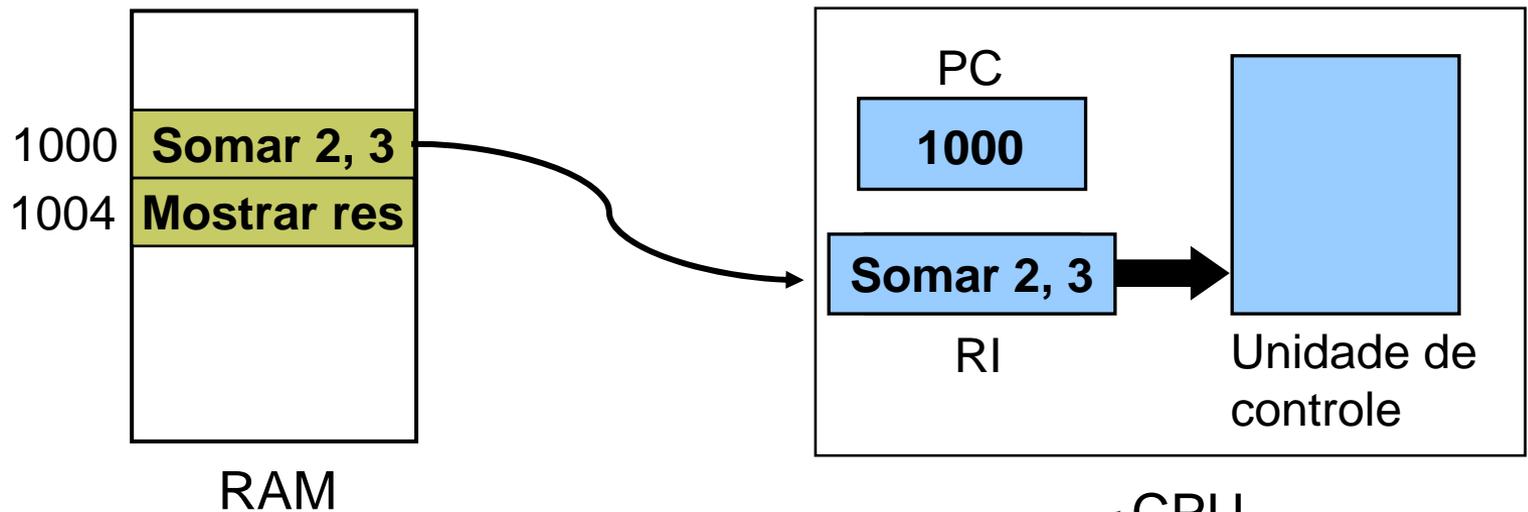


Buscar instrução no endereço 1000

# O Hardware

## Dentro da Unidade do Sistema

### # Ciclo de Busca-Execução

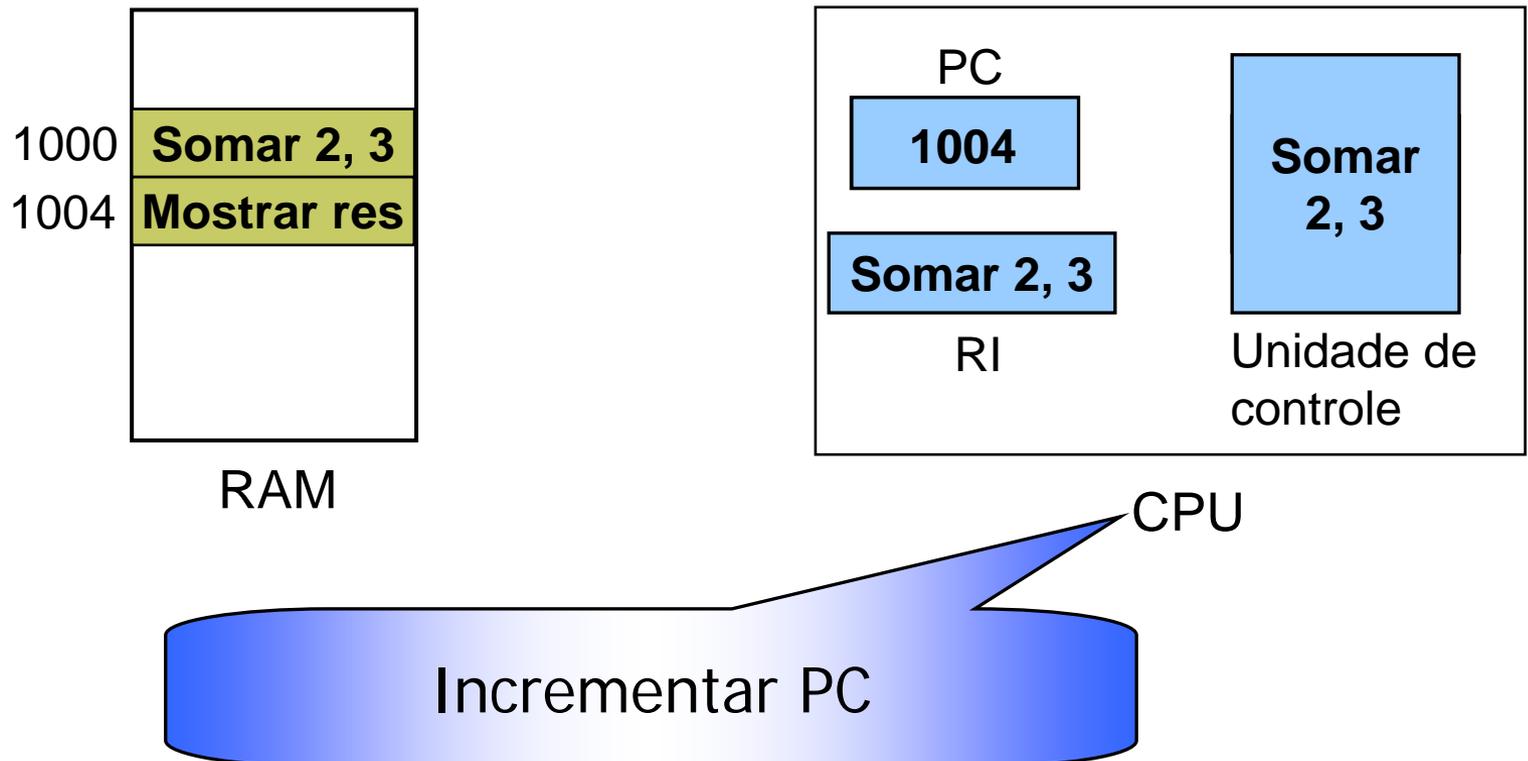


Enviar instrução para ser decodificada na Un. Controle

# O Hardware

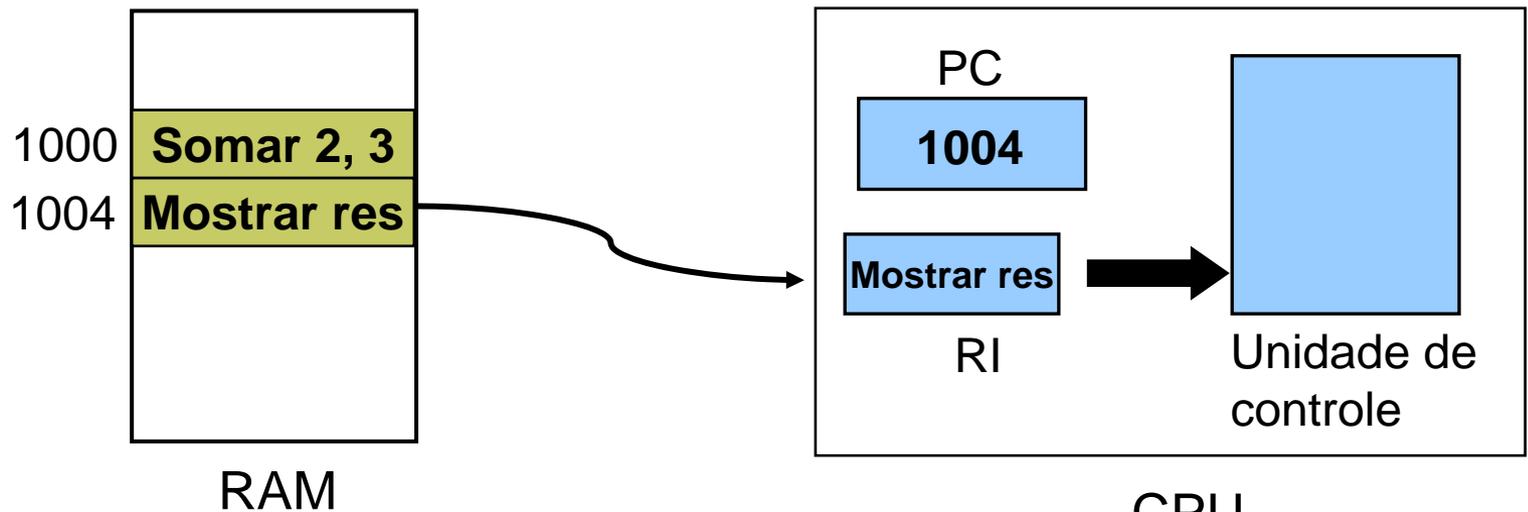
## Dentro da Unidade do Sistema

### # Ciclo de Busca-Execução



# O Hardware Dentro da Unidade do Sistema

## # Ciclo de Busca-Execução



CPU

Tudo outra vez...

# O Hardware

## Dentro da Unidade do Sistema

### # Ciclo de Busca-Execução - detalhado

- ▣ o processador busca uma instrução na memória
  - ▣ cujo **endereço** está guardado no **PC**
- ▣ a instrução é guardada no **registrador de instrução (RI)**
- ▣ a **unidade de controle** decodifica a instrução
- ▣ a **unidade de controle** executa a instrução
  - ▣ isso envolve a ALU, registradores gerais e sinais internos
  - ▣ pode haver necessidade de buscar dados na memória RAM

# O Hardware

## Dentro da Unidade do Sistema

- # E se houver necessidade de efetuar um cálculo com operandos que estão na memória RAM?
- # Exemplo:
  - o processador está executando um programa e buscou a seguinte instrução:
    - some #0 e #2 e guarde o resultado em #3
    - onde #0 e #2 são endereços de memória onde se encontram os valores a serem somados
    - supondo que
      - # #0 contém o número 13
      - # #2 contém o número 20

# Exemplo: 13 + 20



Funcionamento

Ligação entre memória e CPU (barramento)



