

Introdução aos Microprocessadores

António M. G. Pinheiro

Universidade da Beira Interior

Covilhã - Portugal

`pinheiro@ubi.pt`

Evolução histórica

EVOLUÇÃO DE COMPUTADORES

1ª Geração	<i>Década de 40</i>	Baseados em Válvulas
2ª Geração	<i>Década de 50</i>	Baseados em Transistores
3ª Geração	<i>Década de 60</i>	Baseados em C.I.s que implementavam <i>lógica discreta</i>
4ª Geração	<i>Década de 70</i>	Baseados em C.I.s <i>μProcessadores</i>



Evolução histórica

EVOLUÇÃO DE μ PROCESSADORES

Palavra	Performance	Exemplos
4 bit <i>nibble</i>	baixa	INTEL 4004 (\leftarrow 1971)
8 bit <i>byte</i>	média	INTEL 8008; INTEL 8080; INTEL 8085; INTEL 8048; INTEL 8051; ZILOG Z80; MOTOROLA 6800
16 bit	alta	INTEL 8088; INTEL 8086; INTEL 80286; ZILOG Z8000; MOTOROLA 68000
32 bit	muito alta	INTEL 80386; INTEL 80486; INTEL Pentium; MOTOROLA 68020; MOTOROLA 68040
64 bit	muito alta	POWER PC; DIGITAL ALPHA; AMD ATHLON64



Evolução histórica

EVOLUÇÃO DE μ PROCESSADORES

- μ Processadores

↪ 1971 → INTEL 4004 (4 bits)

↪ 1972 → INTEL 8008 (8 bits)
National Semiconductor IMP-16



Evolução histórica

EVOLUÇÃO DE μ PROCESSADORES

↔ 1973 ... 1977 → C.I. de lógica discreta
Tipicamente:

- △ 8 bits de dados
- △ 16 bits de endereço (64 kbytes)
- △ Conjunto de instruções complexo
- △ “Stack” na memória
- △ Assembly
- △ Relógio de 1 MHz
- △ Coprocessadores aritméticos escravos (ponto flutuante)



Evolução histórica

EVOLUÇÃO DE μ PROCESSADORES

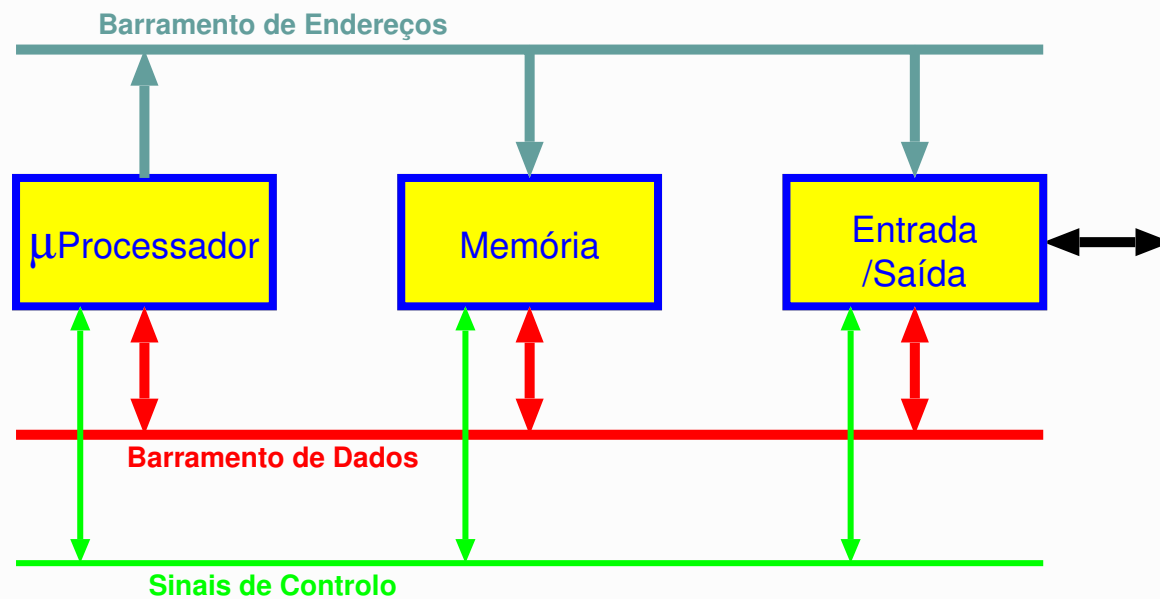
↔ 1977 ... → μ Processadores incluem números de vírgula flutuante
(alguns incluem coprocessador aritmético)

Permite:

