

# **ORGANIZAÇÃO E ARQUITETURA DE COMPUTADORES I**

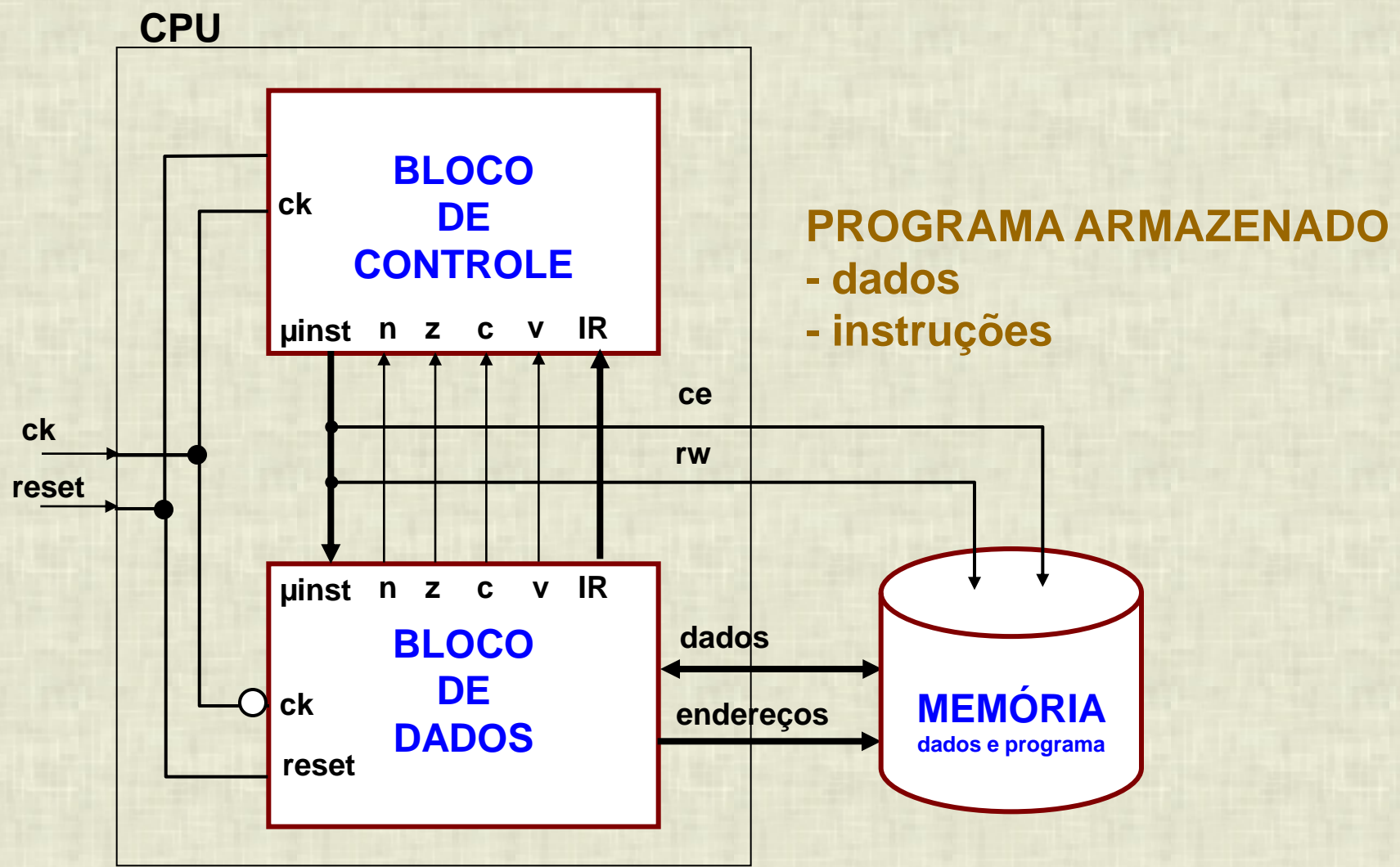
---

## **Computador Cleópatra**

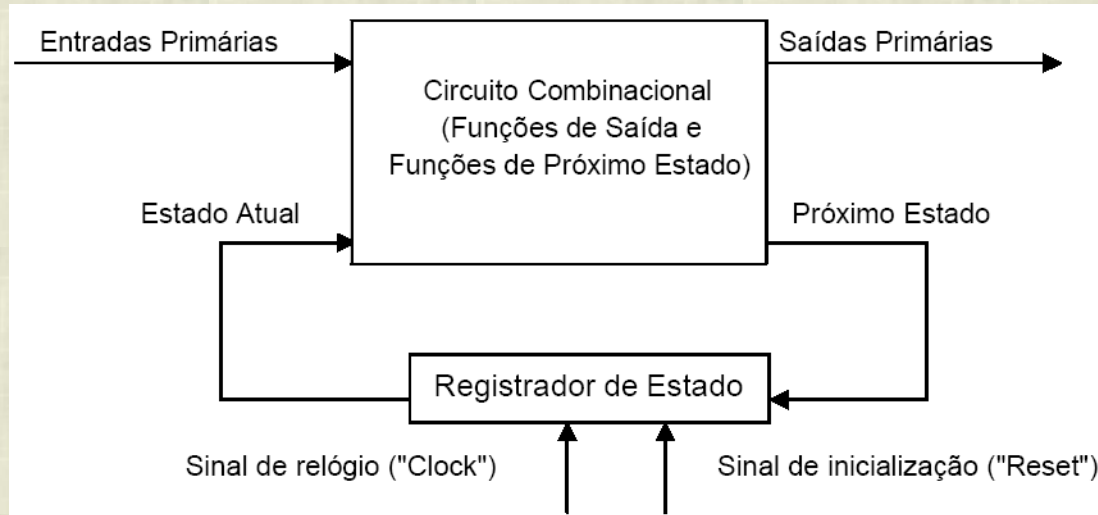
### **Arquitetura e Programação**

**prof. Dr. César Augusto M. Marcon**  
**prof. Dr. Edson Ifarraguirre Moreno**

# Arquitetura Cleópatra - Von Neumann

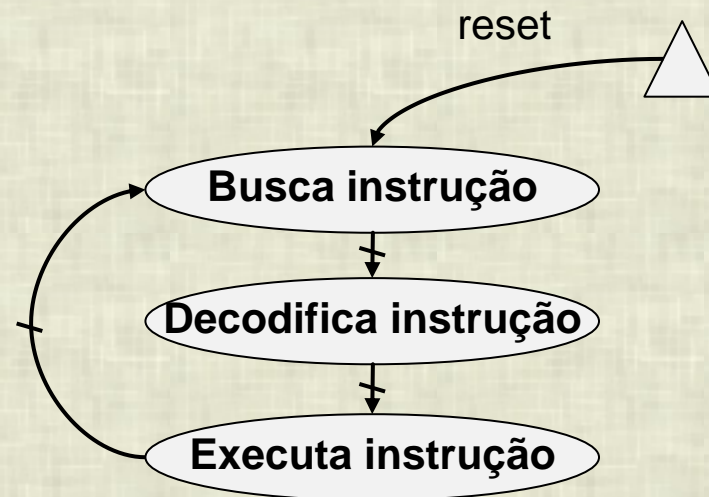


# Bloco de Controle - Circuito Seqüencial

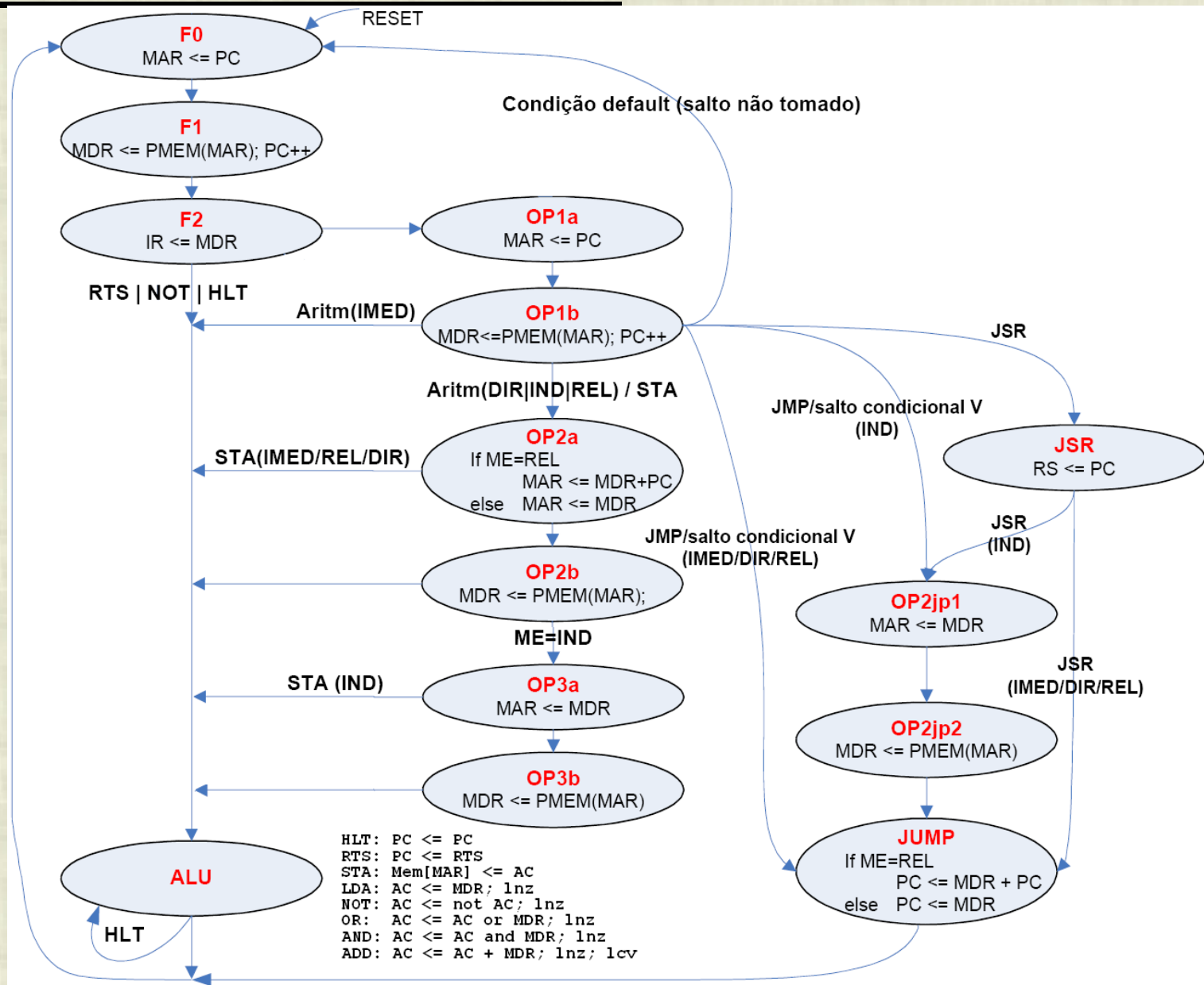


ck **Bloco de Controle**

$\mu$ inst n z c v IR

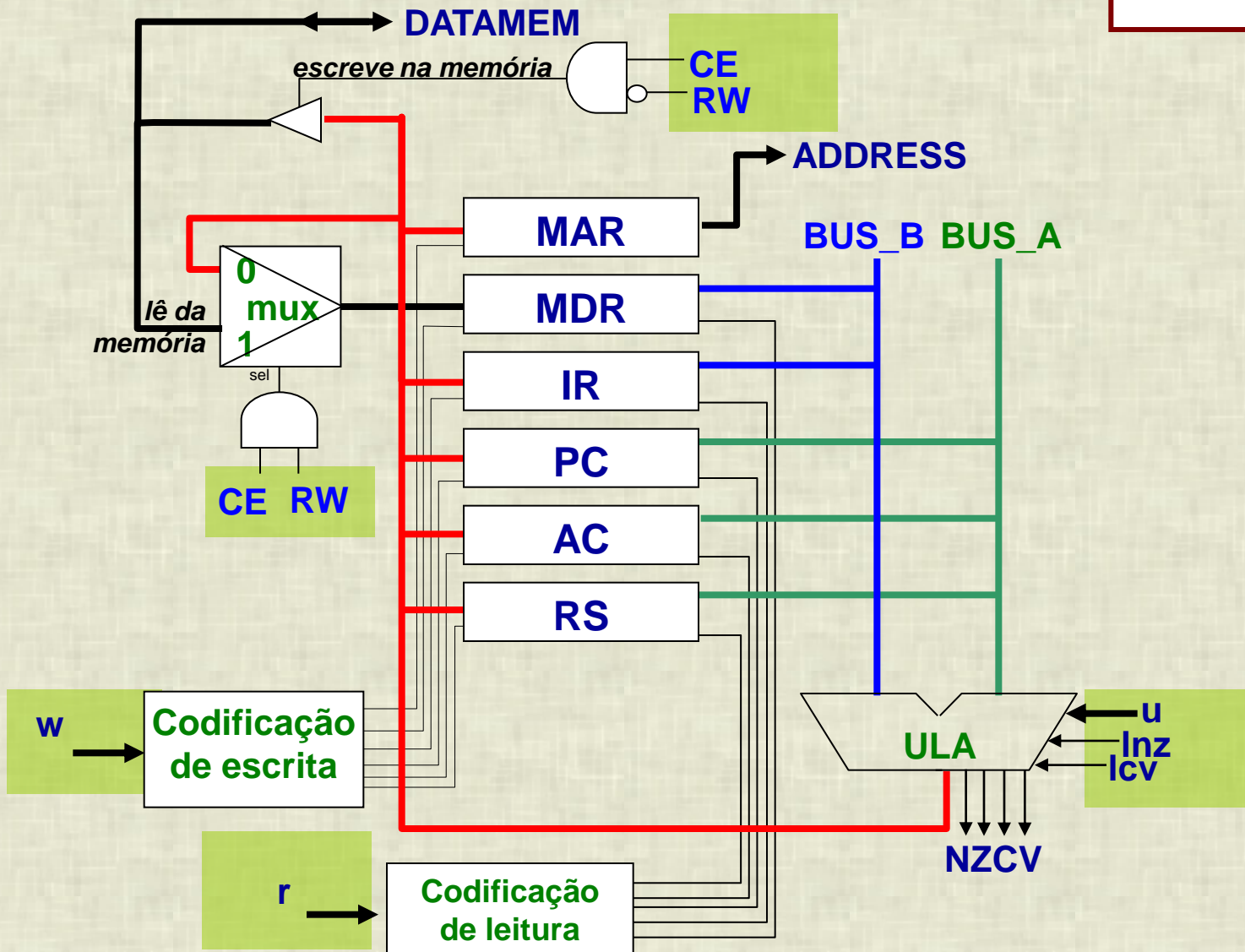


# Bloco de Controle - Máquina de Estados Finita



# Bloco de Dados

μinst	n	z	c	v	IR
ck	Bloco de Dados				
reset					



# Modo de Endereçamento

- **Imediato**

- Operando é o próprio dado
- Usado para representar constantes
  - Ex.: Número **45** do trecho de código C

- **Direto**

- Operando é endereço do dado
- Usado para representar variáveis
