

# Microprocessadores

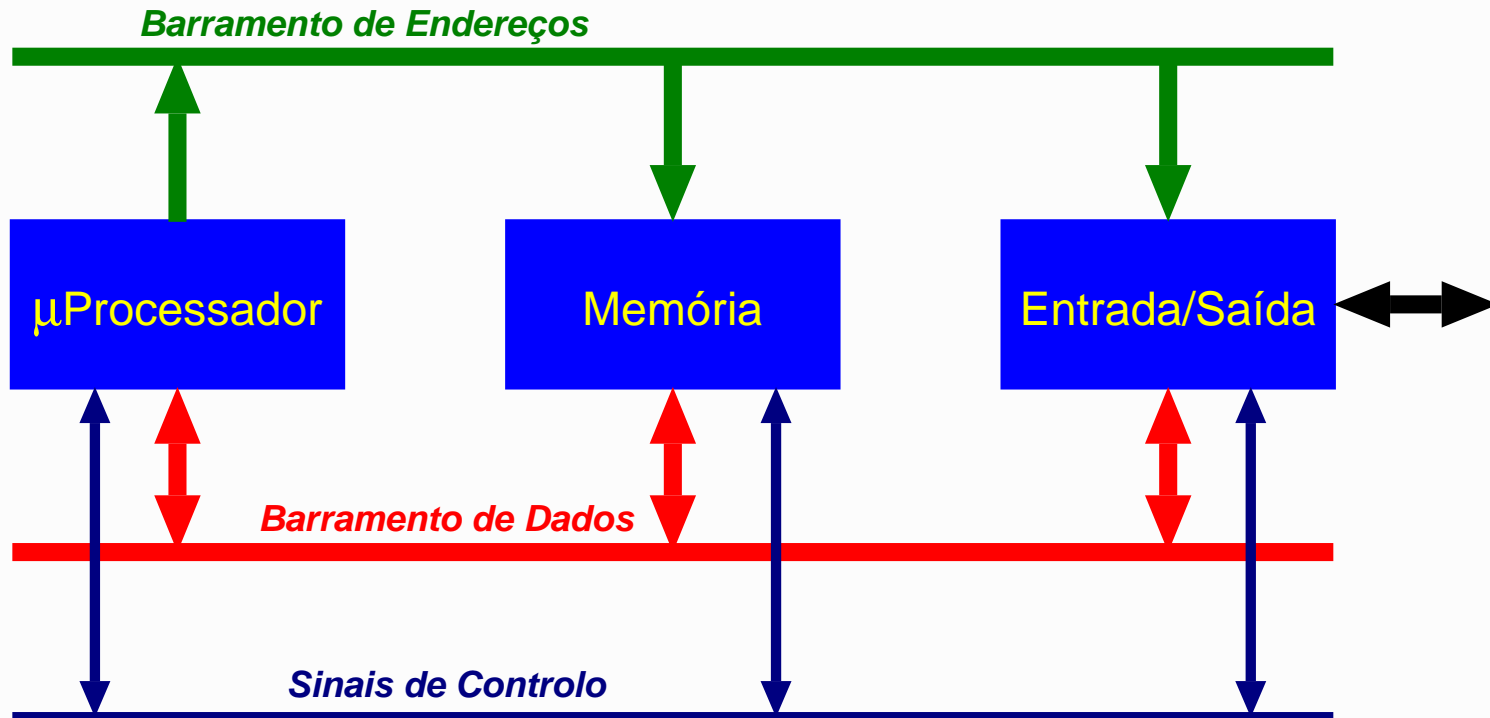
## Arquitectura Geral de Microprocessador

