

# ORGANIZAÇÃO BÁSICA DE COMPUTADORES E LINGUAGEM DE MONTAGEM

---

## Linguagem de programação

- Linguagem de Alto Nível – próximo ao ser humano, escrita de forma textual.
  - Ex: `if (a==b) a=b+c;`
- Linguagem de Montagem (Assembly) – próximo à linguagem de máquina, escrita em códigos (mnemônicos)
  - Ex: `ADD AX,BX;`
- Linguagem de Máquina – linguagem que o computador consegue executar – códigos binários
  - Ex: `01010001`

# ORGANIZAÇÃO BÁSICA DE COMPUTADORES E LINGUAGEM DE MONTAGEM

---

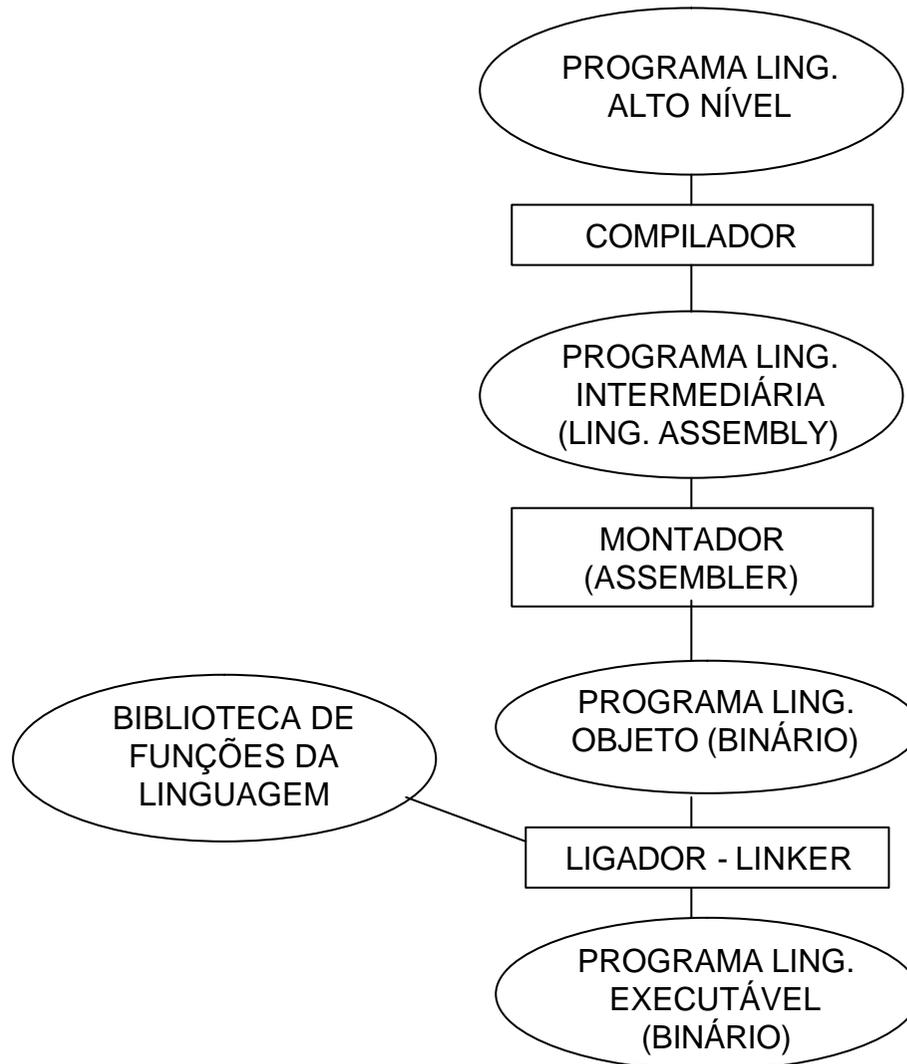
## Execução de um programa

- Um programa escrito em linguagem de alto nível, para ser executado ele deve:
  - Ser traduzido para linguagem de máquina (compiladores);
  - Ter seus endereços realocados, conforme posição onde será carregado na memória (loaders);
  - Alocá-lo em um região da memória (loaders).

