

aula 6: **PROCESSADORES**

disciplina: **Organização e Arquitetura
de Computadores**

professora: **Sara Guimarães Negreiros**

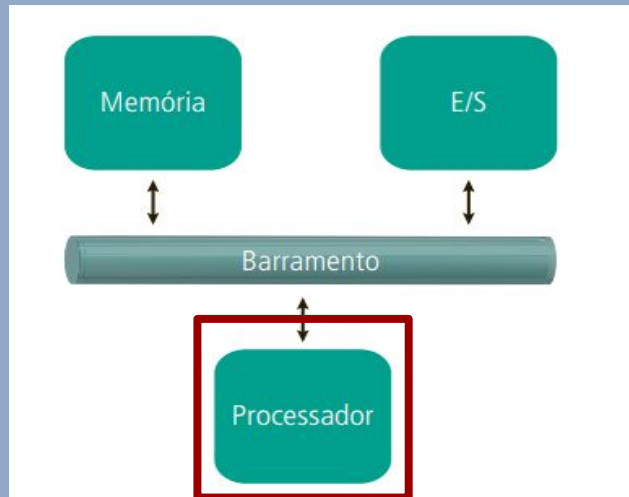
1

VISÃO GERAL



Processador no computador

- Execução de instruções e controle dos demais componentes





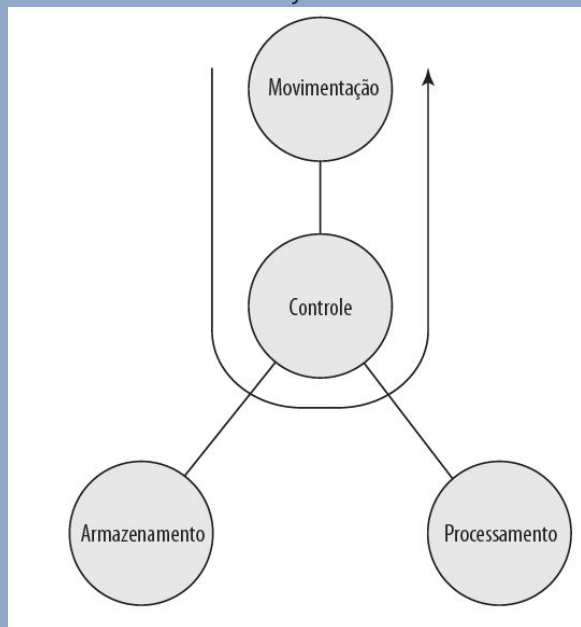
Contexto

- O computador é composto de um pequeno conjunto de componentes lógicos básicos, que podem ser combinados de vários modos para armazenar dados binários e executar operações aritméticas e lógicas sobre esses dados.
- Computadores binários: níveis de tensão.
 - ▷ Informação no interior na base binária (aritmética, dados, armazenamento).

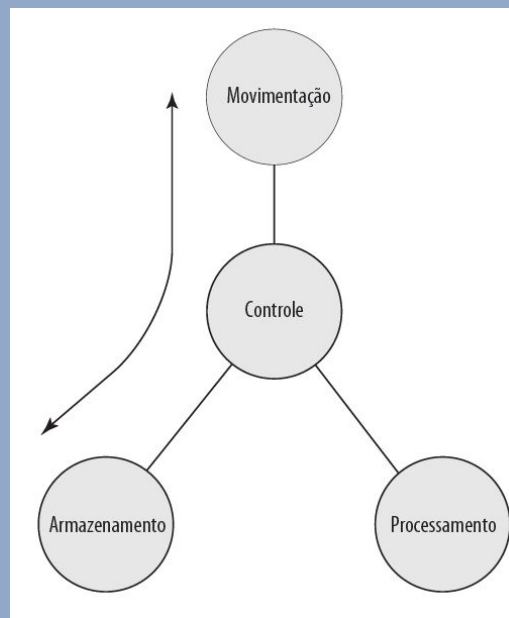


Operações

MOVIMENTAÇÃO DE DADOS



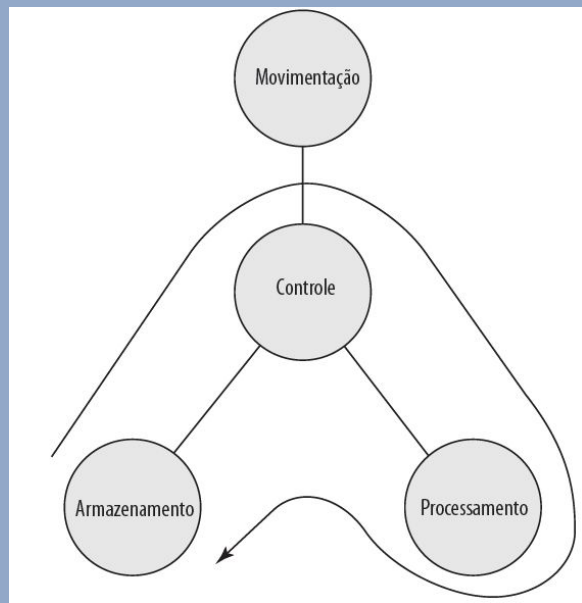
ARMAZENAMENTO DE DADOS



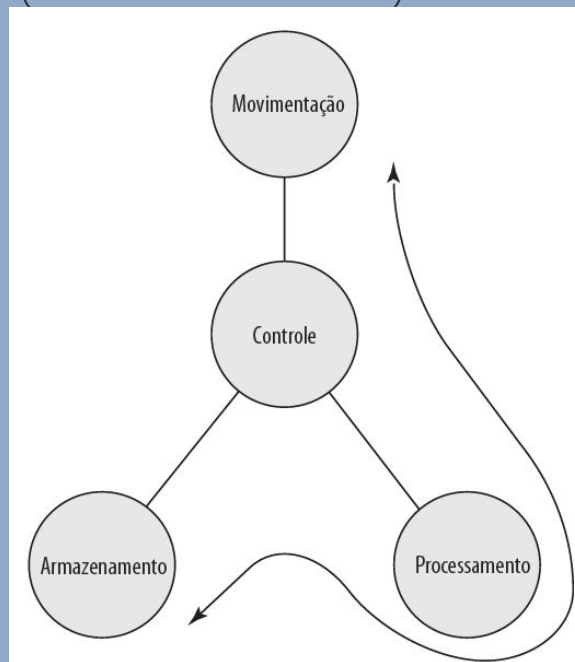


Operações

PROCESSAMENTO DE DADOS



(MUNDO EXTERNO)



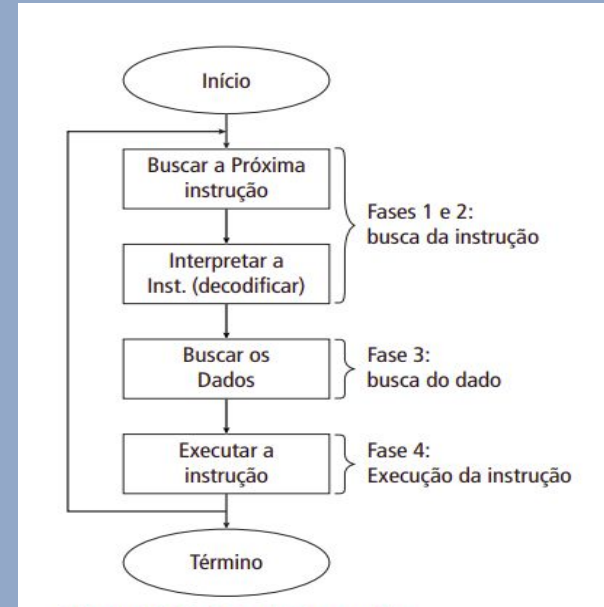
2

CICLO DE INSTRUÇÃO

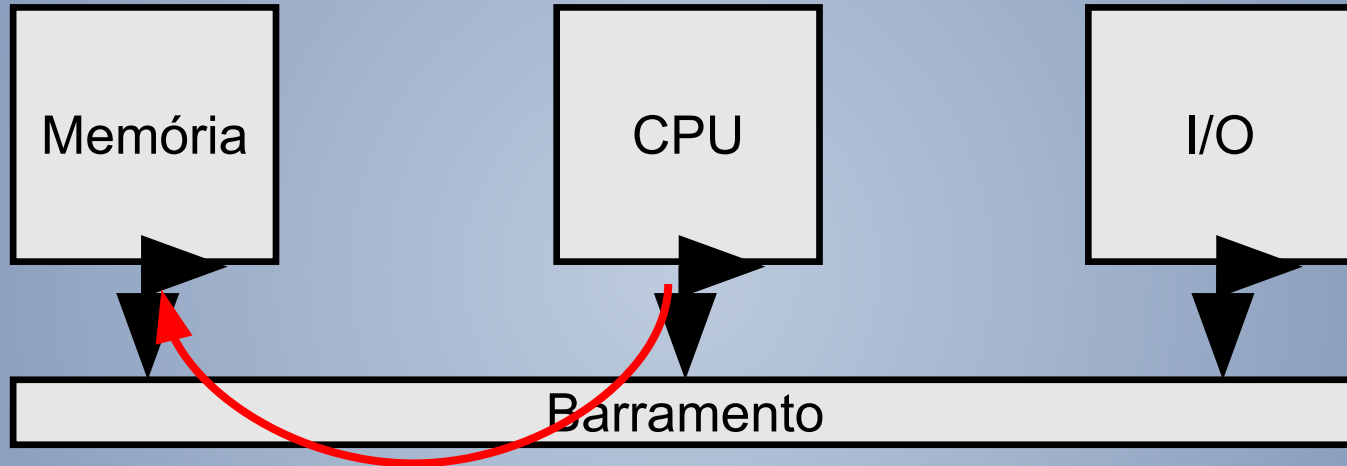


Ciclo de instrução

- Execução em sequência de instruções.
- Conjunto de instruções constitui em um programa.
- O processamento de dados visa obter informação.
- Processo cíclico.