António M. Gonçalves Pinheiro

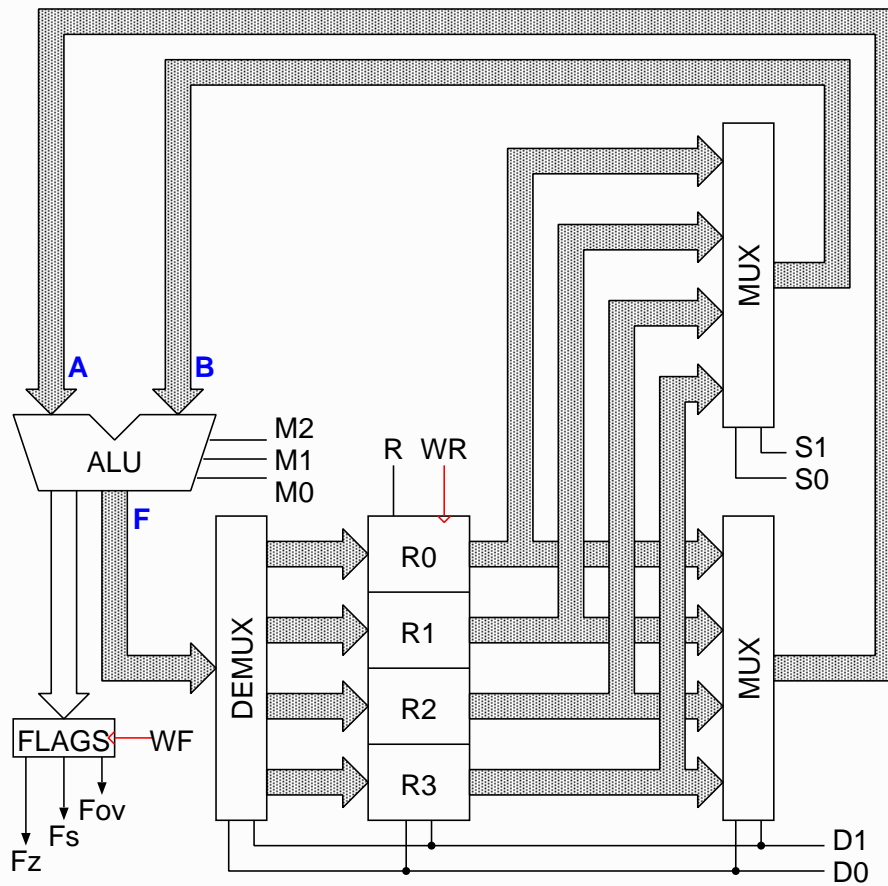
Departamento de Física  
Universidade da Beira Interior  
Covilhã - Portugal