# ORGANIZAÇÃO BÁSICA DE COMPUTADORES E LINGUAGEM DE MONTAGEM

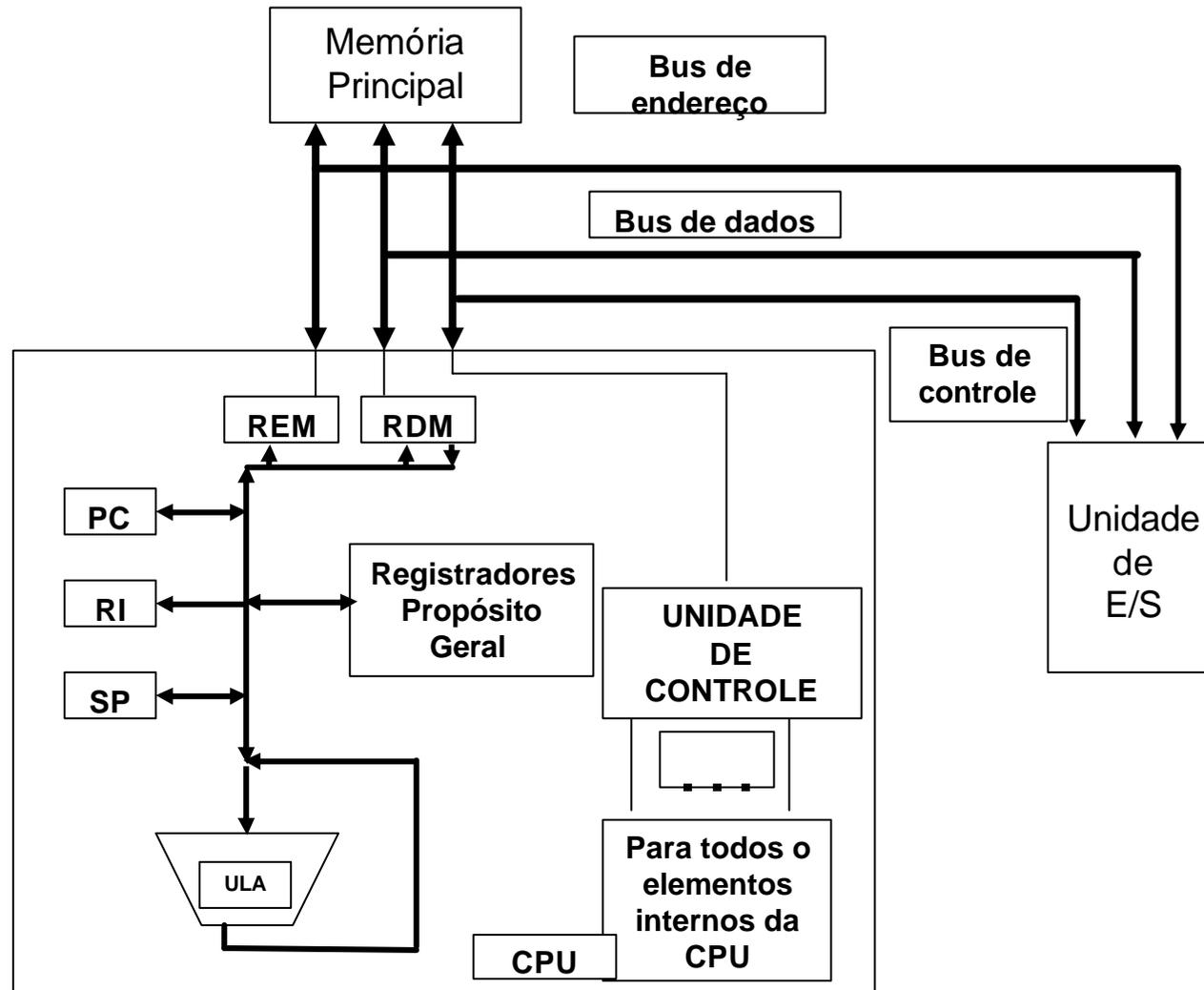
---

Processo de tradução de um programa em linguagem de alto nível



# ORGANIZAÇÃO BÁSICA DE COMPUTADORES E LINGUAGEM DE MONTAGEM

## Organização Básica de um Computador Digital



# ORGANIZAÇÃO BÁSICA DE COMPUTADORES E LINGUAGEM DE MONTAGEM

---

## Organização Básica de um Computador Digital

- **Unidade Central de Processamento – CPU:**
  - Unidade de Controle – UC;
  - Unidade Lógica e Aritmética – ULA;
  - Registradores de Propósito Geral – GPR;
  - Registradores Específicos.
- **Unidade de Memória  hierarquia de memória:**
  - Memória Principal;
  - Memória Secundária;
- **Unidade de Entrada e Saída:**
  - Interfaces;
  - Canais de E/S;
  - Processadores E/S.
- **Barramentos:**
  - Barramento de Endereços;
  - Barramento de Dados;
  - Barramento de Controle.

# **ORGANIZAÇÃO BÁSICA DE COMPUTADORES E LINGUAGEM DE MONTAGEM**

---

## **Organização Básica de um Computador Digital**

- **Unidade Central de Processamento – CPU**
  - **Responsável por todo o processamento (execução de programas) no sistema**
    - **Unidade de Controle: circuito que gera os sinais de controle responsáveis pelo gerenciamento (controle) de todas as atividades do computador.**
    - **Unidade Lógica e Aritmética – ULA: circuito responsável por efetuar todas as operações lógicas e aritméticas.**
    - **Registradores de Propósito Geral – GPR: elementos de memória (circuitos) responsáveis por armazenar os dados que são utilizados durante a execução de um programa (instruções).**

# ORGANIZAÇÃO BÁSICA DE COMPUTADORES E LINGUAGEM DE MONTAGEM

---

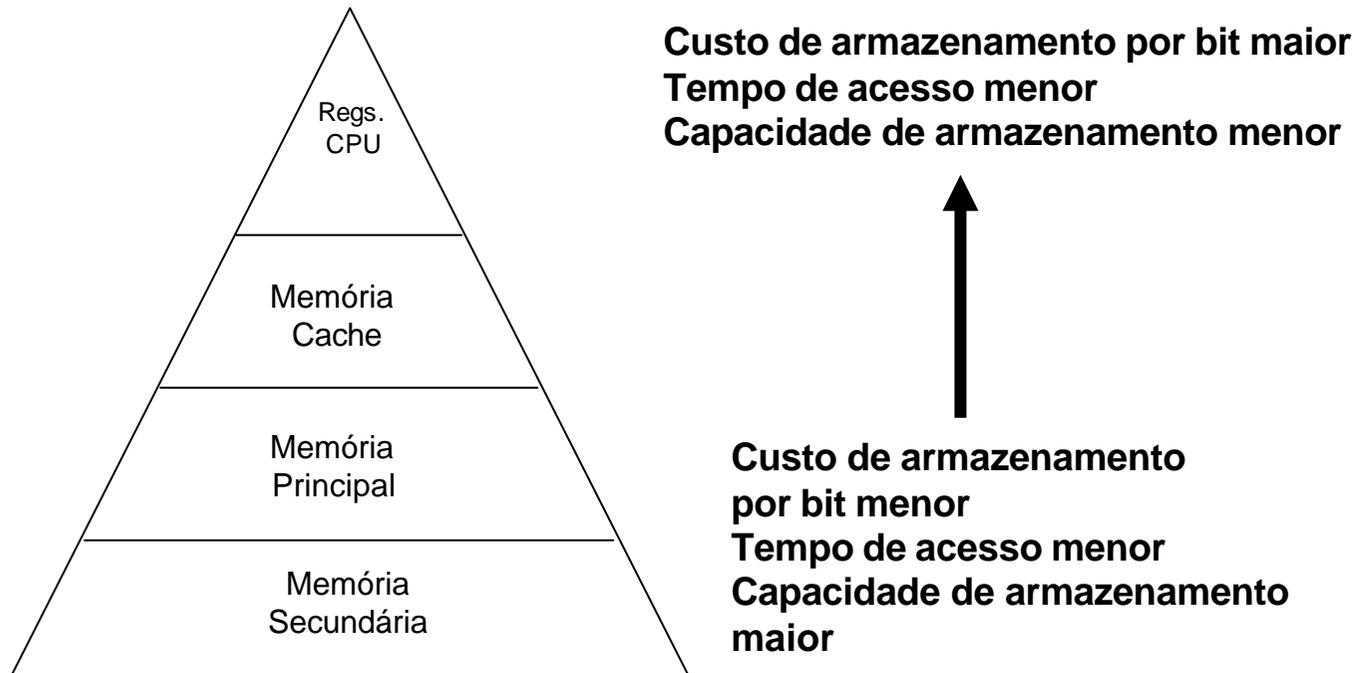
## Organização Básica de um Computador Digital