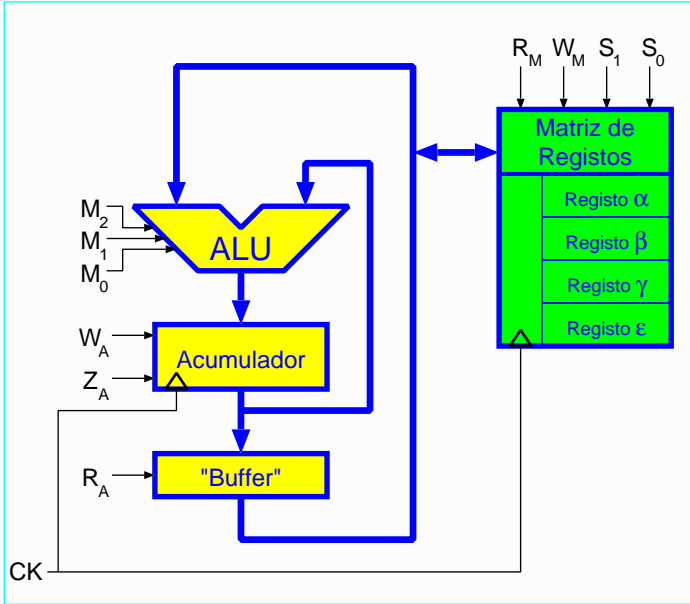
- △ Computadores pessoais (PC)
- △ Desenho Assistido por Computador (CAD)
- △ Sistemas com partilha temporal
- △ Linguagens de Alto Nível (Pascal, Fortran, C, Basic, etc)