pinheiro@ubi.pt

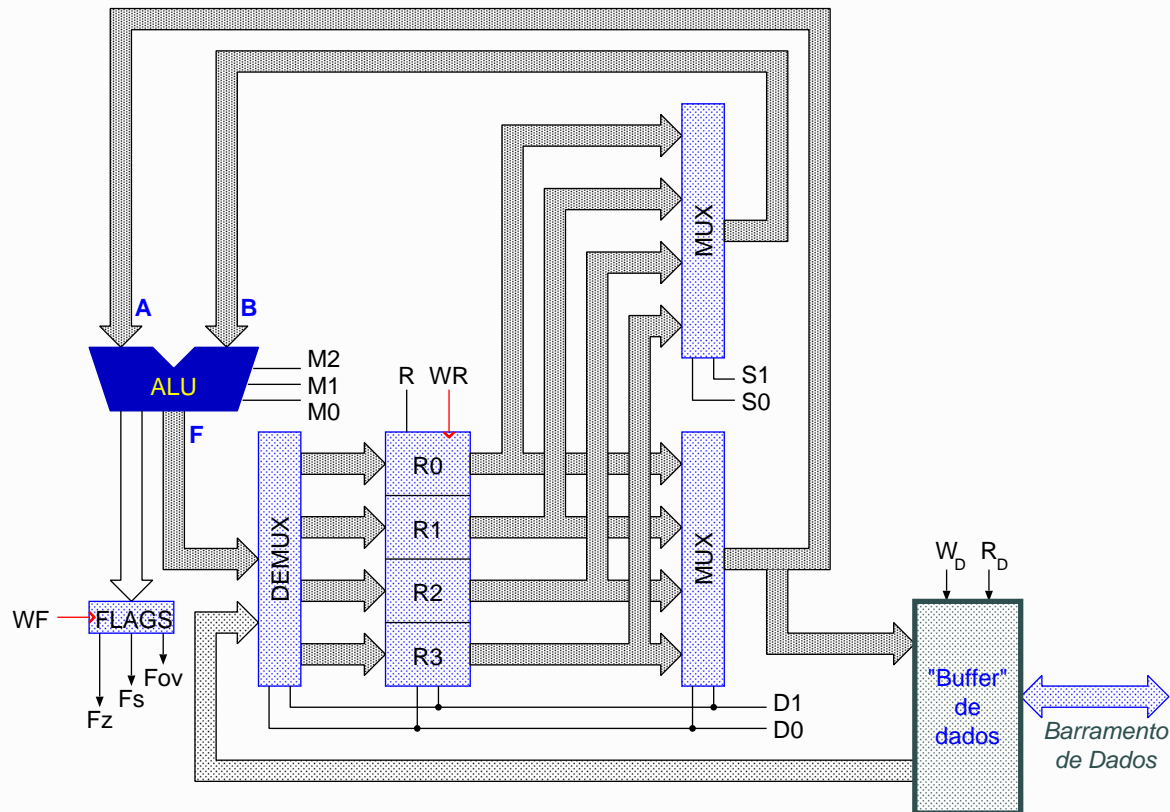
## Modelo de von Neuman



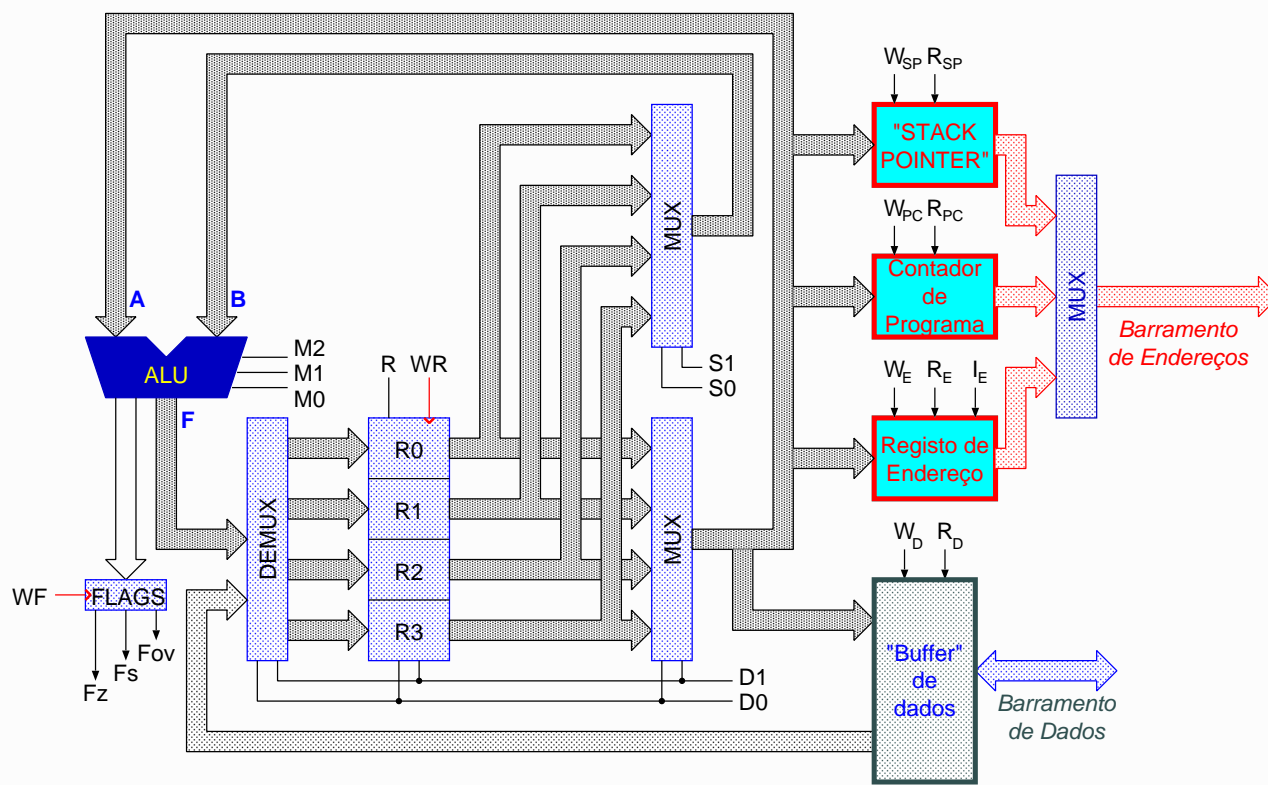
# Estrutura de $\mu$ Processadores



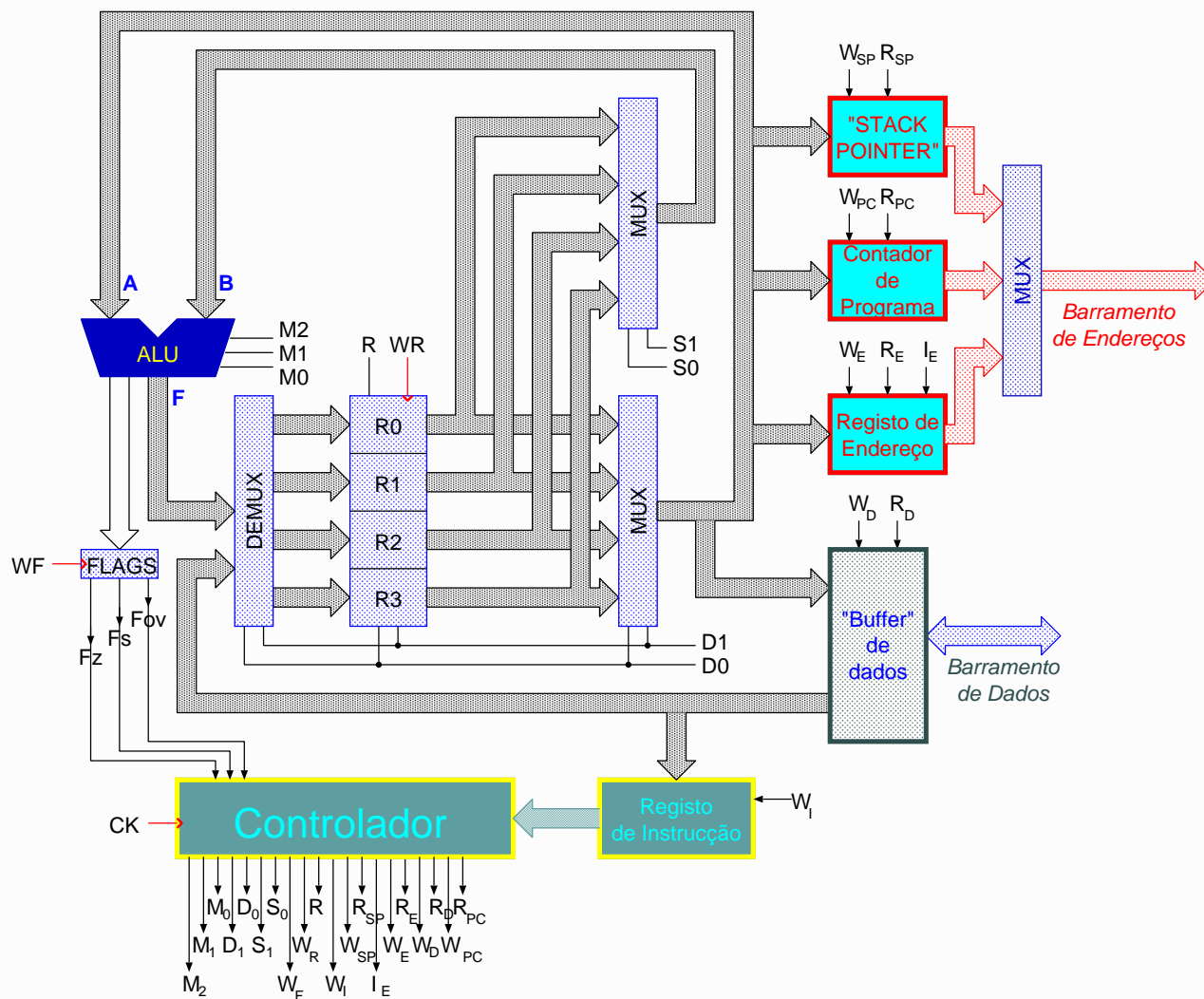
# Estrutura de $\mu$ Processadores



# Estrutura de $\mu$ Processadores



# Estrutura de $\mu$ Processadores



## ALU - Unidade Lógica e Aritmética

$M_2$	$M_1$	$M_0$	OPERAÇÃO
0	0	0	$A + B$
0	0	1	$A - B$
0	1	0	$A + 1$
0	1	1	$A - 1$
1	0	0	$A \text{ AND } B$
1	0	1	$A \text{ OR } B$
1	1	0	$A \text{ EXOR } B$
1	1	1	$\text{NOT}(A)$

## Instruções

TRÊS GRUPOS:

- OPERAÇÕES DE MOVIMENTO DE DADOS
- OPERAÇÕES ARITMÉTICAS E LÓGICAS
- OPERAÇÕES DE CONTROLO DE PROGRAMA

EXEMPLO: INSTRUÇÕES DE 8 BITS:

$I_7$	$I_6$	TIPO DE OPERAÇÃO
0	-	ARITMÉTICAS E LÓGICAS
1	0	MOVIMENTO DE DADOS
1	1	CONTROLO DE PROGRAMA



## Instruções

### OPERAÇÕES DE MOVIMENTO DE DADOS

$I_7$	$I_6$	$I_5$	$I_4$	$I_3$	$I_2$	$I_1$	$I_0$	MNEMÓNICA	OPERAÇÃO
1	0	0	0	$s_1$	$s_0$	$d_1$	$d_0$	MOV $R_d, R_s$	MOVE CONTEÚDO DE $R_s$ PARA $R_d$
1	0	1	0	0	0	$d_1$	$d_0$	LD $R_d$	CARREGA $R_d$ COM CONTEÚDO DO ENDEREÇO
1	0	1	1	0	0	$s_1$	$s_0$	ST $R_s$	ARMAZENA $R_s$ NO ENDEREÇO
1	0	0	1	0	1	$s_1$	$s_0$	MOV $R_E, R_s$	MOVE $R_s$ PARA O REGISTO DE ENDEREÇO
1	0	0	1	1	0	$s_1$	$s_0$	MOV SP, $R_s$	MOVE $R_s$ PARA O "STACK POINTER"
1	0	0	1	1	1	$s_1$	$s_0$	MOV PC, $R_E$	MOVE REGISTO DE ENDEREÇO PARA O CONTADOR DE PROGRAMA