- **Unidade Central de Processamento – CPU (cont.)**
  - **Registradores Específicos:**
    - **Program Counter – PC:** armazena o endereço da próxima instrução a ser executada;
    - **Stack Pointer – SP:** armazena o endereço do topo da pilha;
    - **Registrador de Instrução – RI:** armazena a instrução que está sendo executada;
    - **Registrador de Dados de Memória – RDM:** armazena os dados que vem da memória (lidos) ou que vão para a memória (escritos);
    - **Registrador de Endereços de memória – REM:** armazena o endereço enviado para a memória, quando ocorrer um acesso à mesma (leitura ou escrita)

# ORGANIZAÇÃO BÁSICA DE COMPUTADORES E LINGUAGEM DE MONTAGEM

## Organização Básica de um Computador Digital

- **Unidade de Memória**
  - **Hierarquia de Memória:** sistema de memória com objetivo de melhorar o desempenho de um sistema computacional, diminuindo o tempo de acesso médio



# ORGANIZAÇÃO BÁSICA DE COMPUTADORES E LINGUAGEM DE MONTAGEM

---

## Organização Básica de um Computador Digital

- **Memórias**
  - **Semicondutoras: fabricadas com materiais semicondutores (silício) – circuitos integrados.**
    - **RAM – Random Access Memory : memória de acesso aleatório, volátil.**
      - **SRAM – RAM estática: seu conteúdo só se altera quando se escreve nela ou quando se desliga a tensão de alimentação. Exemplo – registradores da CPU, memória cache.**
      - **DRAM – RAM dinâmica: periodicamente é necessário reescrever o seu conteúdo (refresh de memória) pois há diminuição de cargas elétricas.**  
Exemplo – memória principal.
    - **ROM – Read Only Memory: memória somente de leitura, não volátil.**
      - **ROM: gravação feita pelo fabricante da memória, não apagável;**
      - **PROM – Programmable ROM: programação feita pelo usuário, não apagável;**
      - **EPROM – Erasable PROM: programação feita pelo usuário, apagável através de luz ultra-violeta;**
      - **EEPROM – Electrical EPROM: programação feita pelo usuário, apagável eletricamente;**
    - **Flash – memória semicondutora, não volátil e de escrita e leitura, apagável.**

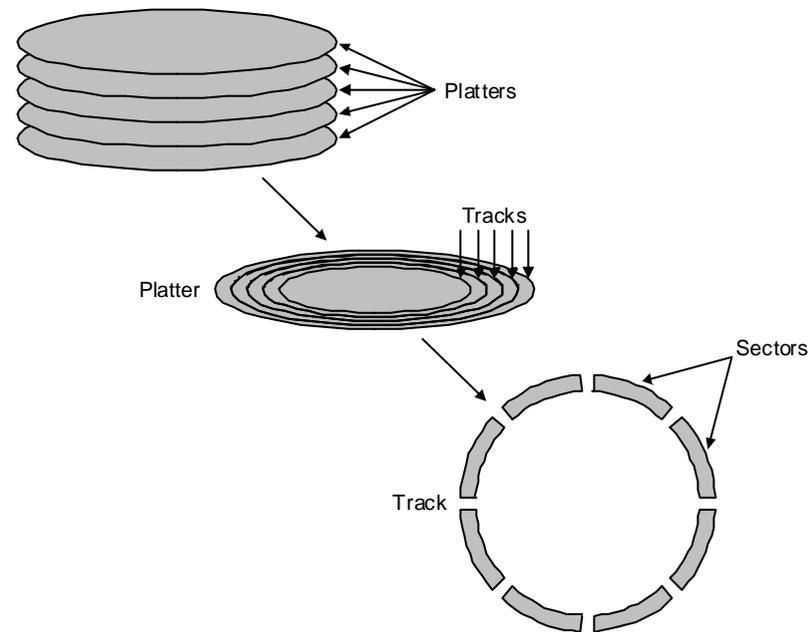
# ORGANIZAÇÃO BÁSICA DE COMPUTADORES E LINGUAGEM DE MONTAGEM

---

## Organização Básica de um Computador Digital

- Memórias (continuação)
  - Magnéticas
    - Discos – Hard Disk – HDs
    - Ópticos – CD-ROM, DVD, etc.
    - Fitas – cartchos, rolos, etc.