Memória RAM

0	00010100
1	
2	00001101
3	
⋮	
n-2	
n-1	
n	

Registadores



Unidade Lógica e Aritmética

Unidade de Controle

SOMA



# Exemplo: 13 + 20

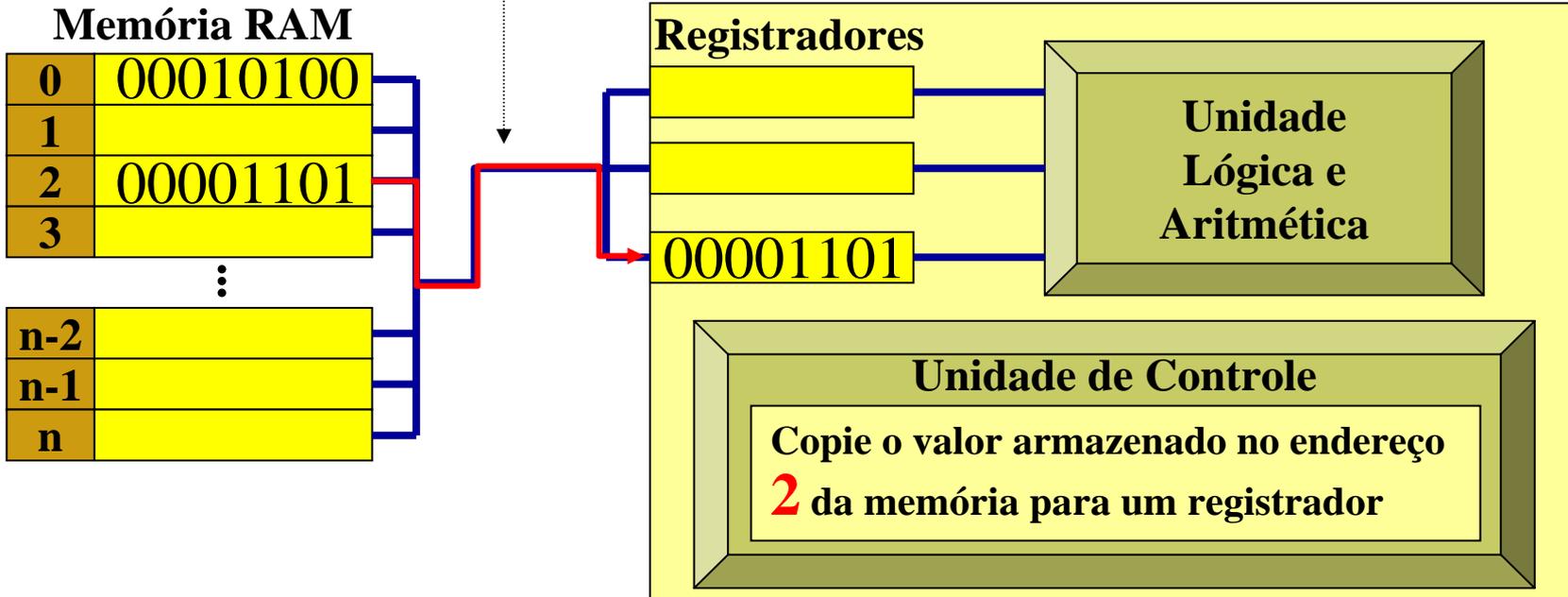


Funcionamento

Ligação entre memória e CPU (barramento)



CPU



# Exemplo: 13 + 20



Funcionamento

Ligação entre memória e CPU (barramento)