## Instruções

### OPERAÇÕES ARITMÉTICAS E LÓGICAS

I <sub>7</sub>	I <sub>6</sub>	I <sub>5</sub>	I <sub>4</sub>	I <sub>3</sub>	I <sub>2</sub>	I <sub>1</sub>	I <sub>0</sub>	MNEMÓNICA	OPERAÇÃO
0	0	0	0	0	0	0	0	NOP	“NO OPERATION”
0	0	1	0	s <sub>1</sub>	s <sub>0</sub>	d <sub>1</sub>	d <sub>0</sub>	ADD R <sub>d</sub> ,R <sub>s</sub>	$R_d \leftarrow R_d + R_s$
0	0	1	1	s <sub>1</sub>	s <sub>0</sub>	d <sub>1</sub>	d <sub>0</sub>	SUB R <sub>d</sub> ,R <sub>s</sub>	$R_d \leftarrow R_d - R_s$
0	0	0	0	1	1	d <sub>1</sub>	d <sub>0</sub>	INC R <sub>d</sub>	$R_d \leftarrow R_d + 1$
0	0	0	1	1	1	d <sub>1</sub>	d <sub>0</sub>	DEC R <sub>d</sub>	$R_d \leftarrow R_d - 1$
0	1	1	0	s <sub>1</sub>	s <sub>0</sub>	d <sub>1</sub>	d <sub>0</sub>	AND R <sub>d</sub> ,R <sub>s</sub>	$R_d \leftarrow R_d \text{ AND } R_s$
0	1	1	1	s <sub>1</sub>	s <sub>0</sub>	d <sub>1</sub>	d <sub>0</sub>	OR R <sub>d</sub> ,R <sub>s</sub>	$R_d \leftarrow R_d \text{ OR } R_s$
0	1	0	0	s <sub>1</sub>	s <sub>0</sub>	d <sub>1</sub>	d <sub>0</sub>	EXOR R <sub>d</sub> ,R <sub>s</sub>	$R_d \leftarrow R_d \text{ EXOR } R_s$
0	1	0	1	1	1	d <sub>1</sub>	d <sub>0</sub>	NOT R <sub>d</sub>	$R_d \leftarrow \text{NOT}(R_d)$