Exemplo: memórias secundárias



**Disco Magnético** ✍  
pratos, lados, trilhas  
e setores

# ORGANIZAÇÃO BÁSICA DE COMPUTADORES E LINGUAGEM DE MONTAGEM

---

## Organização Básica de um Computador Digital

- **Unidade de Entrada e Saída: responsável por gerenciar a ligação entre CPU-Memória-barramentos e os periféricos.**
  - **Interfaces – circuitos simples que apenas compatibilizam a comunicação (protocolo). O controle da transferência é feita pela CPU. Exemplo: interface serial RS232, interface paralela, interface USB;**
  - **Canais de E/S – circuitos que controlam e compatibilizam a comunicação. A CPU apenas inicia a transferência. Exemplo – Controlador de Acesso Direto à Memória (DMA – Direct Access Memory);**
  - **Processadores de E/S – são CPUs dedicadas a fazer E/S de dados. Iniciam e controlam a comunicação.**

# ORGANIZAÇÃO BÁSICA DE COMPUTADORES E LINGUAGEM DE MONTAGEM

---

## Organização Básica de um Computador Digital

- **Barramentos: Conjunto de fios que fazem a ligação física entre as diversas unidades.**
  - **Barramento de Endereços: Por onde trafegam os endereços;**
  - **Barramento de Dados: Por onde trafegam os dados;**
  - **Barramento de Controle: por onde trafegam os sinais de controle;**
- **Observação:**

**Internamente à CPU, existe um barramento interno de dados que liga os registradores com a ULA e a UC, e um barramento interno de controle que liga a UC a todos os elementos da CPU.**

# ORGANIZAÇÃO BÁSICA DE COMPUTADORES E LINGUAGEM DE MONTAGEM

---

## Organização Básica de um Computador Digital

- **Formato das Instruções**
  - **Tamanho (número de bits) e o significado de cada campo de bits de uma instrução de linguagem de máquina.**
  
- **Conjunto de Instruções**
  - **Cada processador tem o seu conjunto de instruções de linguagem de máquina (ISA – Instruction Set Architecture). Este conjunto contém todas as instruções, em linguagem de máquina, que o processador pode executar.**

# ORGANIZAÇÃO BÁSICA DE COMPUTADORES E LINGUAGEM DE MONTAGEM

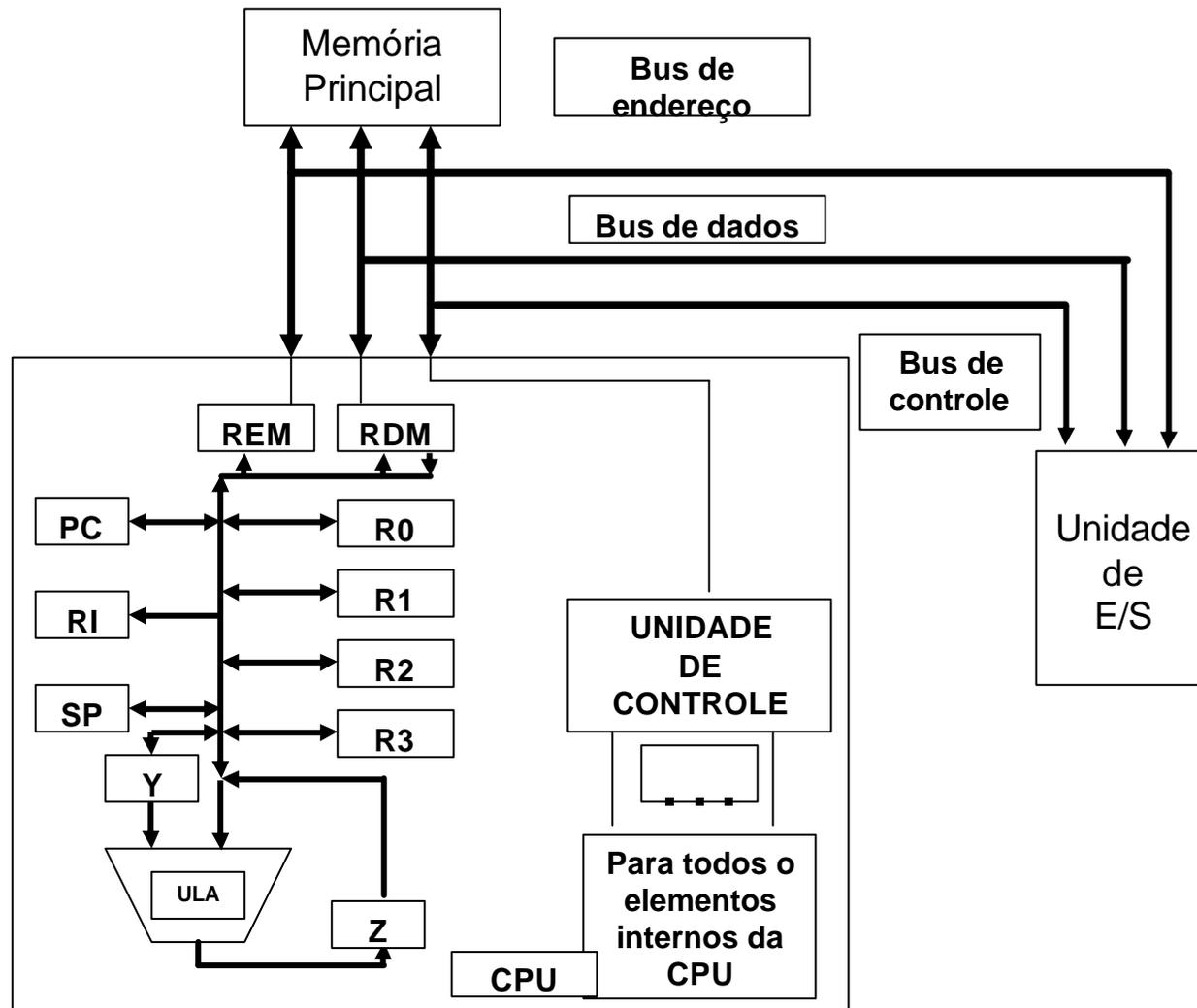
---

## Execução de uma instrução pela CPU

- **Ciclo de execução de uma instrução:**
- **Leitura da instrução da memória principal – Fetch da Instrução**
  - REM ↗ PC
  - Read (sinal de controle)
  - PC ↗ PC atualizado
  - RDM ↗ [REM] (instrução lida)
- **Decodificação da instrução**
  - RI ↗ RDM (instrução)
  - É feita a decodificação pela Unidade de Controle
- **Busca dos operandos da instrução na memória – se houver**
  - REM ↗ PC
  - Read (sinal de controle)
  - PC ↗ PC atualizado
  - RDM ↗ [REM] (operando lido)
- **Execução da instrução – depende da instrução**
- **Obs – Quando usamos [..], significa que estamos acessando um conteúdo de memória, cujo endereço está dentro dos colchetes.**