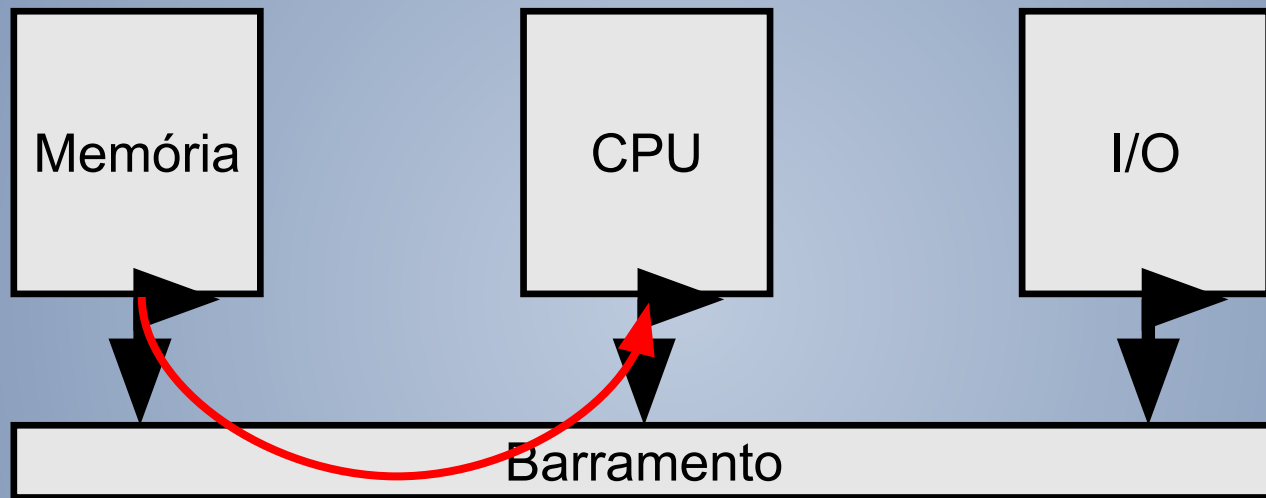


A CPU busca uma instrução na
memória principal



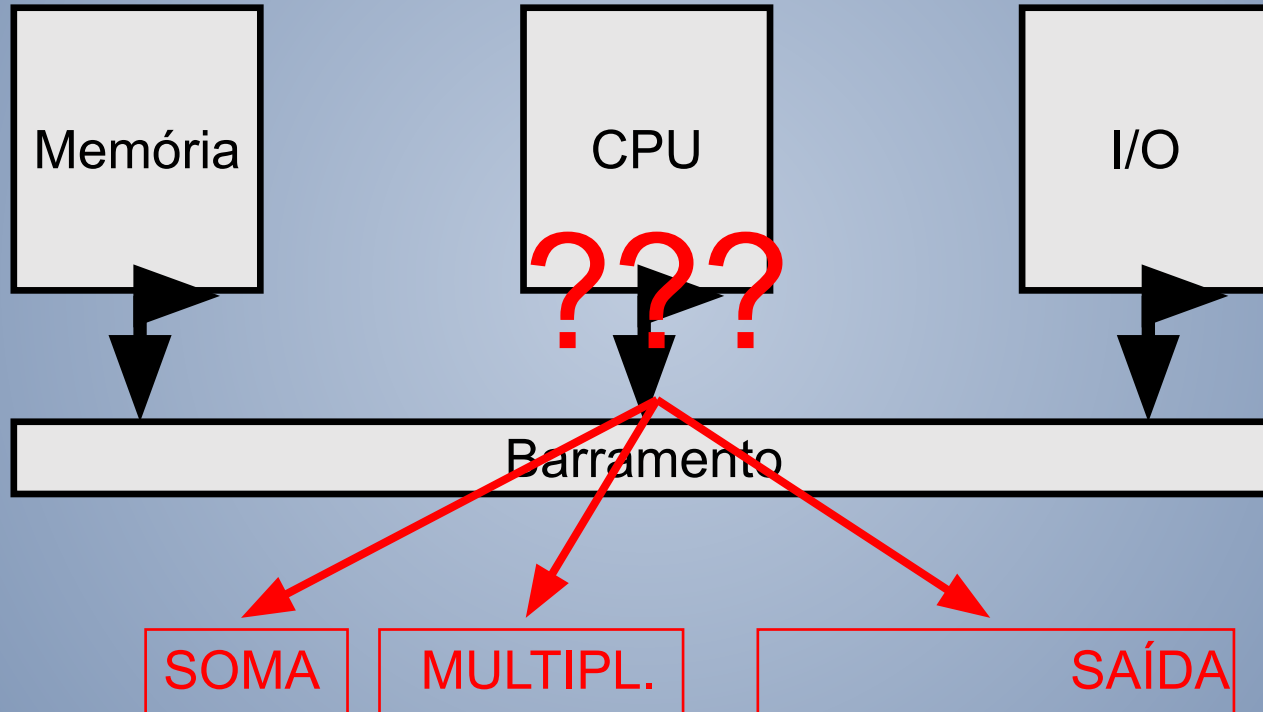
*Essa requisição acontece através
de sinais “colocados” barramento
(sinais de controle)*

A memória responde com a
instrução a ser executada na CPU

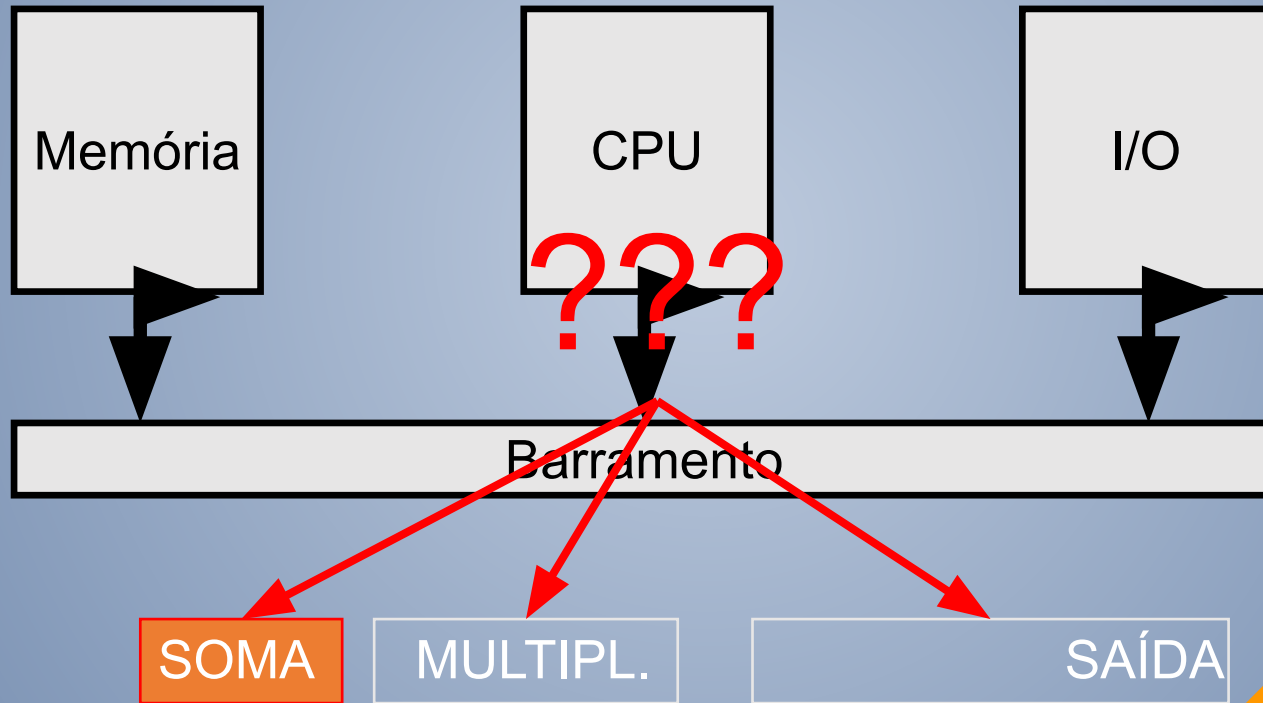


*O envio dessa informação (a
instrução) acontece através do
barramento.*

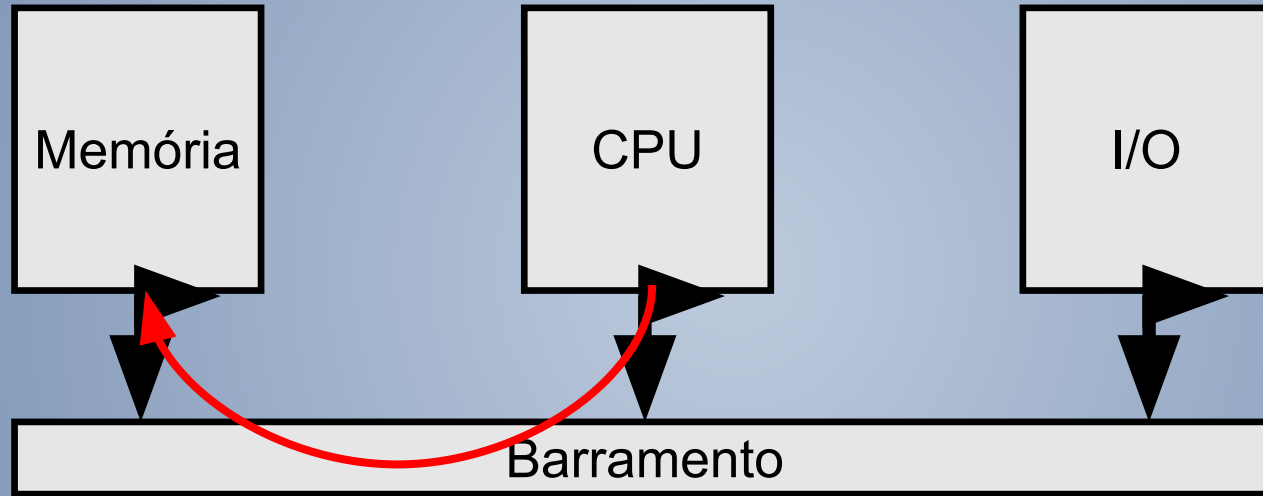
A CPU irá verificar agora qual a instrução ele recebeu...



A CPU agora já sabe qual a instrução
ela recebeu, e começa a preparar
para executar essa instrução

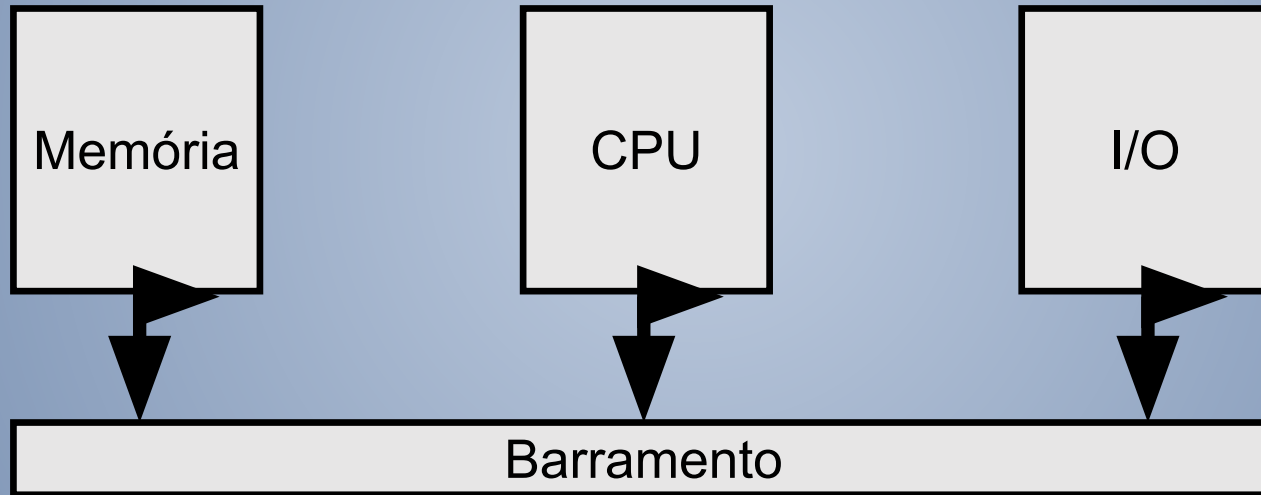


Um desses passos pode ser requisitar alguns dados que estão na memória...