## Instruções

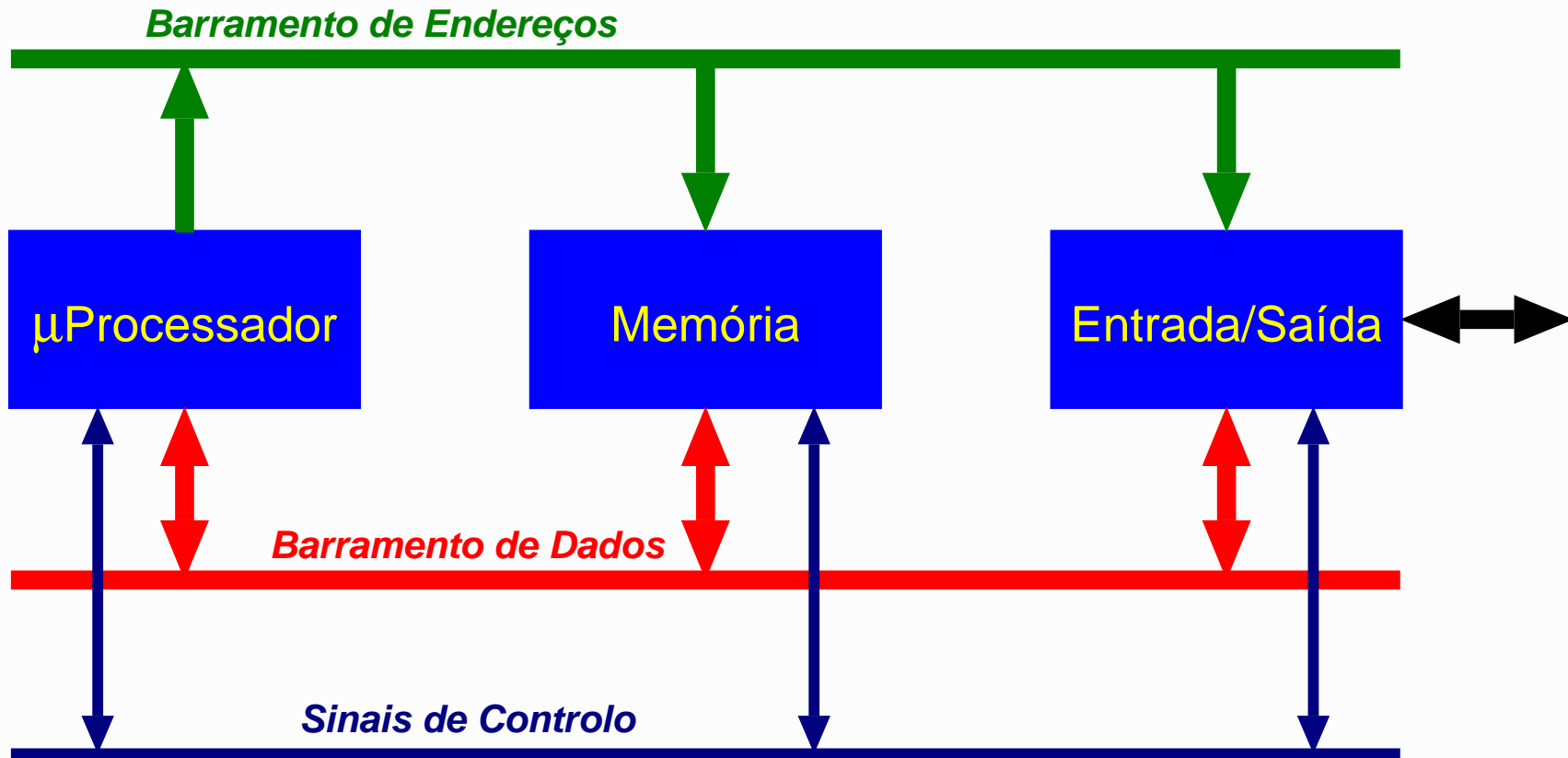
### OPERAÇÕES DE CONTROLO DE PROGRAMA

I <sub>7</sub>	I <sub>6</sub>	I <sub>5</sub>	I <sub>4</sub>	I <sub>3</sub>	I <sub>2</sub>	I <sub>1</sub>	I <sub>0</sub>	MNEMÓNICA	OPERAÇÃO
1	1	0	0	0	0	0	0	JMP	SALTO INCONDICIONAL
1	1	0	0	0	C <sub>2</sub>	C <sub>1</sub>	C <sub>0</sub>	JCC	SALTO CONDICIONAL
1	1	0	0	1	0	0	0	CALL	ROTINA INCONDICIONAL
1	1	0	0	1	C <sub>2</sub>	C <sub>1</sub>	C <sub>0</sub>	CCC	ROTINA CONDICIONAL
1	1	0	1	0	0	0	0	RET	RETORNO DE ROTINA