# ORGANIZAÇÃO BÁSICA DE COMPUTADORES E LINGUAGEM DE MONTAGEM

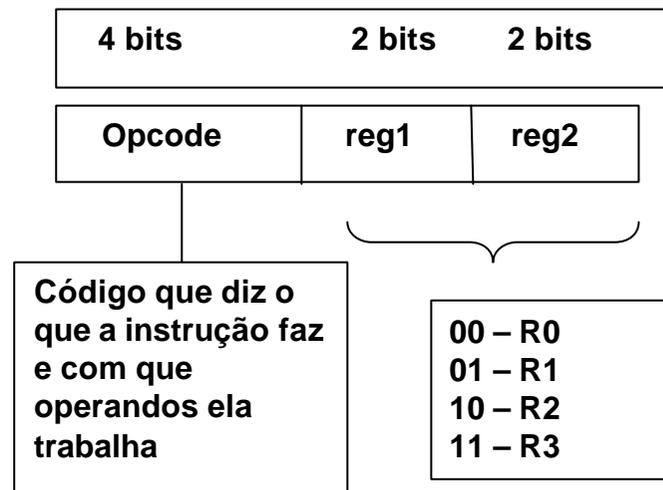
## ESTUDO DE CASO - CPU HIPOTÉTICA



# ORGANIZAÇÃO BÁSICA DE COMPUTADORES E LINGUAGEM DE MONTAGEM

## ESTUDO DE CASO - CPU HIPOTÉTICA

- **Formatos das instruções da CPU HIPOTÉTICA:**
  - **Formato tipo I – Uma palavra de 8 bits, com os seguintes campos:**

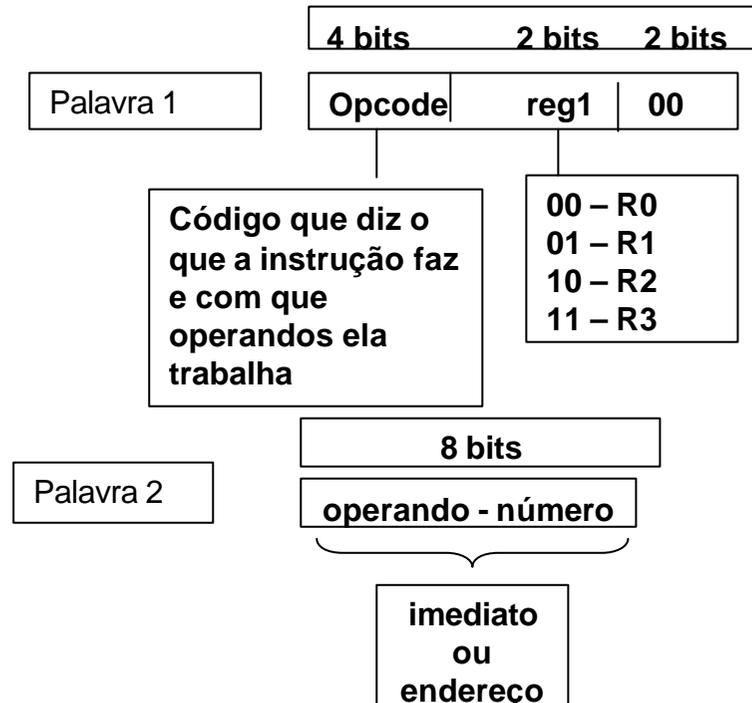


**Exemplo: MOV R0,R1 ; R0 ~~R1~~ R1**

# ORGANIZAÇÃO BÁSICA DE COMPUTADORES E LINGUAGEM DE MONTAGEM

## ESTUDO DE CASO - CPU HIPOTÉTICA

- Formatos das instruções da CPU HIPOTÉTICA:
  - Formato tipo II – Duas palavras de 8 bits, com os seguintes campos:



Exemplos:

**MOV R0, 5 ; R0  $\leftarrow$  5**

**MOV R0, [5] ; R0  $\leftarrow$  [5]**

# ORGANIZAÇÃO BÁSICA DE COMPUTADORES E LINGUAGEM DE MONTAGEM

Mnemônico	Operandos	Opcode	Significado
<b>Instruções de Movimentação de Dados</b>			
MOV	Reg1,Reg2	0000	Reg1 $\leftarrow$ Reg2
MOV	Reg,imed	1000	Reg $\leftarrow$ imed
MOV	Reg,[end]	1001	Reg $\leftarrow$ [end]
MOV	[end],Reg	1010	[end] $\leftarrow$ Reg
<b>Instruções Aritméticas e Lógicas</b>			
ADD	Reg1,Reg2	0001	Reg1 $\leftarrow$ Reg1 + Reg2
ADD	Reg,imed	1011	Reg $\leftarrow$ Reg + imed
SUB	Reg1,Reg2	0010	Reg1 $\leftarrow$ Reg1 - Reg2
SUB	Reg,imed	1100	Reg $\leftarrow$ Reg - imed
AND	Reg1,Reg2	0011	Reg1 $\leftarrow$ Reg1 <u>e</u> Reg2
AND	Reg,imed	1101	Reg $\leftarrow$ Reg <u>e</u> imed
OR	Reg1,Reg2	0100	Reg1 $\leftarrow$ Reg1 <u>ou</u> Reg2
<b>Instruções de Manipulação de Pilha</b>			
PUSH	Reg	0101	SP-- , [SP] $\leftarrow$ Reg
POP	Reg	0110	Reg $\leftarrow$ [SP], SP++
<b>Instruções de Controle de Fluxo de Execução</b>			
JMP	end	1110	PC $\leftarrow$ end
CALL	end	1111	SP-- , [SP] $\leftarrow$ PC , PC $\leftarrow$ end
RET	---	0111	PC $\leftarrow$ [SP] , SP++