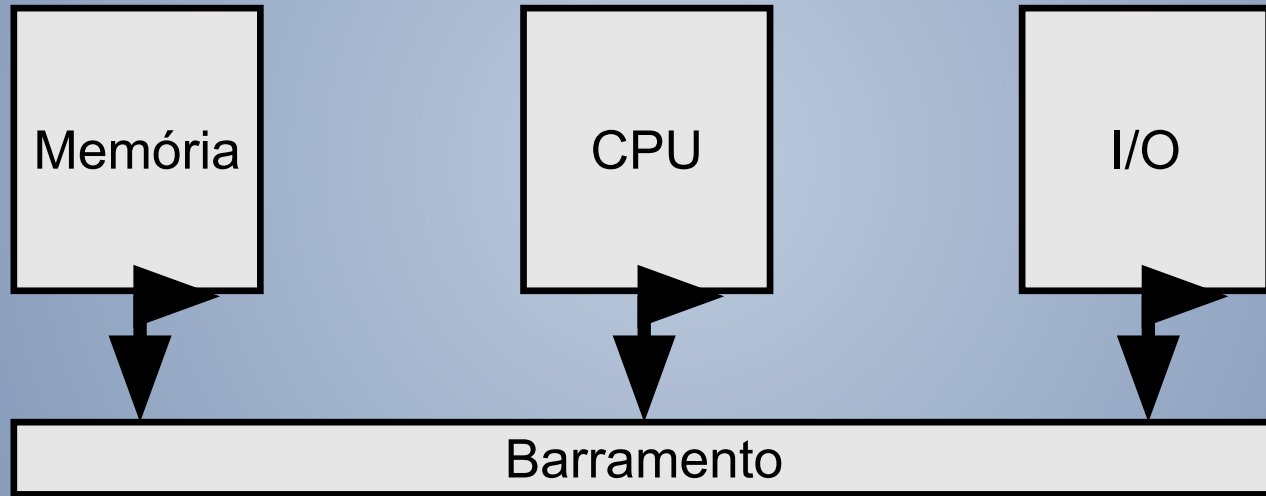


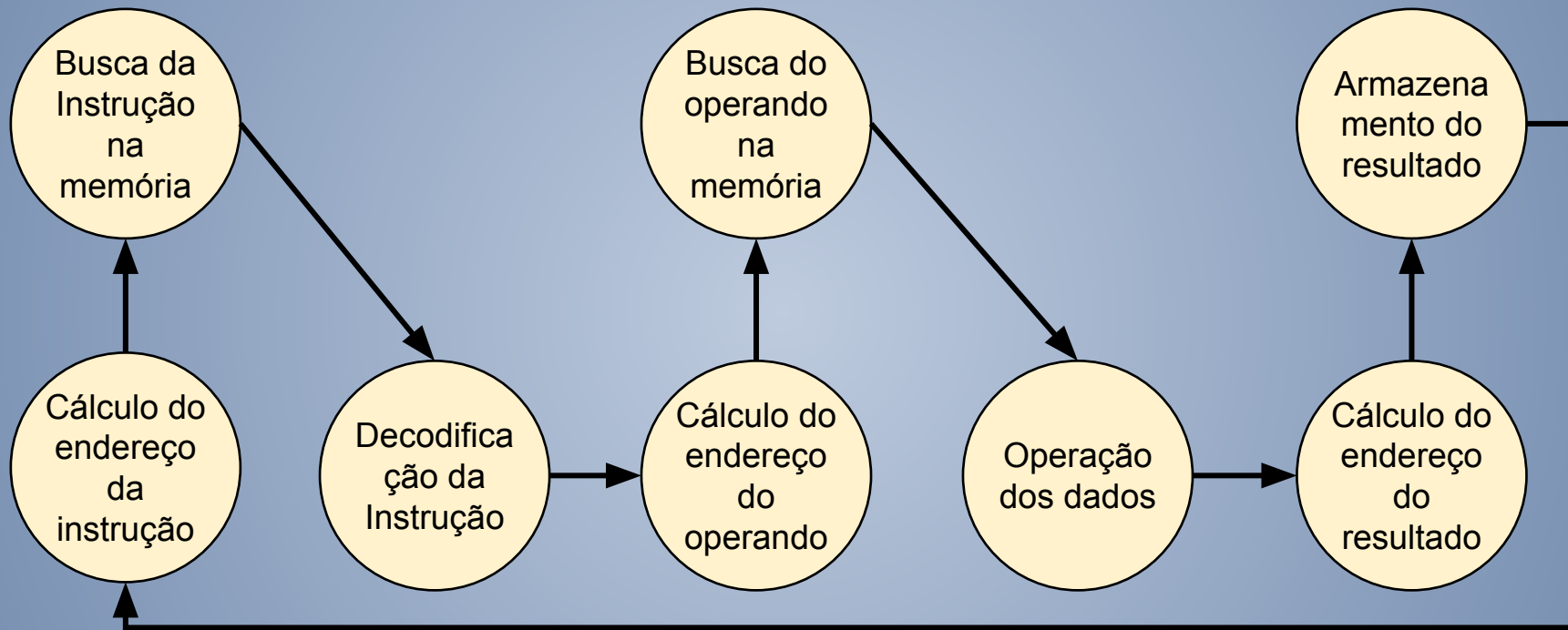
*O processo de requisição
acontece de forma semelhante
ao da instrução.*

A outra etapa é executar propriamente a instrução, **que é feita na CPU.**



Repete-se todo o ciclo novamente....