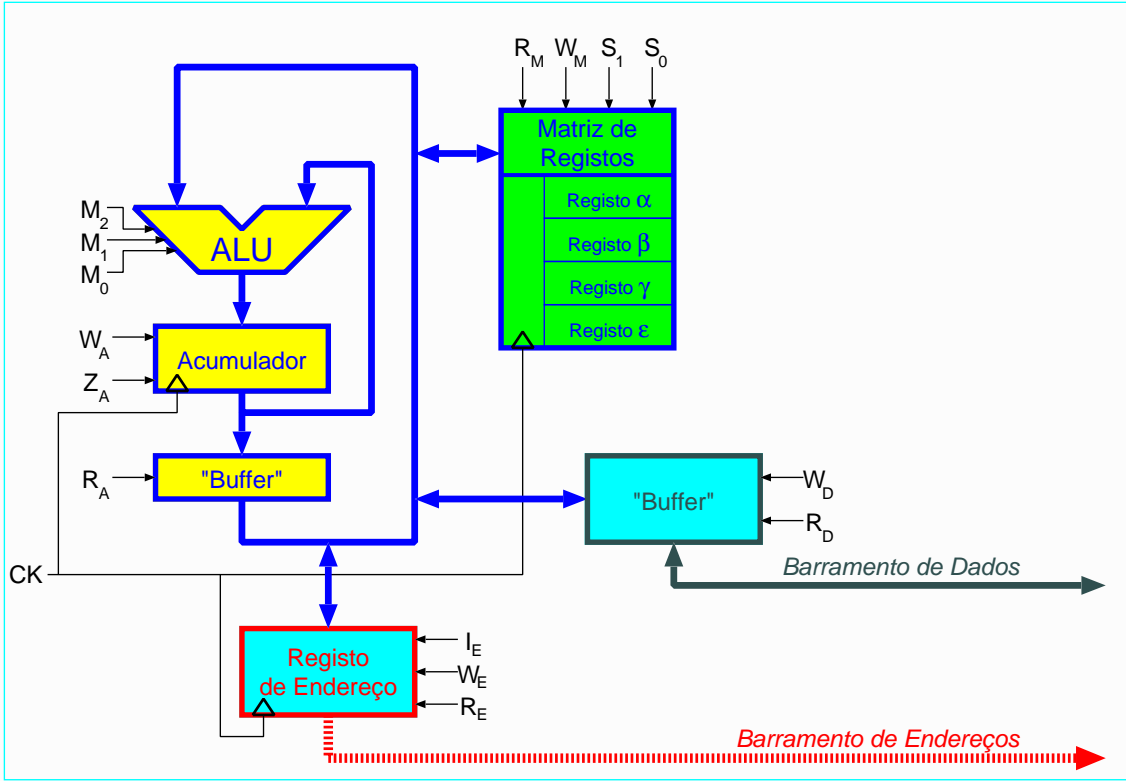
Modelo de von Neumann



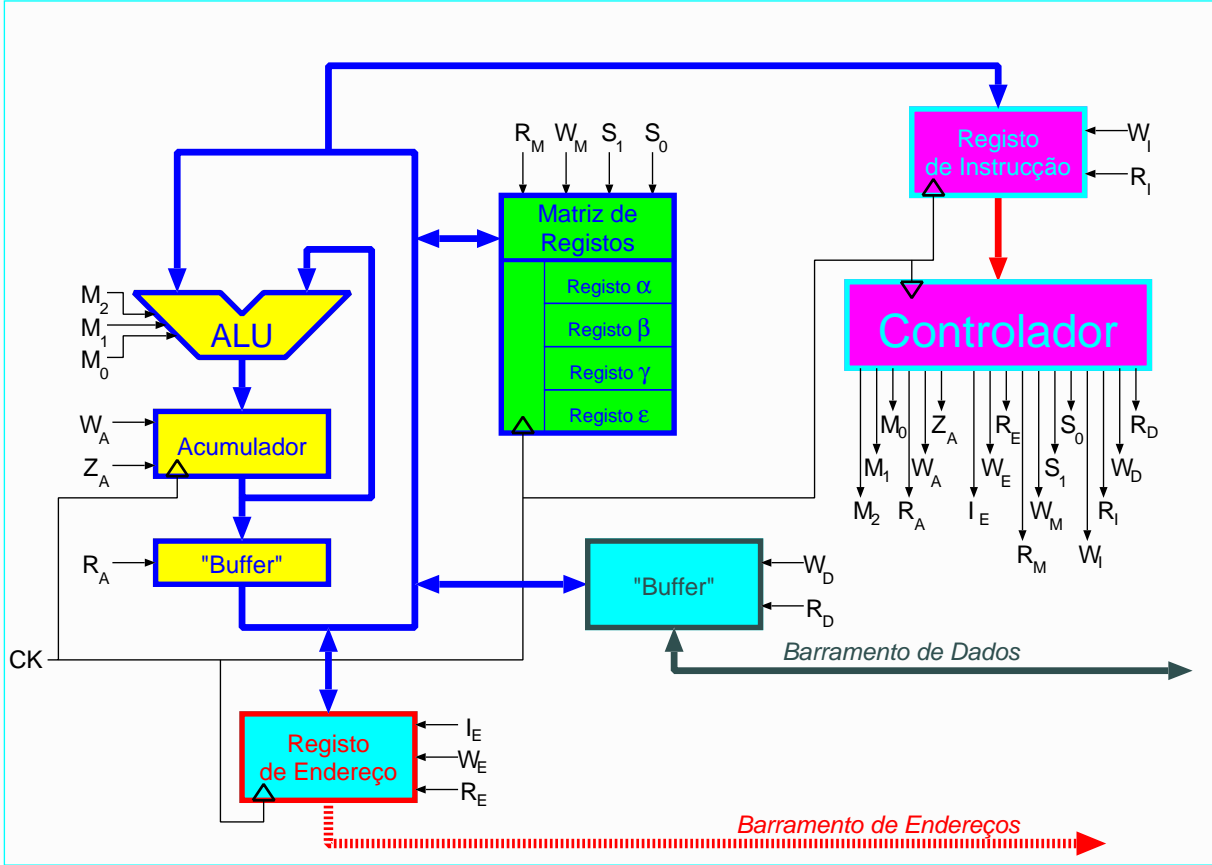
Estrutura de μ Processadores



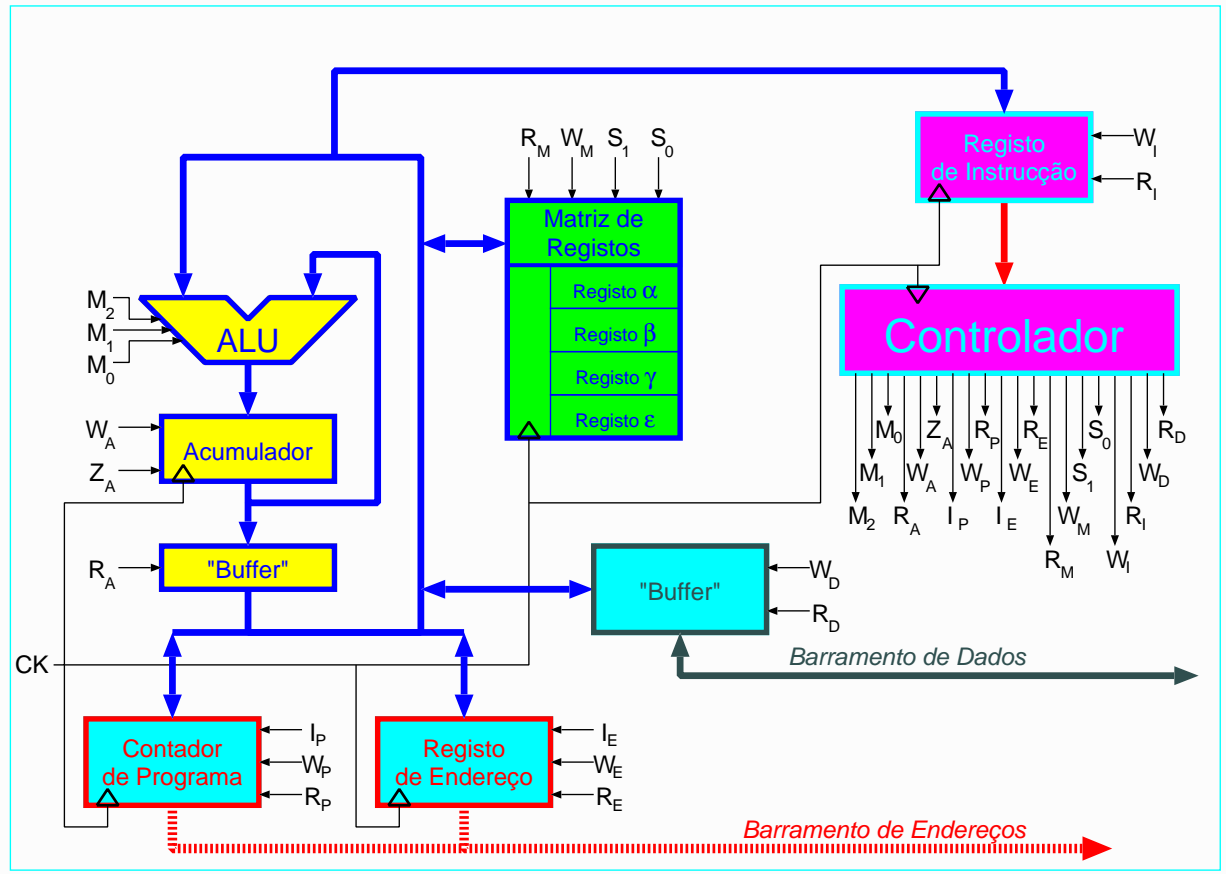
Estrutura de μ Processadores



Estrutura de μ Processadores



Estrutura de μ 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 AND B
1	0	1	A OR B
1	1	0	A EXOR B
1	1	1	NOT(A)



Instrucções

TRÊS GRUPOS:

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

EXEMPLO: INSTRUCÇÕES 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	s_1	s_0	0	0	ST R_s	ARMAZENA R_s NO ENDEREÇO
1	0	0	1	s_1	s_0	0	1	MOV EL, R_s	MOVE R_s PARA A PARTE BAIXA DO REGISTO DE ENDEREÇO
1	0	0	1	s_1	s_0	1	0	MOV EH, R_s	MOVE R_s PARA A PARTE ALTA DO REGISTO DE ENDEREÇO
1	0	0	1	s_1	s_0	1	1	MOV PC, EH	MOVE REGISTO DE ENDEREÇO PARA O CONTADOR DE PROGRAMA



Instruções

OPERAÇÕES ARITMÉTICAS E LÓGICAS

I_7	I_6	I_5	I_4	I_3	I_2	I_1	I_0	MNEMÓNICA	OPERAÇÃO