# ORGANIZAÇÃO BÁSICA DE COMPUTADORES E LINGUAGEM DE MONTAGEM

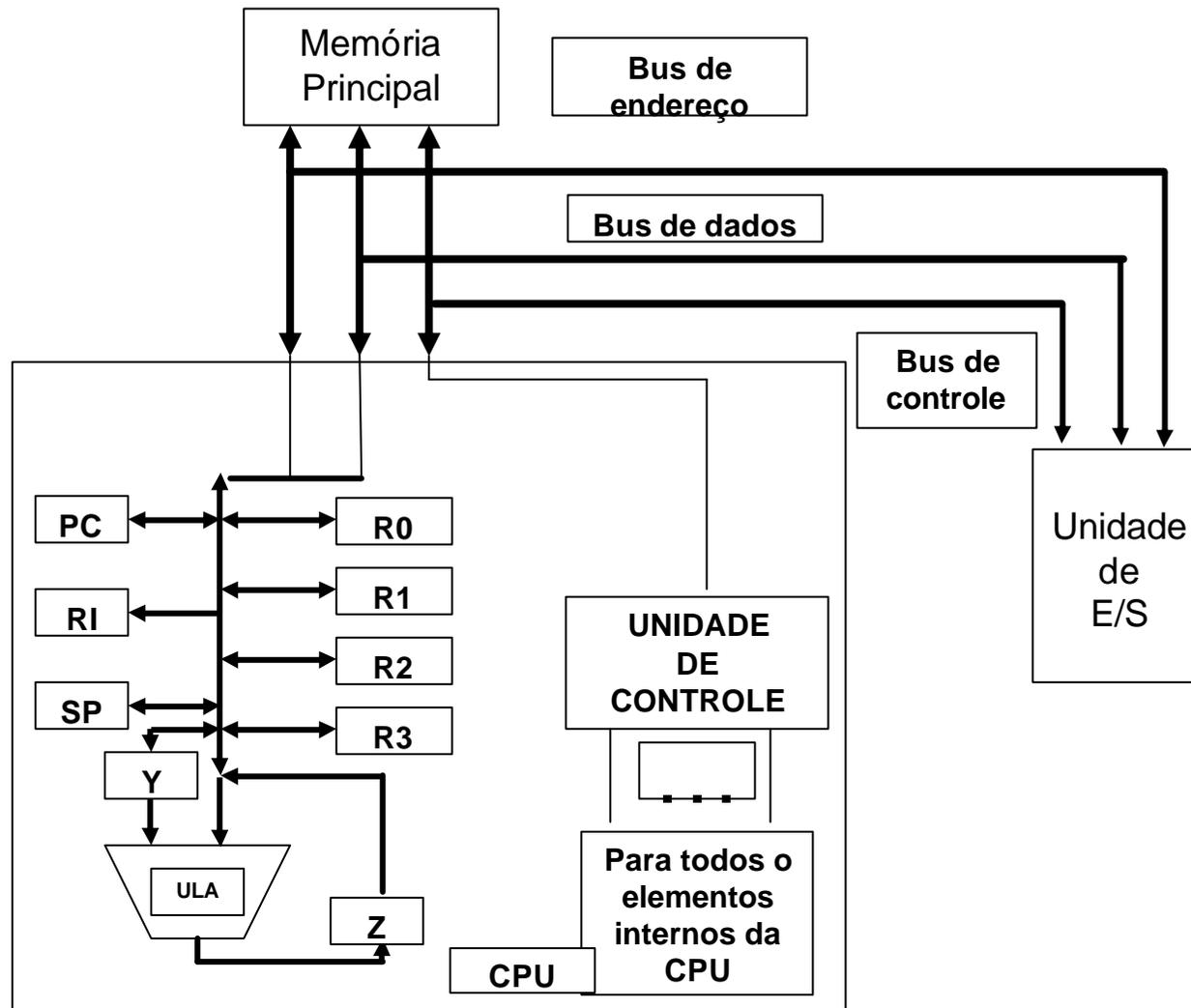
---

## Exercícios

- **Mostrar o ciclo de execução de instruções para todas as instruções do ISA da CPU Hipotética 1 e 2.**