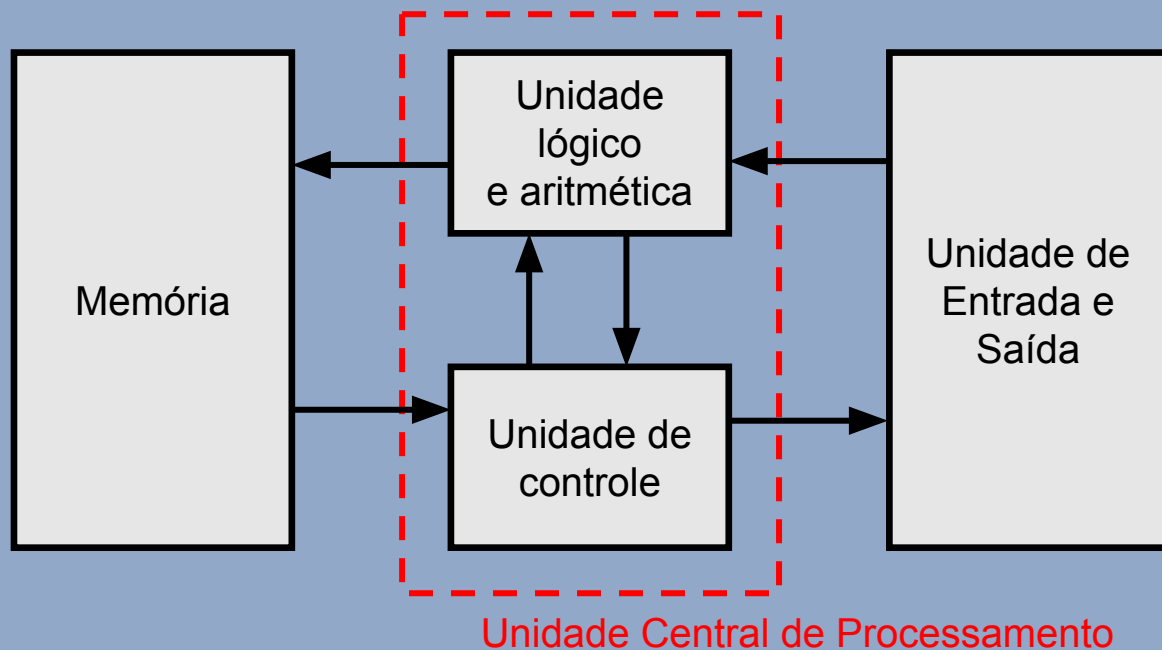


3

ESTRUTURA E COMPONENTES



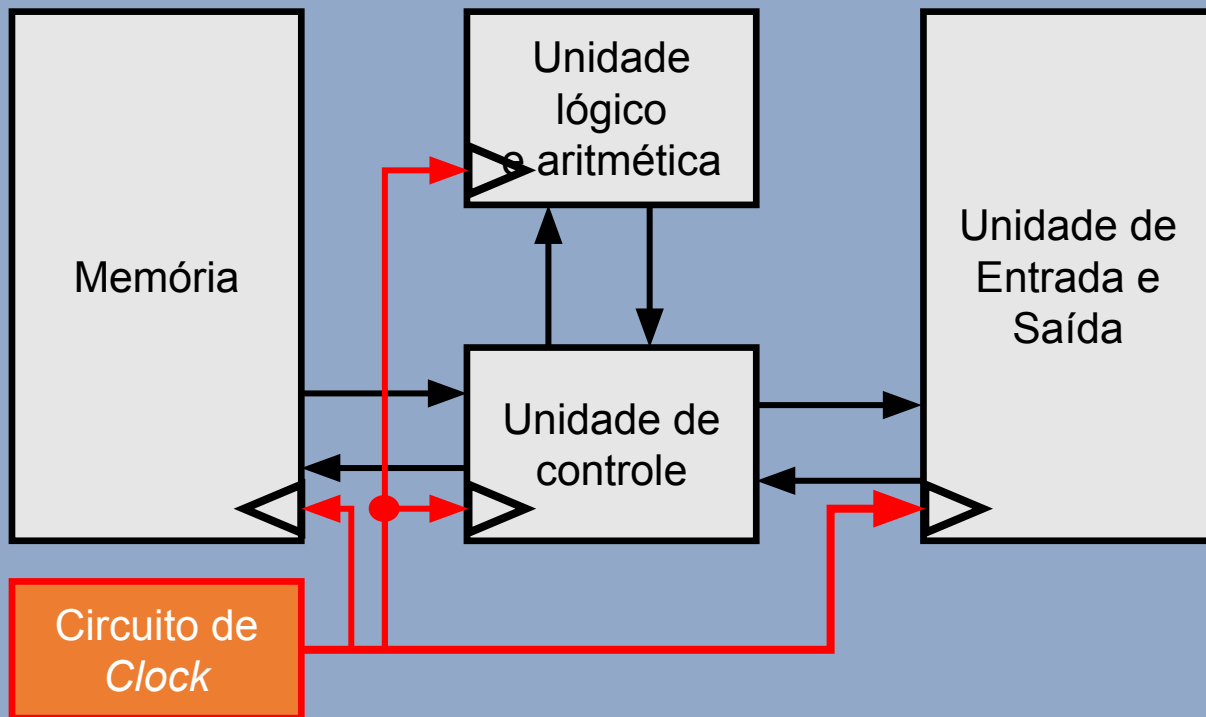
Estrutura



**E A
SINCRONIZAÇÃO?**

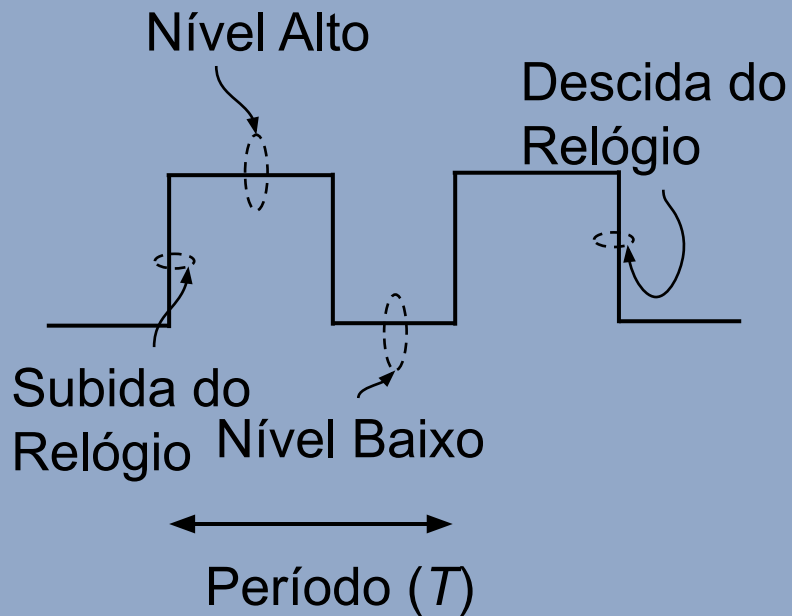


Estrutura com clock

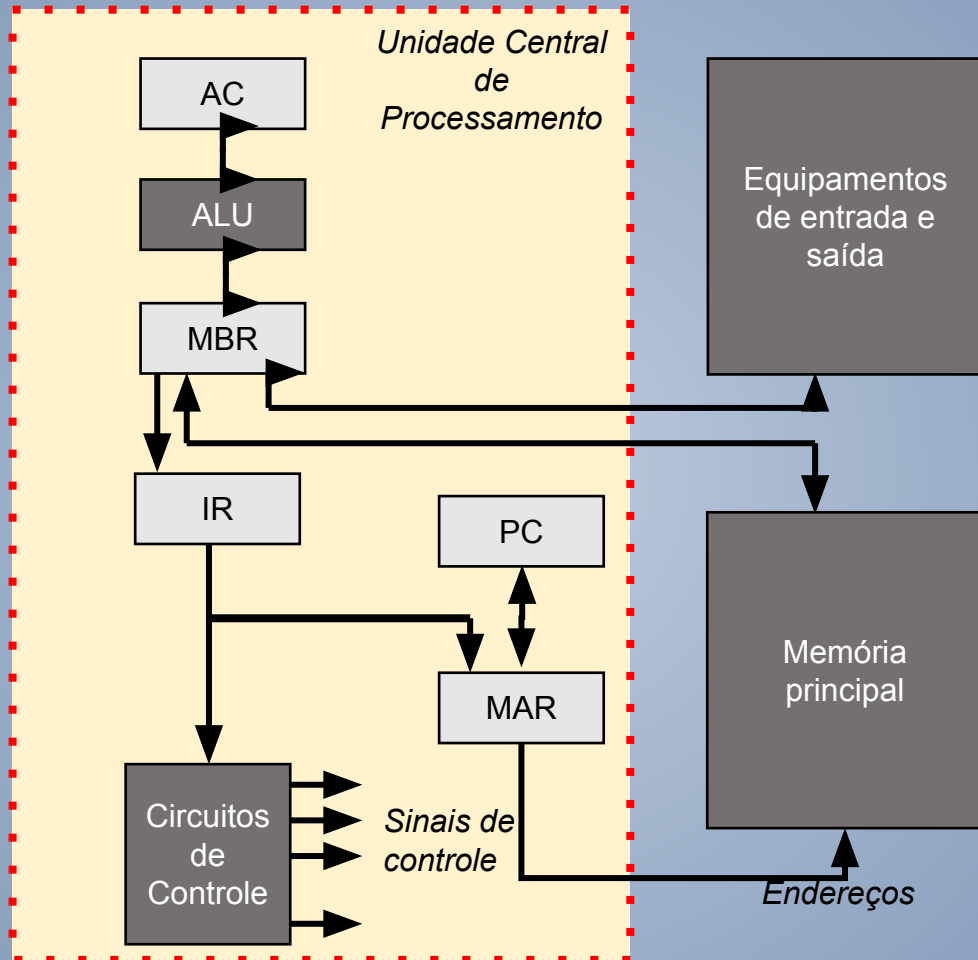




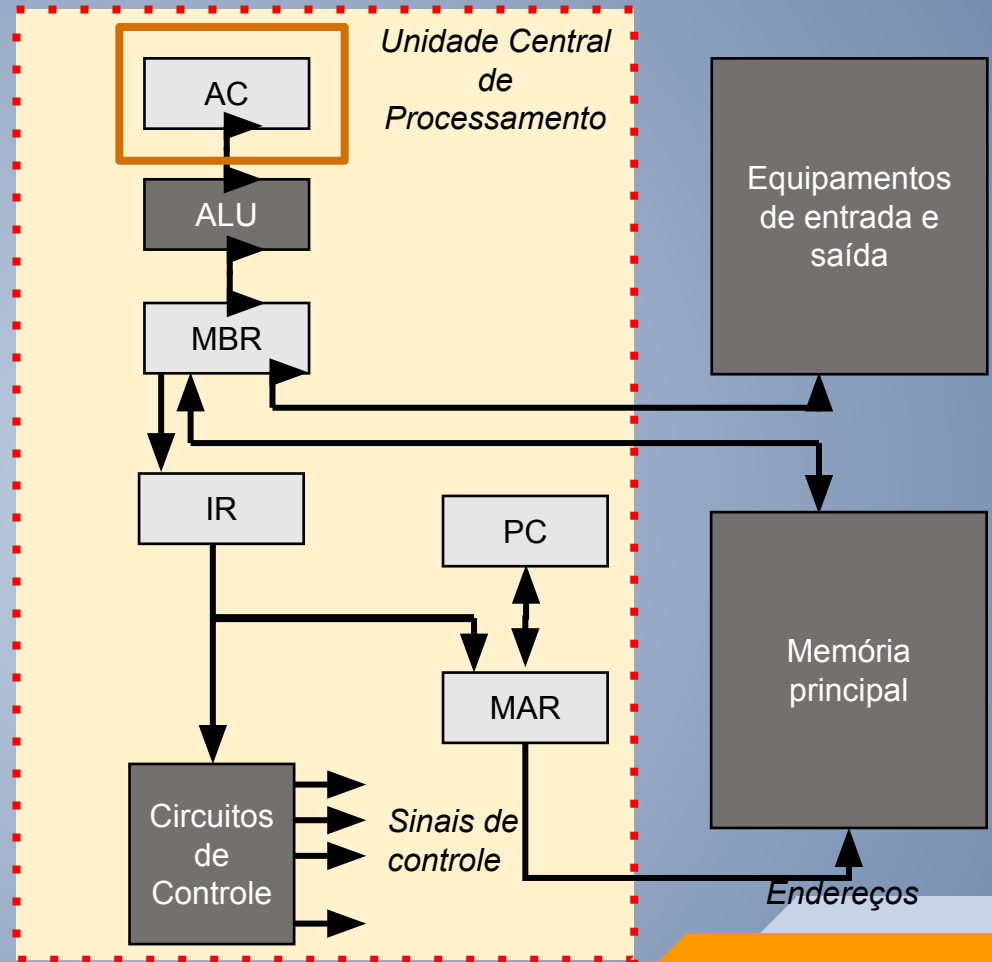
Clock



**DETALHANDO UM
POUCO MAIS A CPU**

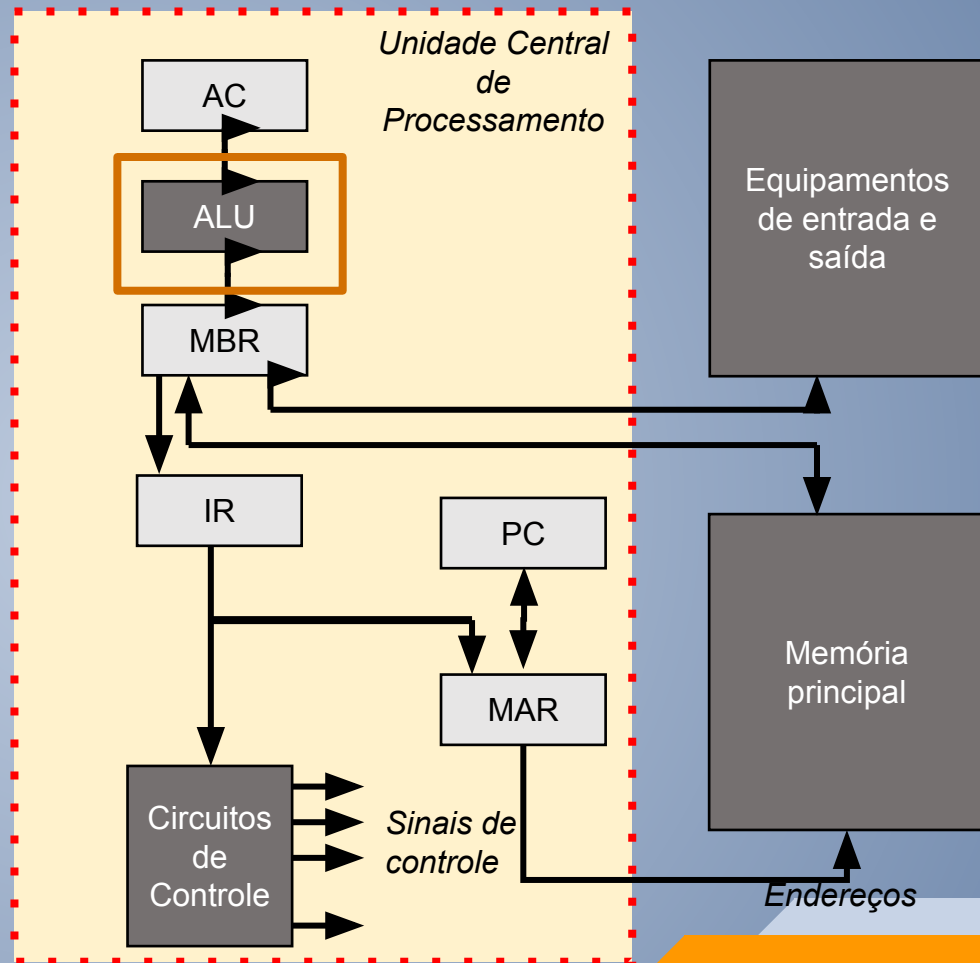


Acumulador (AC):
empregados para armazenar
temporariamente os
resultados a ALU;



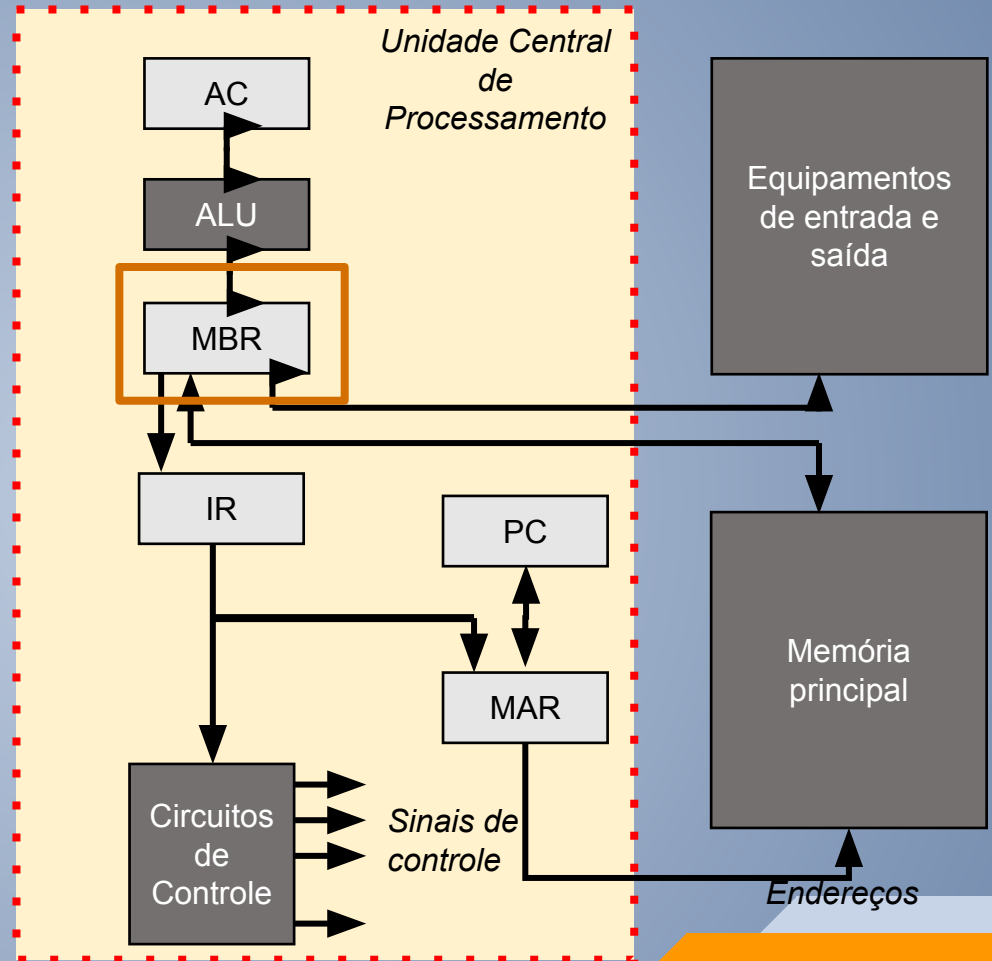
Unidade Lógico-Aritmética

(ALU): Realizar os cálculos de operação aritméticas e lógica sobre os dados;