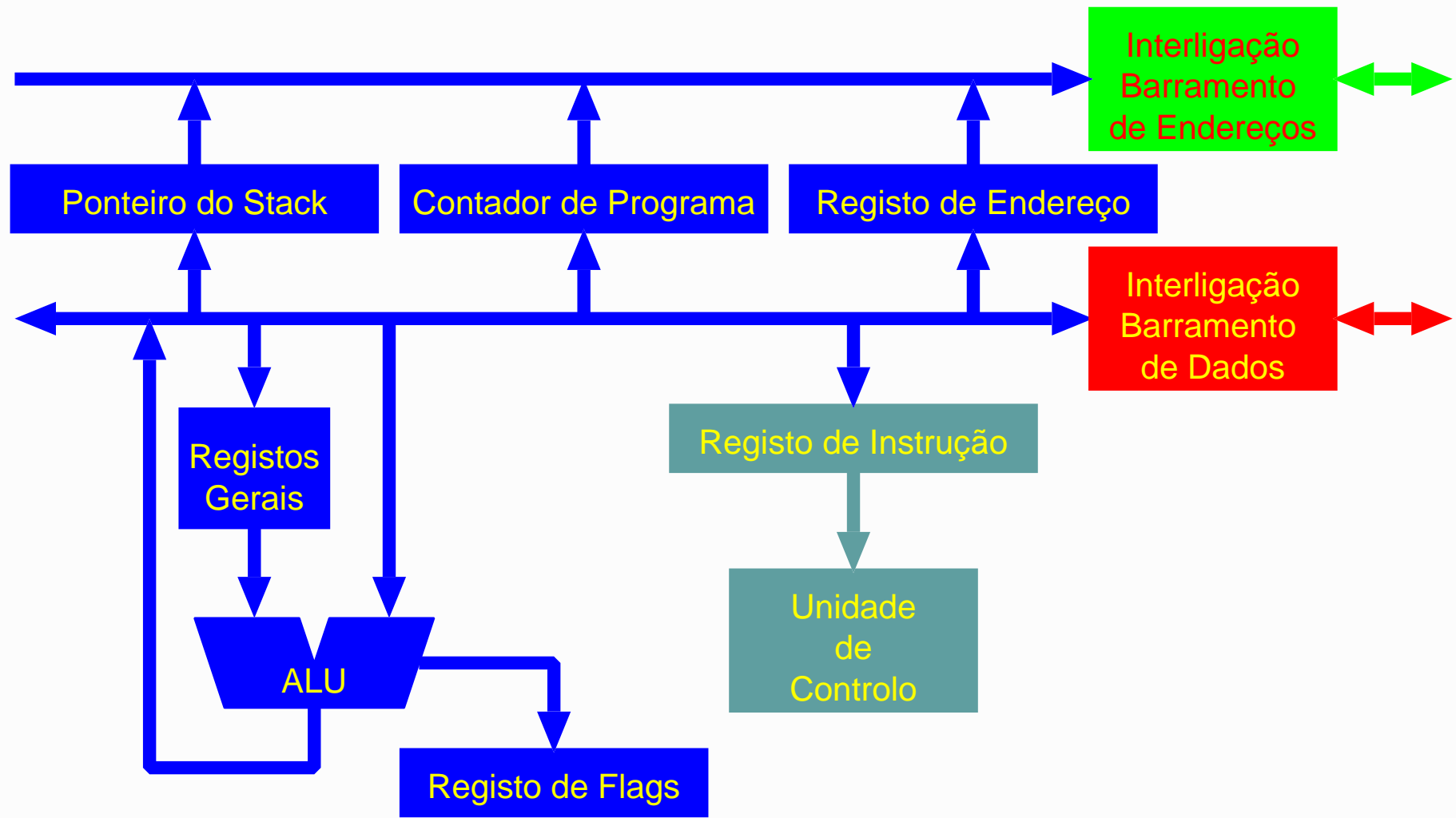
## Arquitecturas de Microprocessadores

### Modelo de Von Neumann



# Arquitecturas de Microprocessadores

## Arquitetura Geral



## Arquitecturas de Microprocessadores

### Arquitetura Geral

	<b>Linguagem de Alto Nível</b>	<b>Microprocessador</b>
<b>Organização da Memória</b>	Variáveis com Nome	“Array” Linear
<b>Estrutura de Dados</b>	Multidimensionais	Unidimensionais
<b>Programação</b>	- Rotinas e Retornos - Parâmetros - Memória Dinâmica	Operações de “Stack”
<b>Tipo de Memória</b>	Dados $\neq$ Instrução	Dados = Instrução

Normalmente feitas para superar o **Semantic Gp**

## Arquitecturas de Microprocessadores

### Arquitetura Geral

#### Arquitetura e Eficiência de Execução Tipos de Dados

- +/– idênticos no Microprocessador e na Linguagem de Alto Nível

#### Modos de Endereçamento

- Definem mecanismos para aceder aos dados
- Podem proporcionar mecanismos eficientes para mapear estruturas de dados complexas

#### Conjunto de Instruções

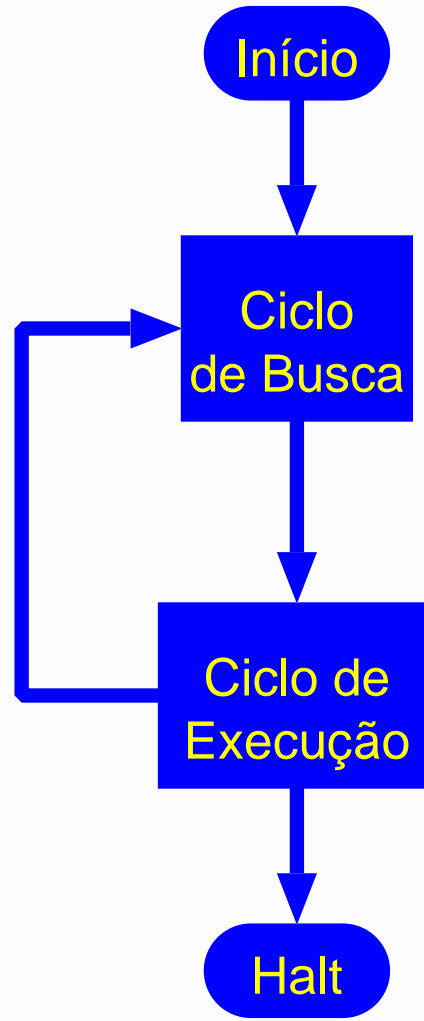
Podem aumentar eficiência de duas formas:

- CISC - Complex Instruction Set Computer
- RISC - Reduced Instruction Set Computer