# ORGANIZAÇÃO BÁSICA DE COMPUTADORES E LINGUAGEM DE MONTAGEM

## ESTUDO DE CASO 2 - CPU HIPOTÉTICA

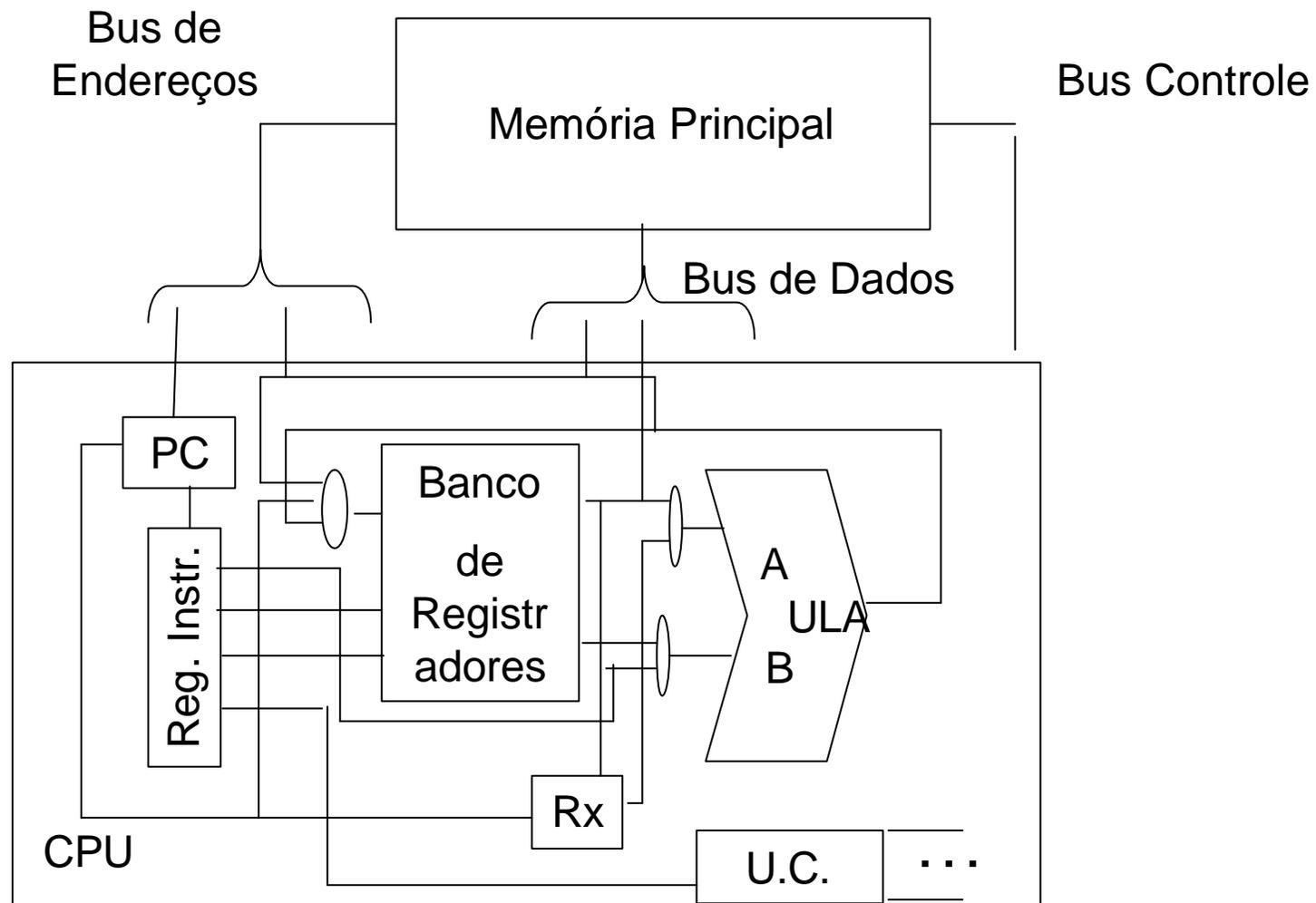


# ORGANIZAÇÃO BÁSICA DE COMPUTADORES E LINGUAGEM DE MONTAGEM

Mnemônico	Operandos	Opcode	Significado
<b>Instruções de Movimentação de Dados</b>			
MOV	Reg1,Reg2	0000	Reg1 $\leftarrow$ Reg2
MOV	Reg,imed	1000	Reg $\leftarrow$ imed
MOV	Reg,[end]	1001	Reg $\leftarrow$ [end]
MOV	[end],Reg	1010	[end] $\leftarrow$ Reg
<b>Instruções Aritméticas e Lógicas</b>			
ADD	Reg1,Reg2	0001	Reg1 $\leftarrow$ Reg1 + Reg2
ADD	Reg,imed	1011	Reg $\leftarrow$ Reg + imed
SUB	Reg1,Reg2	0010	Reg1 $\leftarrow$ Reg1 - Reg2
SUB	Reg,imed	1100	Reg $\leftarrow$ Reg - imed
AND	Reg1,Reg2	0011	Reg1 $\leftarrow$ Reg1 <u>e</u> Reg2
AND	Reg,imed	1101	Reg $\leftarrow$ Reg <u>e</u> imed
OR	Reg1,Reg2	0100	Reg1 $\leftarrow$ Reg1 <u>ou</u> Reg2
<b>Instruções de Manipulação de Pilha</b>			
PUSH	Reg	0101	SP-- , [SP] $\leftarrow$ Reg
POP	Reg	0110	Reg $\leftarrow$ [SP], SP++
<b>Instruções de Controle de Fluxo de Execução</b>			
JMP	end	1110	PC $\leftarrow$ end
CALL	end	1111	SP-- , [SP] $\leftarrow$ PC , PC $\leftarrow$ end
RET	---	0111	PC $\leftarrow$ [SP] , SP++