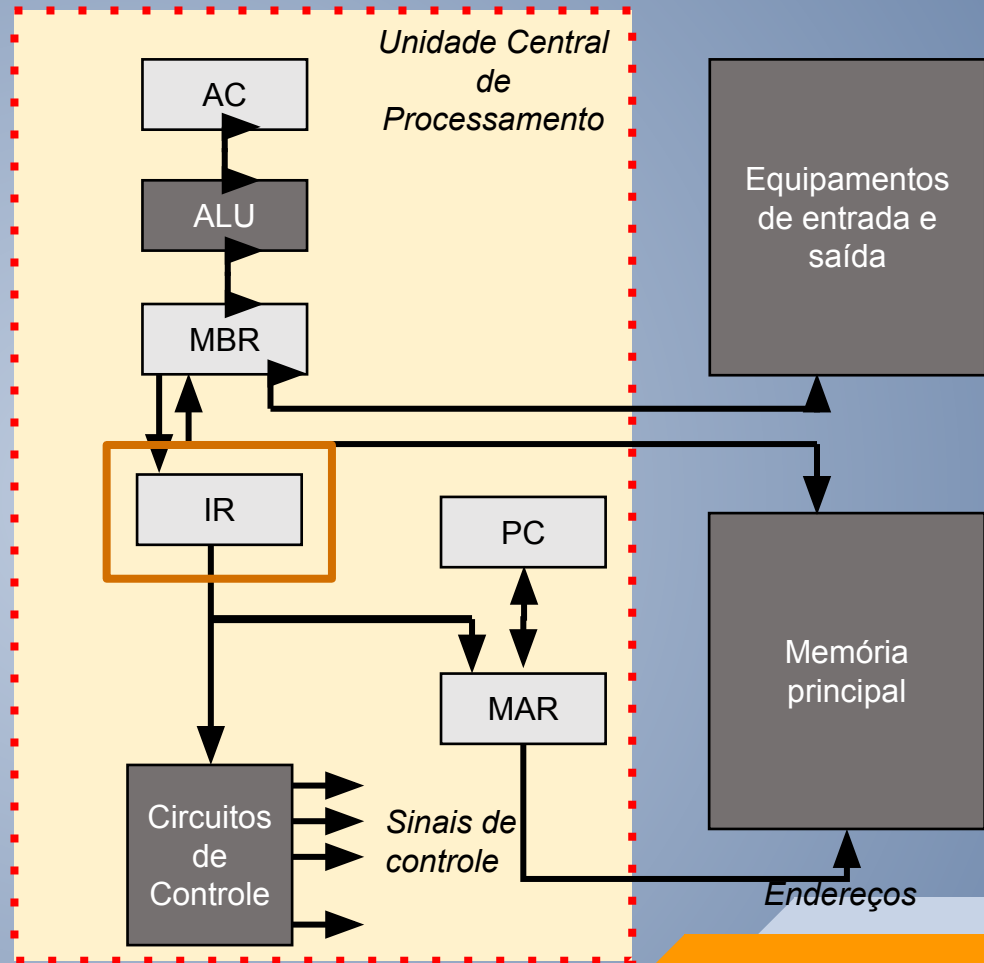


Registrador de buffer de memória (MBR): contém uma palavra a ser:

- Armazenada na memória;
- Enviada para E/S;
- Recebida da memória;
- Recebida da E/S.

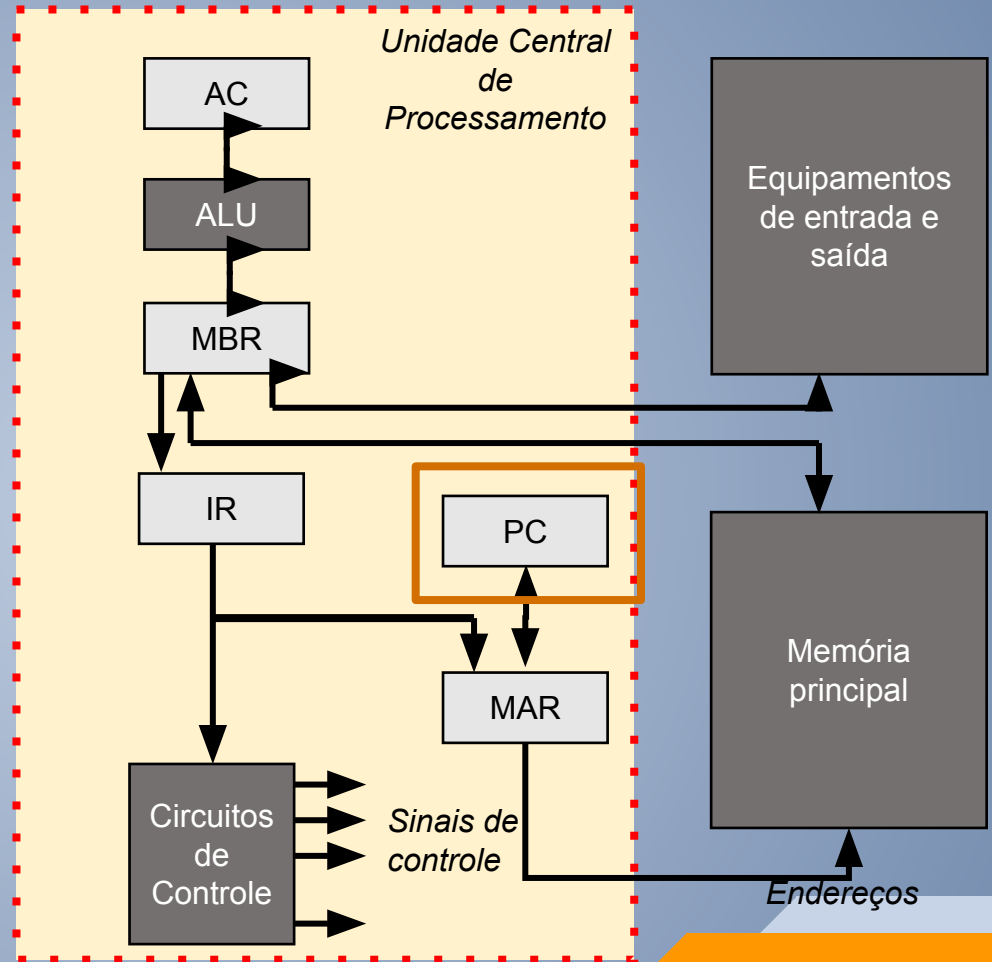


Registrador de instrução (IR):
contém a instrução (binário) a
ser executado pelo
computador.

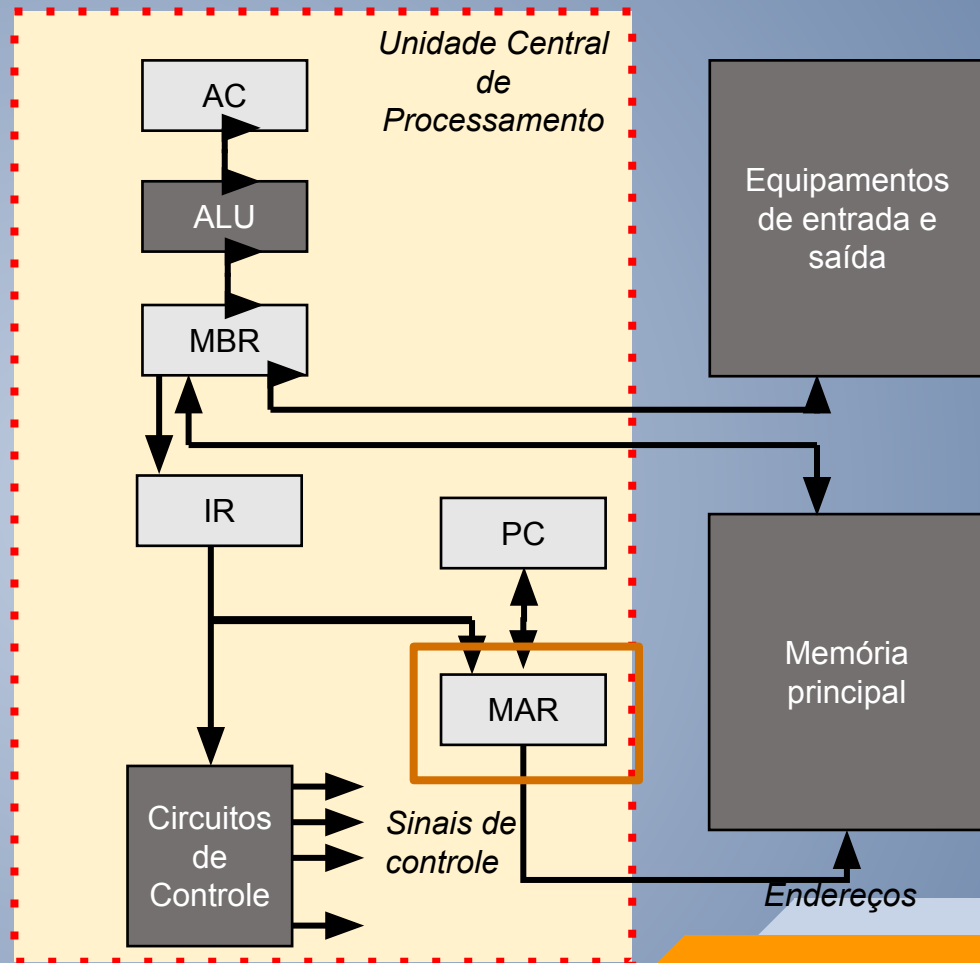


Contador de Programa (PC):

contém o endereço do próximo par de instruções a ser apanhado da memória;