## Ciclos de Instrução

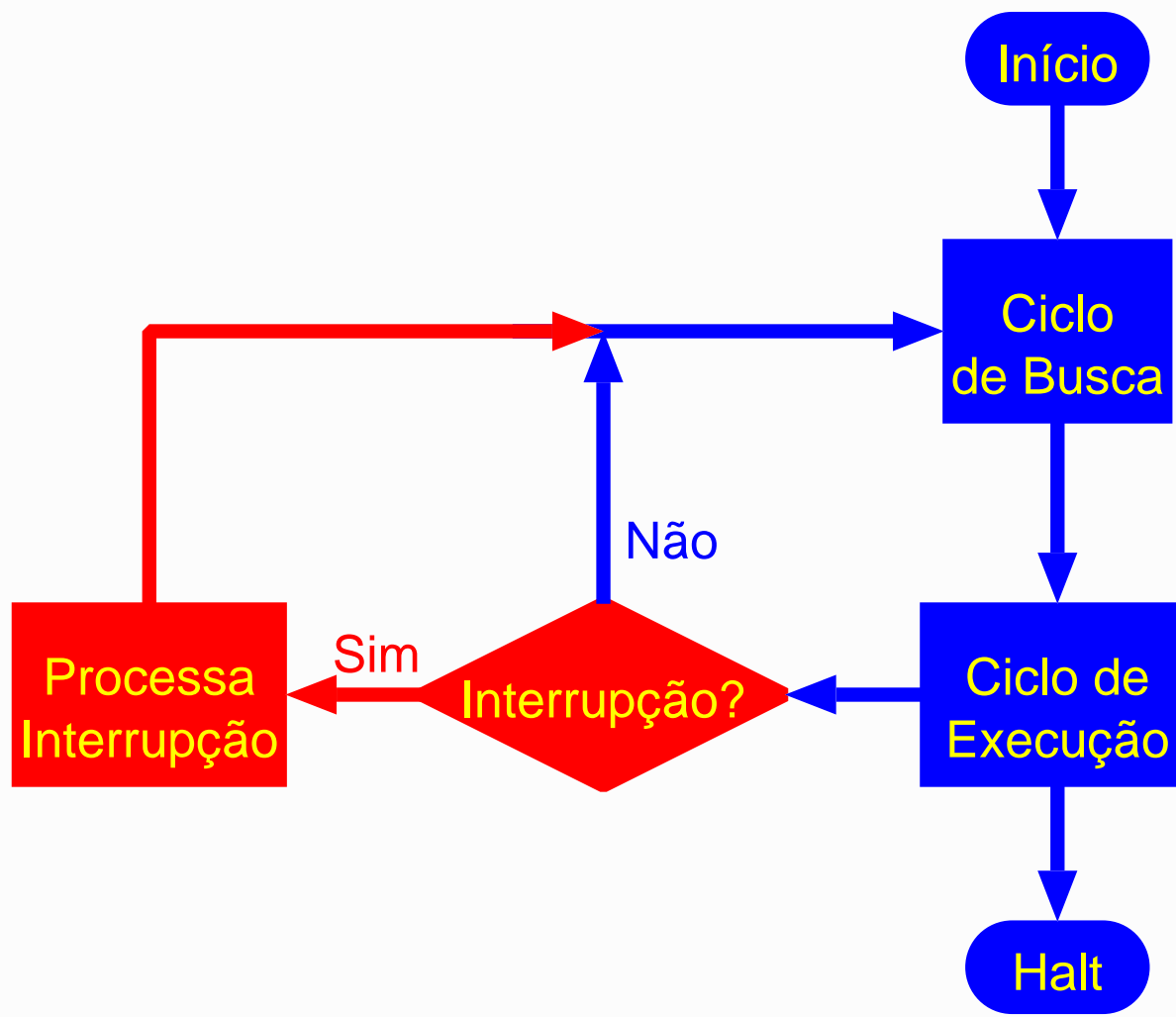
### Ciclo Básico de Instrução





## Ciclos de Instrução

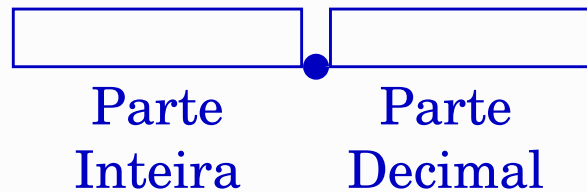
### Ciclo Básico de Instrução com Interrupções



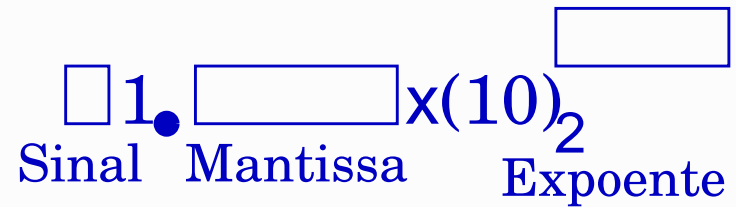
## FPU - “Floating Point Unit”

### Representação de Números Reais

#### Ponto Fixo



#### Ponto Flutuante



#### Underflow -

Número com valor absoluto muito pequeno diferente de zero que não pode ser representado.

#### Overflow -

Número com valor absoluto muito grande que não pode ser representado.

#### FPU - “Floating Point Unit”

Circuito que se destina a fazer operações com números em vírgula flutuante.



## FPU - “Floating Point Unit”

### Norma do IEEE 754-1985

Nome	Número de bits	Sinal	Mantissa	Expoente	Polarização
Precisão Simples	32	1	23	8	127
Precisão Dupla	64	1	52	11	1023

Valor representado em decimal:

$$(-1)^{\text{Sinal}} \times \left( 1 + \sum_{n=1}^{p-1} \text{Mantissa}(n) \times 2^{-n} \right) \times 2^{\text{Expoente-Polarização}}$$

Formato:

Sinal	<b>Expoente</b>	<b>Mantissa</b>
-------	-----------------	-----------------

Casos especiais:

Tipo	Expoente	Mantissa
Zero	00...000	00...000
Infinito	11...111	00...000
NaN	11...111	≠ 0
Num. não normalizado	00...000	≠ 0