0	0	0	0	0	0	0	0	NOP	“NO OPERATION”
0	0	1	0	d_1	d_0	s_1	s_0	ADD R_d, R_s	$R_d \rightarrow R_d + R_s$
0	0	1	1	d_1	d_0	s_1	s_0	SUB R_d, R_s	$R_d \rightarrow R_d - R_s$
0	0	0	0	d_1	d_0	1	1	INC R_d	$R_d \rightarrow R_d + 1$
0	0	0	1	d_1	d_0	1	1	DEC R_d	$R_d \rightarrow R_d - 1$
0	1	1	0	d_1	d_0	s_1	s_0	AND R_d, R_s	$R_d \rightarrow R_d \text{ AND } R_s$
0	1	1	1	d_1	d_0	s_1	s_0	OR R_d, R_s	$R_d \rightarrow R_d \text{ OR } R_s$
0	1	0	0	d_1	d_0	s_1	s_0	EXOR R_d, R_s	$R_d \rightarrow R_d \text{ EXOR } R_s$
0	1	0	1	d_1	d_0	1	1	NOT R_d	$R_d \rightarrow \text{NOT}(R_d)$



Instruções

OPERAÇÕES DE CONTROLO DE PROGRAMA

I_7	I_6	I_5	I_4	I_3	I_2	I_1	I_0	MNEMÓNICA	OPERAÇÃO
1	1	0	0	0	0	0	0	JMP	SALTO INCONDICIONAL
1	1	0	0	0	C_2	C_1	C_0	JCC	SALTO CONDICIONAL
1	1	0	0	1	0	0	0	CALL	ROTINA INCONDICIONAL
1	1	0	0	1	C_2	C_1	C_0	CCC	ROTINA CONDICIONAL
1	1	0	1	0	0	0	0	RET	RETORNO DE ROTINA