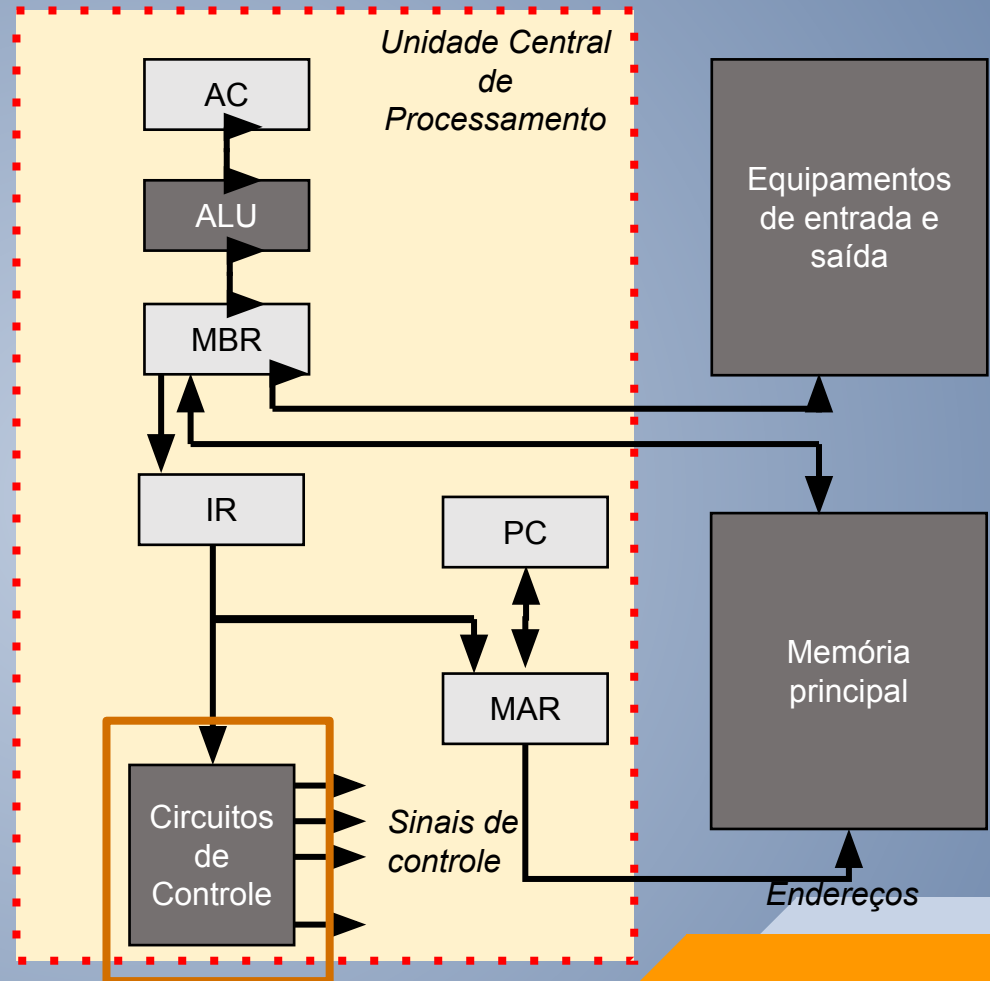


Registrador de endereço de memória (MAR): especifica o endereço na memória da palavra a ser escrita ou lida no MBR.

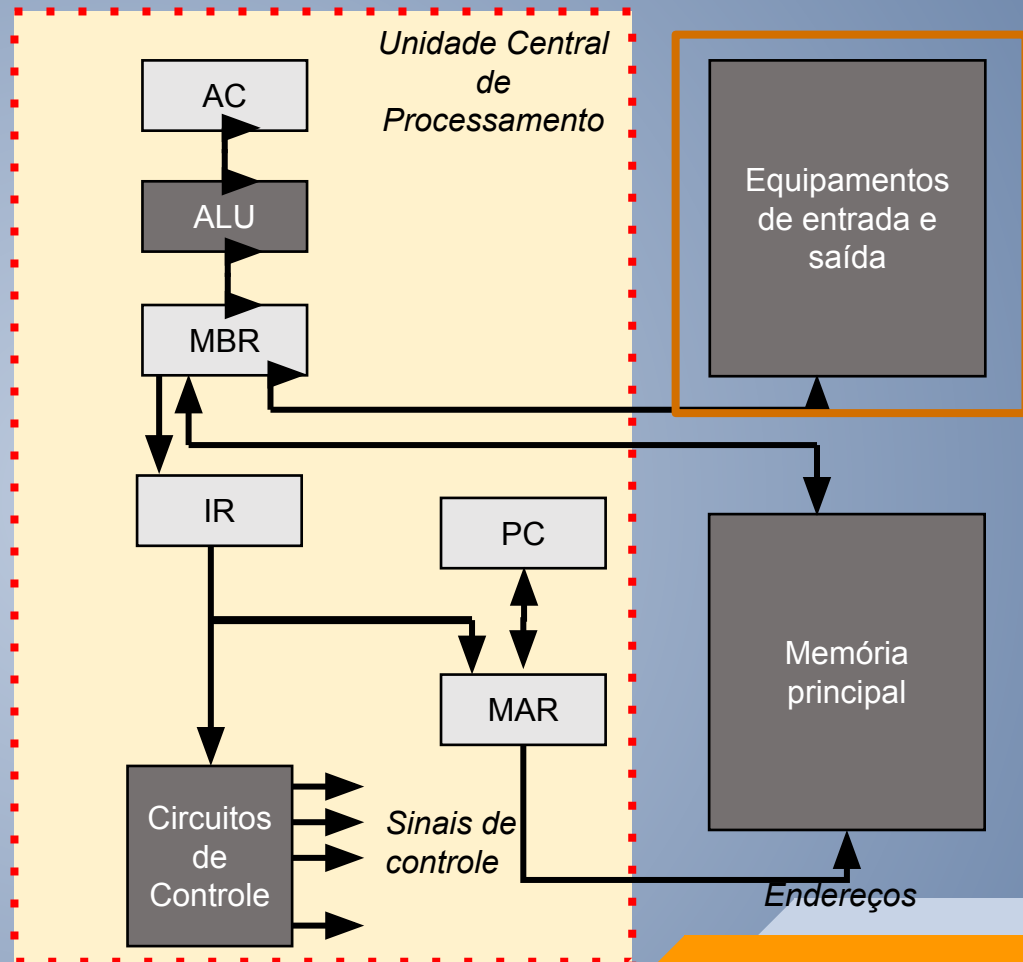


Circuitos de controle: interpreta o opcode (em IR) e executa uma instrução;

- Envio de sinais de controle adequados para que:
 - ▷ Os dados sejam movidos e/ou;
 - ▷ Uma operação seja executada na ALU.

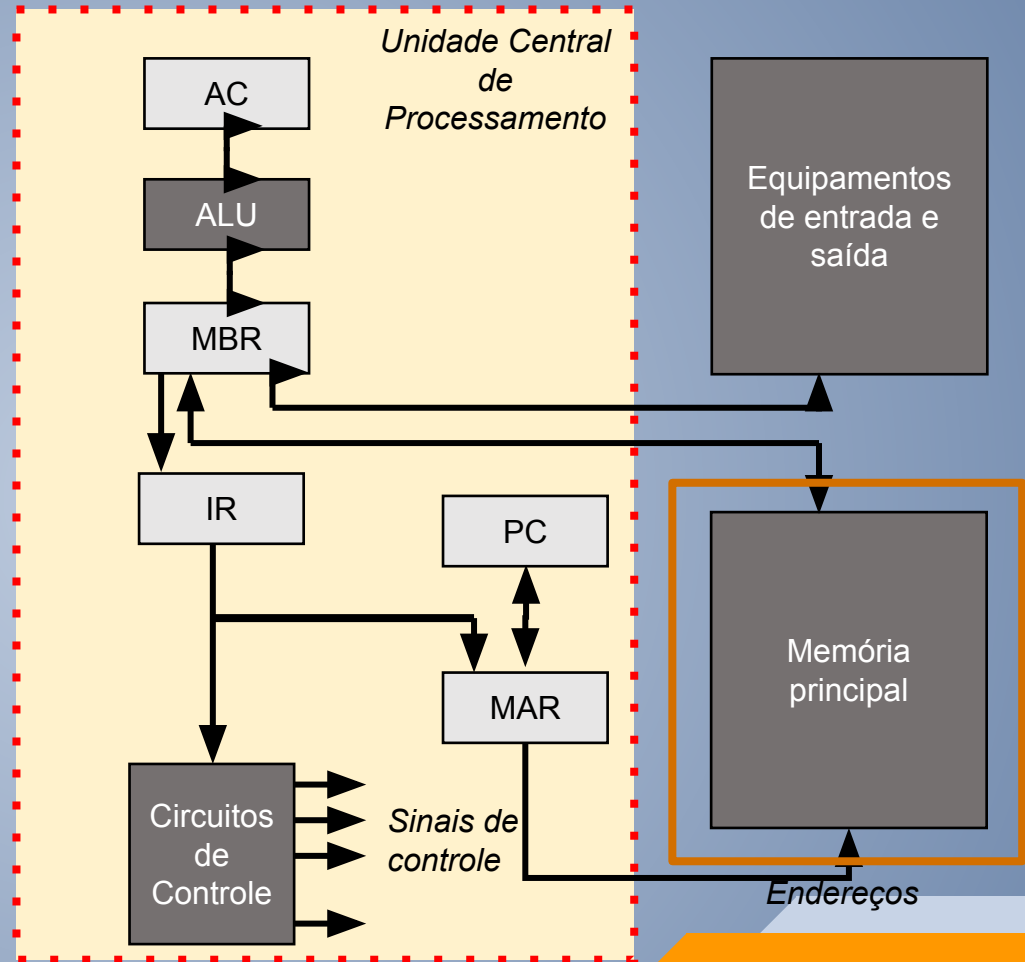


Equipamento de entrada e saída: realiza o comunicação do computador com o mundo externo;



Memória Principal (MP):

Armazena os programas e dados (que estão sendo utilizados no momento ou que ainda vão ser utilizados);





Registradores (memória de rascunho)

- **PC (Contador de Programa):** Endereço da próxima instrução a ser executada;
- **IR (Registrador de Instrução):** Instrução (em binário) que está sendo executado;
- **ACC (Acumulador):** Armazenamento temporário;
- **MBR (Buffer de Dado):** Contém o dado a ser lido ou escrito na memória;
- **MAR (Buffer de Endereço):** Contém o endereço em que vai ser escrito ou lido um dado na memória;