## Memória RAM

0	00010100
1	
2	00001101
3	
⋮	
n-2	
n-1	
n	

## CPU

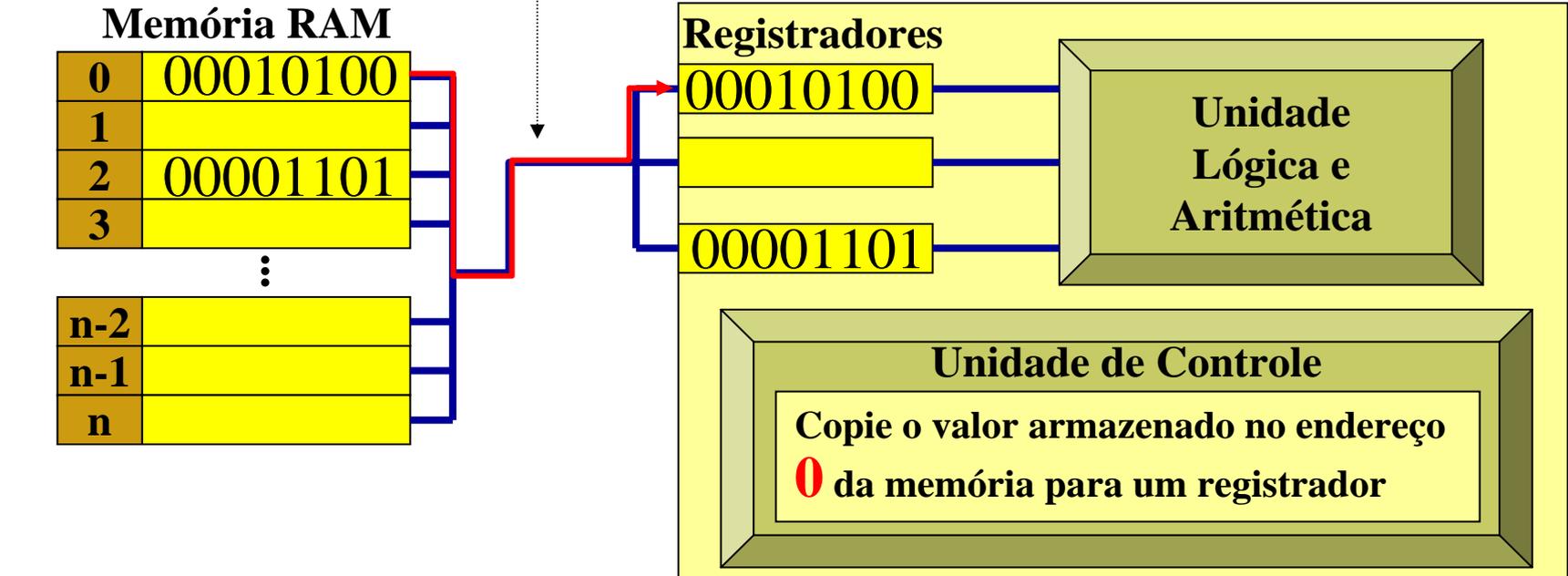
### Registadores

00010100
00001101

Unidade Lógica e Aritmética

### Unidade de Controle

Copie o valor armazenado no endereço **0** da memória para um registrador



# Exemplo: 13 + 20

13+20

Funcionamento

$$\begin{array}{r} 00001101 \\ + 00010100 \\ \hline 00100001 \end{array}$$



## Memória RAM

0	00010100
1	
2	00001101
3	
⋮	
n-2	
n-1	
n	

## Registradores

00010100  
Unidade Lógica e Aritmética  
00001101

Unidade Lógica e Aritmética

## Unidade de Controle

Some o valor dos dois registradores

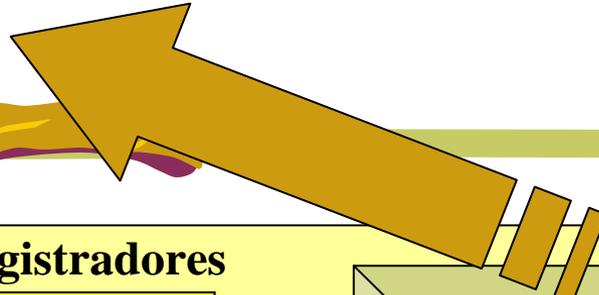


# Exemplo: 13 + 20

13+20

Funcionamento

$$\begin{array}{r} 00001101 \\ + 00010100 \\ \hline 00100001 \end{array}$$



Memória RAM

0	00010100
1	
2	00001101
3	
⋮	
n-2	
n-1	
n	

Registradores

00010100  
Unidade Lógica e Aritmética  
00100001

Unidade Lógica e Aritmética

Unidade de Controle

Armazene o resultado da soma em um registrador

# Exemplo: 13 + 20



Funcionamento

Ligação entre memória e CPU (barramento)



CPU

Memória RAM

0	00010100
1	
2	00001101
3	00100001
⋮	
n-2	
n-1	
n	

Registadores

00010100
00100001

Unidade Lógica e Aritmética

Unidade de Controle

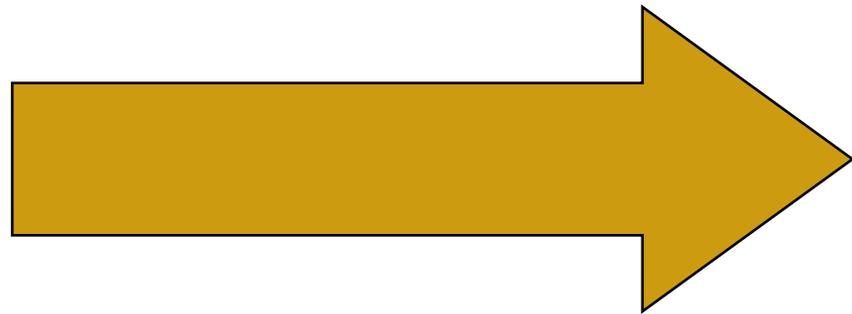
Copie o valor armazenado no registrador para o endereço de memória **3**

# O Hardware

## Dentro da Unidade do Sistema

---

- # Supondo agora que a CPU buscou a seguinte instrução
  - ▣ "Apresente valor armazenado no endereço de memória 3 na tela do computador"



# Exemplo: 13 + 20



Funcionamento

Ligação entre memória e CPU (barramento)