# ORGANIZAÇÃO BÁSICA DE COMPUTADORES E LINGUAGEM DE MONTAGEM

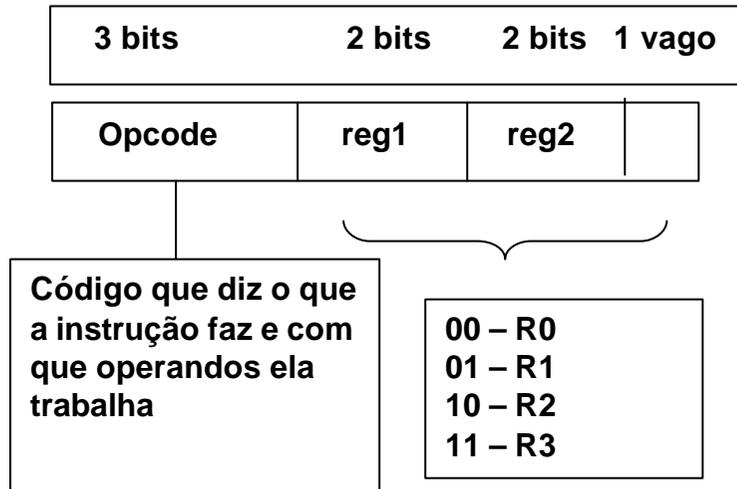
## CPU HIPOTÉTICA 3



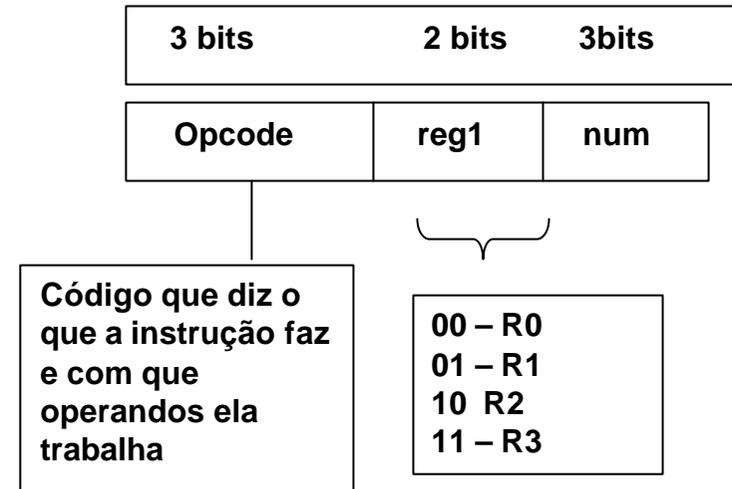
# ORGANIZAÇÃO BÁSICA DE COMPUTADORES E LINGUAGEM DE MONTAGEM

## ESTUDO DE CASO - CPU HIPOTÉTICA 3 – Formato de Instruções

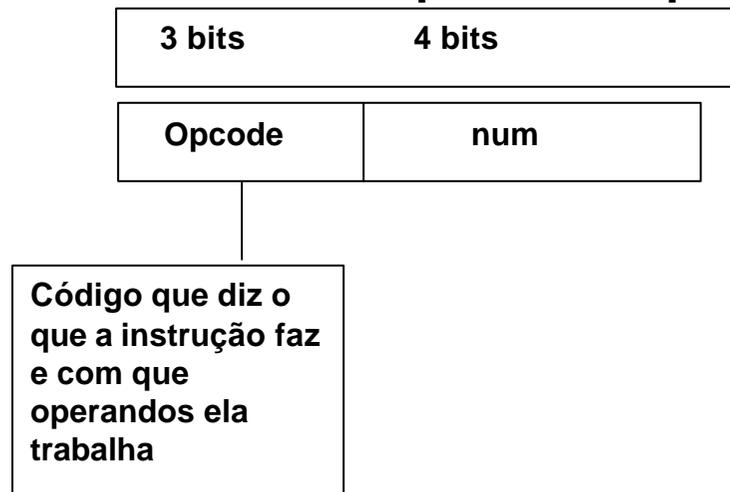
### – Formato tipo R – Registrador



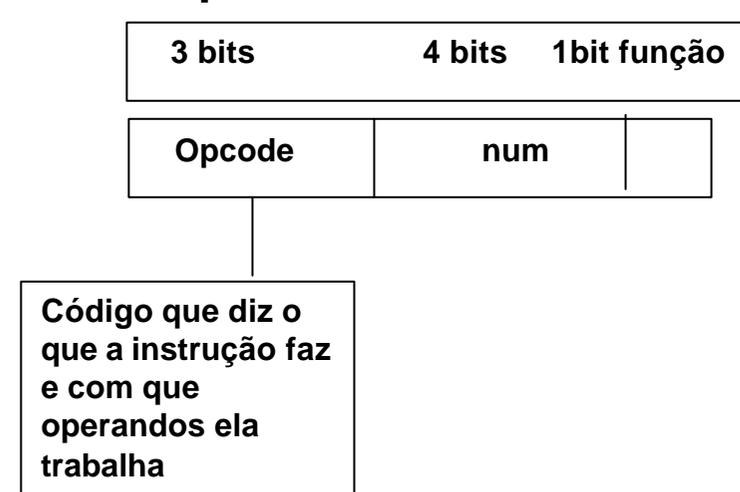
### – Formato I – Imediato



### – Formato tipo J – Jump



### – Formato tipo S - subrotina



# ORGANIZAÇÃO BÁSICA DE COMPUTADORES E LINGUAGEM DE MONTAGEM

## Conjunto de Instruções – CPU Hipotética 3

Mnemônico	Operandos	Opcode	Significado
<b>Instrução especial</b>			
<b>MV</b>	Rx,reg	000	Rx $\leftarrow$ Reg
<b>Instruções de load e store</b>			
<b>LW</b>	Reg,num	001	Reg $\leftarrow$ [Rx + num]
<b>SW</b>	Reg,num	010	[Rx + num] $\leftarrow$ Reg
<b>Instruções Aritméticas e Lógicas</b>			
<b>ADD</b>	Reg1,Reg2	011	Reg1 $\leftarrow$ Reg1 + Reg2
<b>SUB</b>	Reg1,Reg2	100	Reg1 $\leftarrow$ Reg1 - Reg2
<b>AND</b>	Reg1,Reg2	101	Reg1 $\leftarrow$ Reg1 <u>e</u> Reg2
<b>Instruções de Controle de Fluxo de Execução</b>			
<b>JMP</b>	num	110	PC $\leftarrow$ num
<b>JAL</b>	num	111 0	Rx $\leftarrow$ PC , PC $\leftarrow$ num
<b>RET</b>	---	111 1	PC $\leftarrow$ Rx