Exemplo de operação

Memória Principal

300	1940
301	5941
302	2941
	...
900	0003
901	002

CPU

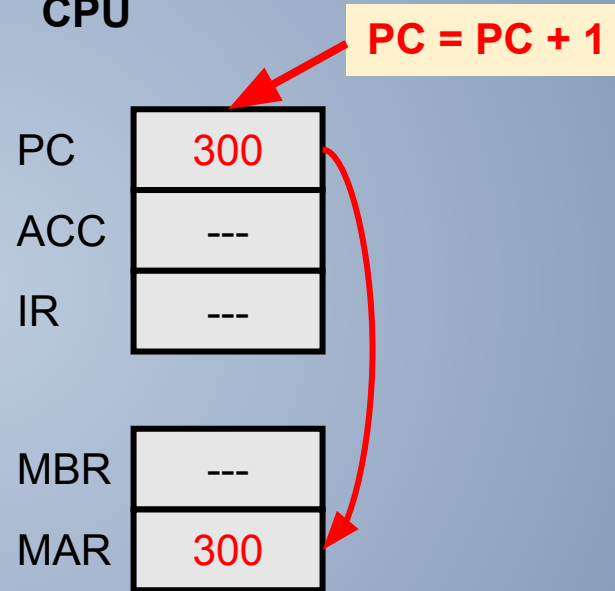
PC	300
ACC	---
IR	---
MBR	---
MAR	---

*Início da instrução no
endereço 300.*

Memória Principal

300	1940
301	5941
302	2941
	...
900	0003
901	002

CPU

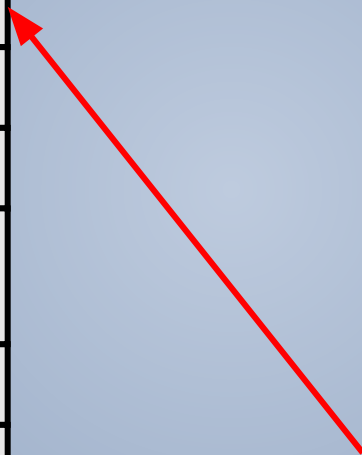


Memória Principal

300	1940
301	5941
302	2941
	...
900	0003
901	002

CPU

PC	301
ACC	---
IR	---
MBR	---
MAR	300

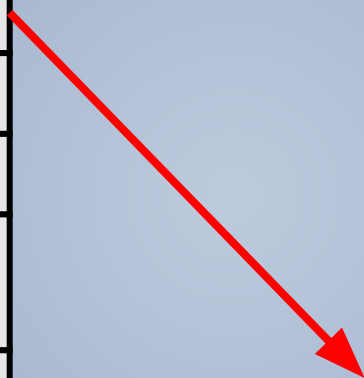


Memória Principal

300	1940
301	5941
302	2941
	...
900	0003
901	002

CPU

PC	301
ACC	---
IR	---
MBR	1940
MAR	300



Memória Principal

300	1940
301	5941
302	2941
	...
900	0003
901	002

CPU

PC	301
ACC	---
IR	1940
MBR	1940
MAR	300



Memória Principal

300	1940
301	5941
302	2941
	...
940	0003
941	002

CPU

PC	301
ACC	---
IR	1940
MBR	1940
MAR	300


*Decodificação da instrução:
Carregar o endereço 940
para ACC*

Memória Principal

300	1940
301	5941
302	2941
	...
940	0003
941	002

CPU

PC	301
ACC	---
IR	1940
MBR	1940
MAR	940



Memória Principal

300	1940
301	5941
302	2941
	...
940	0003
941	002

CPU

PC	301
ACC	---
IR	1940
MBR	1940
MAR	940



Memória Principal

300	1940
301	5941
302	2941
	...
940	0003
941	002

CPU

PC	301
ACC	---
IR	1940
MBR	0003
MAR	940



Memória Principal

300	1940
301	5941
302	2941
	...
940	0003
941	002

CPU

PC	301
ACC	0003
IR	1940
MBR	0003
MAR	940



*Fim da instrução no
endereço 300.*

Memória Principal

300	1940
301	5941
302	2941
	...
900	0003
901	002

CPU

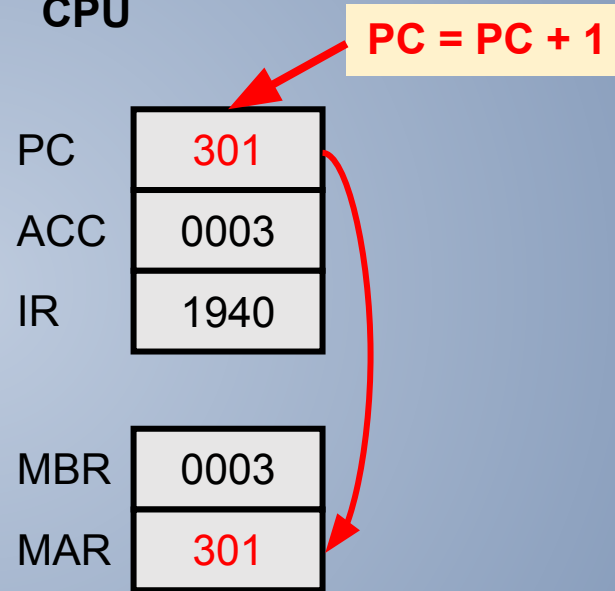
PC	301
ACC	0003
IR	1940
MBR	0003
MAR	940

Início da instrução no endereço 301.

Memória Principal

300	1940
301	5941
302	2941
	...
900	0003
901	002

CPU

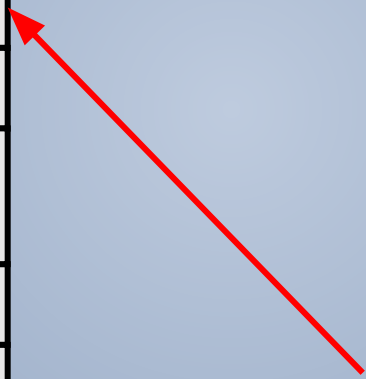


Memória Principal

300	1940
301	5941
302	2941
	...
900	0003
901	002

CPU

PC	302
ACC	0003
IR	1940
MBR	0003
MAR	301

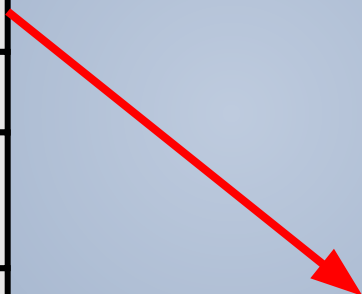


Memória Principal

300	1940
301	5941
302	2941
	...
900	0003
901	002

CPU

PC	302
ACC	0003
IR	1940
MBR	5941
MAR	301



Memória Principal

300	1940
301	5941
302	2941
	...
900	0003
901	002

CPU

PC	302
ACC	0003
IR	5941
MBR	5941
MAR	301



Memória Principal

300	1940
301	5941
302	2941
	...
940	0003
941	002

CPU

PC	302
ACC	0003
IR	5941
MBR	5941
MAR	301

*Decodificação da instrução:
Somar ACC com o endereço
941*

Memória Principal

300	1940
301	5941
302	2941
	...
940	0003
941	002

CPU

PC	302
ACC	0003
IR	5941
MBR	5941
MAR	941



Memória Principal

300	1940
301	5941
302	2941
	...
940	0003
941	002

CPU

PC	302
ACC	0003
IR	5941
MBR	5941
MAR	941

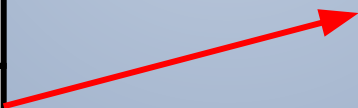


Memória Principal

300	1940
301	5941
302	2941
	...
940	0003
941	0002

CPU

PC	302
ACC	0003
IR	5941
MBR	0002
MAR	941



Memória Principal

300	1940
301	5941
302	2941
	...
940	0003
941	002

CPU

PC	302
ACC	0003
IR	5941
MBR	0002
MAR	941

$0003 + 0002 = 0005$

Memória Principal

300	1940
301	5941
302	2941
	...
940	0003
941	002

CPU

PC	302
ACC	0005
IR	5941
MBR	0002
MAR	941

*Fim da instrução no
endereço 301.*

Memória Principal

300	1940
301	5941
302	2941
	...
900	0003
901	002

CPU

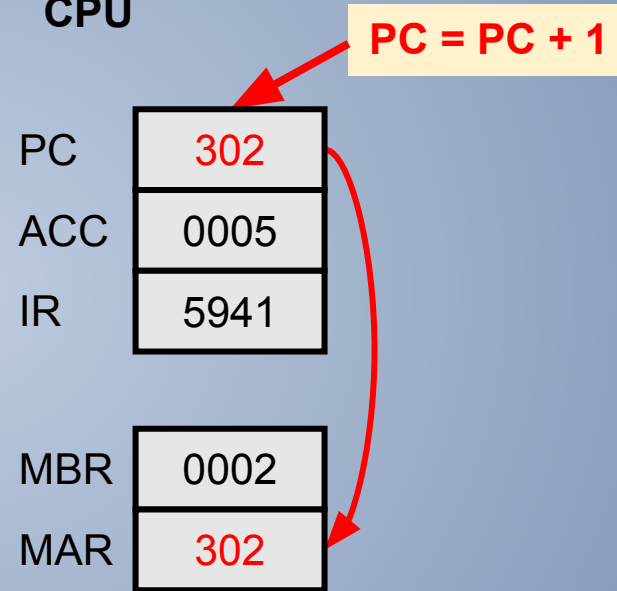
PC	302
ACC	0005
IR	5941
MBR	0002
MAR	941

Início da instrução no endereço 302.

Memória Principal

300	1940
301	5941
302	2941
	...
900	0003
901	002

CPU

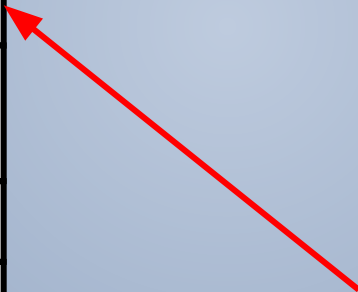


Memória Principal

300	1940
301	5941
302	2941
	...
900	0003
901	002

CPU

PC	303
ACC	0005
IR	5941
MBR	0002
MAR	302

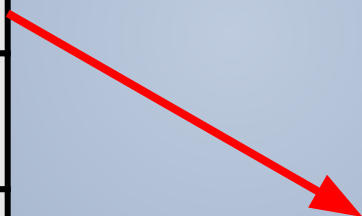


Memória Principal

300	1940
301	5941
302	2941
	...
900	0003
901	002

CPU

PC	303
ACC	0005
IR	5941
MBR	2941
MAR	302



Memória Principal

300	1940
301	5941
302	2941
	...
900	0003
901	002

CPU

PC	303
ACC	0005
IR	2941
MBR	2941
MAR	302



Memória Principal

300	1940
301	5941
302	2941
	...
940	0003
941	002

CPU

PC	303
ACC	0005
IR	2941
MBR	2941
MAR	302


*Decodificação da instrução:
Salvar ACC na posição 941
De memória*

Memória Principal

300	1940
301	5941
302	2941
	...
940	0003
941	002

CPU

PC	303
ACC	0005
IR	2941
MBR	2941
MAR	941



Memória Principal

300	1940
301	5941
302	2941
	...
940	0003
941	002

CPU

PC	303
ACC	0005
IR	2941
MBR	0005
MAR	941



Memória Principal

300	1940
301	5941
302	2941
	...
940	0003
941	0005

CPU

PC	303
ACC	0005
IR	2941
MBR	0005
MAR	941

*Fim da instrução no
endereço 302.*

4

INTERPRETAÇÃO DE INSTRUÇÃO



CISC *versus* RISC

- Interpretação por microprogramação (software);
 - ▷ CISC (Complex Instruction Set Computer);
- Interpretação por hardware;
 - ▷ RISC (Reduced Instruction Set Computer);



CISC (Complex Instruction Set Computer)

- A interpretação das instruções acontece por meio de um software interno ao processador (Microcódigo).
- A introdução do microcódigo foi interessante no início da computação, pois a memória era cara e pequena;
- Cada instrução deveria fazer muito, de modo que o programa completo tivesse poucas instruções;
- A criação de novas instruções é, na maioria das vezes, quase sem custo e sem aumento de espaço do processador;

- As instruções não são executadas diretamente pelo hardware;
- Instruções com largura variável;
- As instruções requerem múltiplos ciclos de relógio para sua execução;
- A inclusão de novas instruções é feita pela mudança no microcódigo.