  - Ex. Variável **entrada** do trecho de código C

- **Indireto**

- Operando é endereço do endereço do dado
- Usado para representar ponteiros
  - Ex. Ponteiro **\*p** (que aponta para valor) do trecho de código C

- **Relativo**

- Operando é deslocamento de 8 bits em complemento de 2, para ser adicionado ao valor atual do PC, gerando o endereço do dado
- Usado em código relocável

```
int valor;  
int *p = &valor;  
int entrada = 8;  
  
*p = entrada + 45;
```

# Instruções

Mnemônico	Operação
NOT	Complementa (inverte) todos os bits de AC.
STA oper	Armazena AC na memória dada por oper.
LDA oper	Carrega AC com conteúdos de memória da posição dada por oper.
ADD oper	Adiciona AC ao conteúdo da memória dada por oper.
OR oper	Realiza OU lógico do AC com conteúdo da memória dada por oper.
AND oper	Realiza E lógico do AC com conteúdo da memória dada por oper.
JMP oper	PC recebe dado especificado por oper (desvio incondicional).
JC oper	Se C=1, então PC recebe valor dado por oper (desvio condicional).
JV oper	Se V=1, então PC recebe valor dado por oper (desvio condicional).
JN oper	Se N=1 então PC recebe valor dado por oper (desvio condicional).
JZ oper	Se Z=1, então PC recebe valor dado por oper (desvio condicional).
JSR oper	RS recebe conteúdo de PC e PC recebe dado de oper (subrotina).
RTS	PC recebe conteúdos de RS (retorno de subrotina).
HLT	Suspende processo de busca e execução de instruções.

# Exemplo de Programa Assembly

- Identificar diretivas, instruções e rótulos!

## Memória de Programa

Rótulo	Mnem	Operando
.CODE		
	ORG	#00H
INIT:	LDA	END1
	ADD	END2,I
	STA	END3
	AND	#0FH
	JZ	FIM,R
	NOT	
FIM:	HLT	
.ENDCODE		

## Memória de Dados

Rótulo	Mnem	Operando
.DATA		
	ORG	#90H
END1:	DB	#30H
END2:	DB	#END4
END3:	DB	#00H
END4:	DB	#5BH
.ENDDATA		



# Exercícios

---

1. Fazer a descrição assembly de um programa que contendo duas variáveis **A** e **B**, com valores 13 e 16, respectivamente, efetua a soma das mesmas. O resultado deve ser colocado na variável **C**
2. Fazer um programa que calcula o ou-exclusivo das variáveis **A** e **B** acima, sendo o resultado colocado na variável **C** (dica: utilize soma de produtos para fazer o ou-exclusivo)
3. Fazer um programa que, a partir das variáveis **A** e **B** acima, coloca em **C** a maior das duas
4. Fazer um programa que calcula o somatório de 3 a 15. O programa TEM que fazer este cálculo utilizando um laço com variável de controle. O resultado deve ser colocado na variável **total**
5. Fazer um programa que calcula o fatorial de uma variável **A**, colocando o resultado na variável **fat**. (observação: não deve ser feito com recursividade, apenas laços)