RISC (Reduced Instruction Set Computer)

- Em sistemas CISC, certas operações complexas tendem a ser mais lentas do que uma sequência de operações mais simples fazendo a mesma coisa;
- Muitos programas do mundo real passam a maior parte do seu tempo executando operações simples;
- Todas as instruções implementadas em hardware.
- Se um processador RISC dispõem de uma determinada instrução, ele deve, necessariamente, conter blocos de hardware para executar aquela operação;

- Todas as instruções possuem tamanho fixo;
- São mais previsíveis e rápidos, dado que todas as instruções são executadas em um ciclo de máquina;
- Custam menos para serem produzidos;
- São mais fáceis de projetar e menos suscetíveis a erros de projeto;
- A inclusão de uma nova instrução é feita através da mudança do hardware;

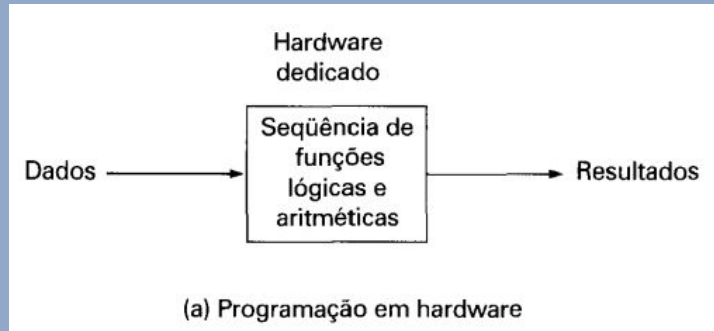
5

ARQUITETURA DE PROGRAMAÇÃO E ACESSO À MEMÓRIA



Programação em hardware

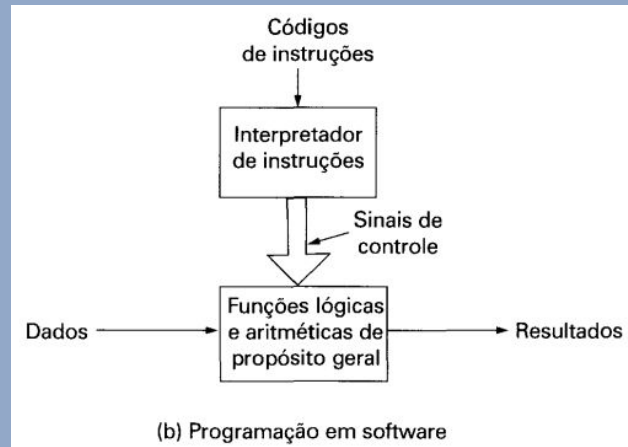
- É possível obter, para cada aplicação particular, uma configuração de componentes lógicos (hardware) projetada especificamente para executar essa aplicação.





Programação em software

- Também é possível construirmos uma configuração de funções lógicas e aritméticas de propósito geral.
- Em cada passo no programa (software) alguma operação lógica ou aritmética é executada sobre algum dado.



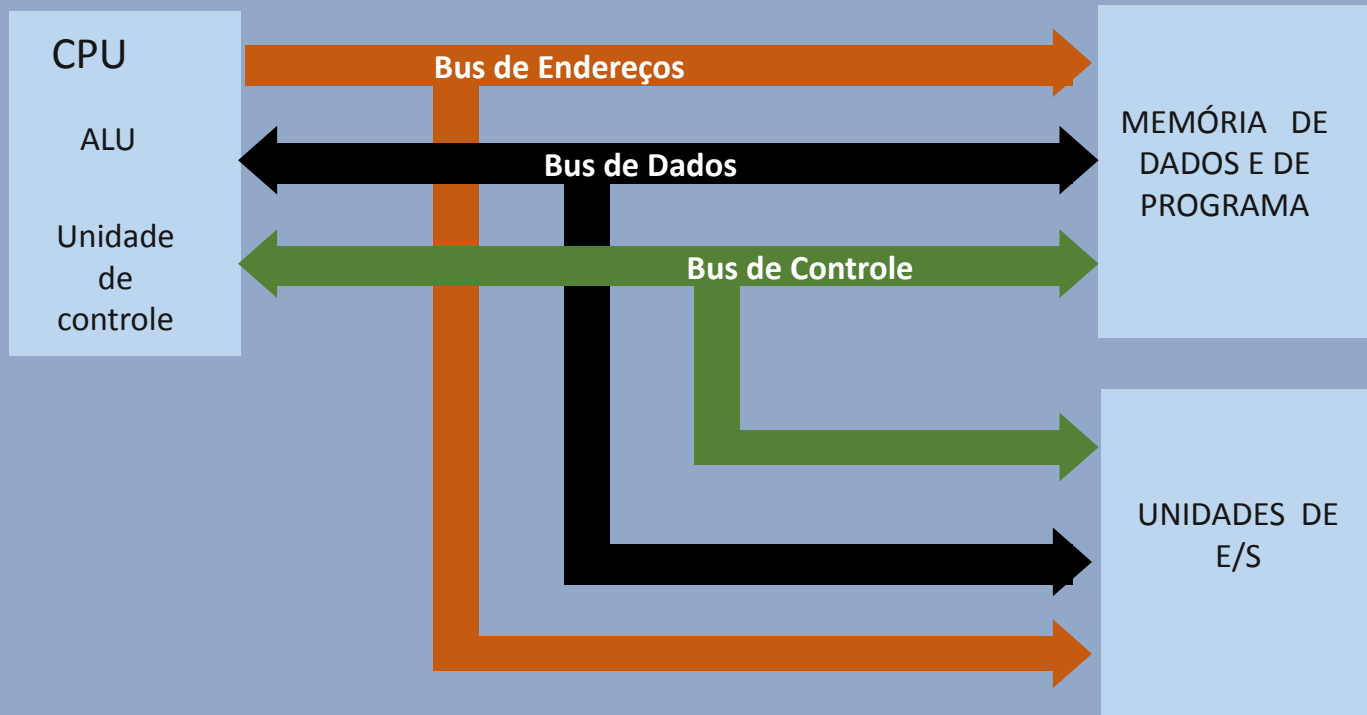


Arquitetura de Von Neumman

- Programação em software.
- Os dados e as instruções são armazenados em uma única memória de leitura e escrita.
- O conteúdo dessa memória é endereçado pela sua posição, independentemente do tipo de dados nela contidos.
- A execução de instruções ocorre de modo sequencial (exceto quando essa sequência é explicitamente alterada de uma instrução para a seguinte).



Arquitetura de Von Neumman



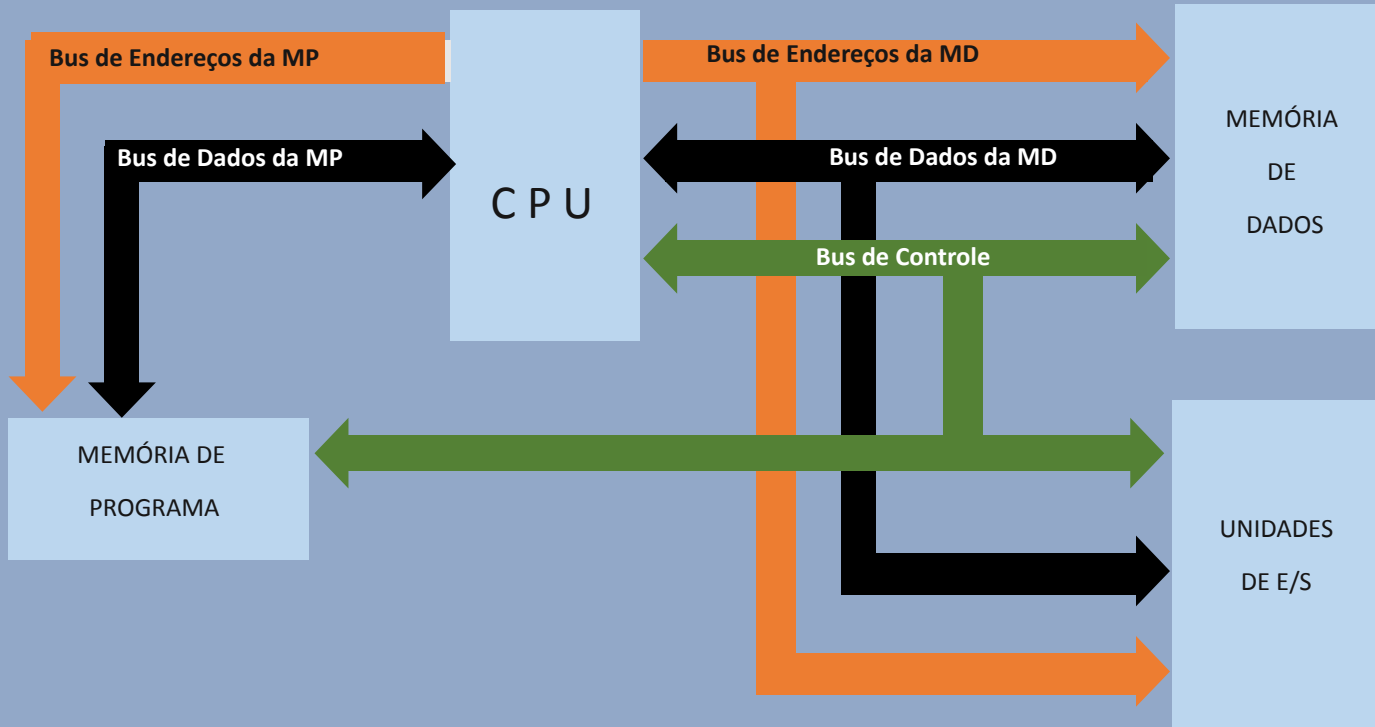


Arquitetura de Harvard

- Presença de duas memórias diferentes:
 - ▷ Memória de dados;
 - ▷ Memória de instrução (ou memória de programa);
- Existem dois barramentos distintos - um em que só trafegam os dados e o outro que só trafega instruções;
- Esta proposta surgiu devido a necessidade de se pôr o processador para trabalhar mais rápido;
 - ▷ Pode-se buscar uma instrução enquanto busca-se os operandos da instrução anterior;



Arquitetura de Harvard





Harvard *versus* Von Neumman

Característica	Harvard	Von Neumman
Complexidade	Mais complexo.	Menos complexo.
Velocidade	Mais rápido.	Mais lento.
Interpretação	Geralmente RISC.	Geralmente CISC.

6

MICROPROCESSADORES E MICROCONTROLADORES



Microprocessador

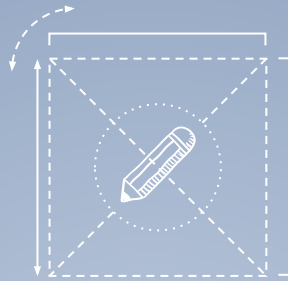
- Os computadores digitais utilizam o microprocessador para executar e controlar as suas ações.
- O microprocessador não é um sistema completo, pois necessita de elementos de memória, controles de E/S, barramentos, entre outros.
- É qualquer componente que implementa no chip as funções de uma unidade central de processamento (é uma CPU em um chip).



Microcontrolador

- É qualquer componente que incorpore on chip todas as unidades de um microcomputador: CPU, memória e portas de E/S.
- Exemplos de sistemas que utilizam microcontroladores:
 - ▷ Automação residencial.
 - ▷ Computadores de bordo.

EXERCÍCIOS



1. Esquematize e descreva a unidade central de processamento e seus componentes.
2. Justifique o motivo do MAR ser unidirecional?
3. Descreva os três estados básicos (busca, decodificação e execução) do ciclo de instrução no processador.
4. Qual a grande vantagem da arquitetura de Von Neumman que implementa programação em software?



DÚVIDAS?

sara.negreiros@ufersa.edu.br