

Universidade de São Paulo

Organização de Computadores

Dr. Jorge Luiz e Silva

Cap 1

Arquitetura de Computadores

.Organização Básica de Computadores

- Histórico da Evolução dos Computadores
- Organização Básica do Computador
- Unidade Lógica e Aritmética
- Unidade de Memória
- Microprocessador 8088/8086

Bibliografia:

- 1- TANEMBAUM, A.S. - "*Organização Estruturada de Computadores*". Prentice Hall, 1990.
- 2- DANDAMUDI, S.P. - "*Introduction to Assembly Language Programming: From 8086 to Pentium Processors (Undergraduate Text in Computer Science)*". Springer Verlag, 1998.
- 3 - NORTON, P - "*Linguagem Assembler para IBM/PC*" - Ed. Campos, 1990.
- 4 - HOLZNER, S - "*Linguagem Assembler Avançado para IBM/PC*" - McGraw-Hill, 1991.

Avaliação

Provas - P1	(70%)
Exercícios	(30%)

Arquiteturas Sequencias

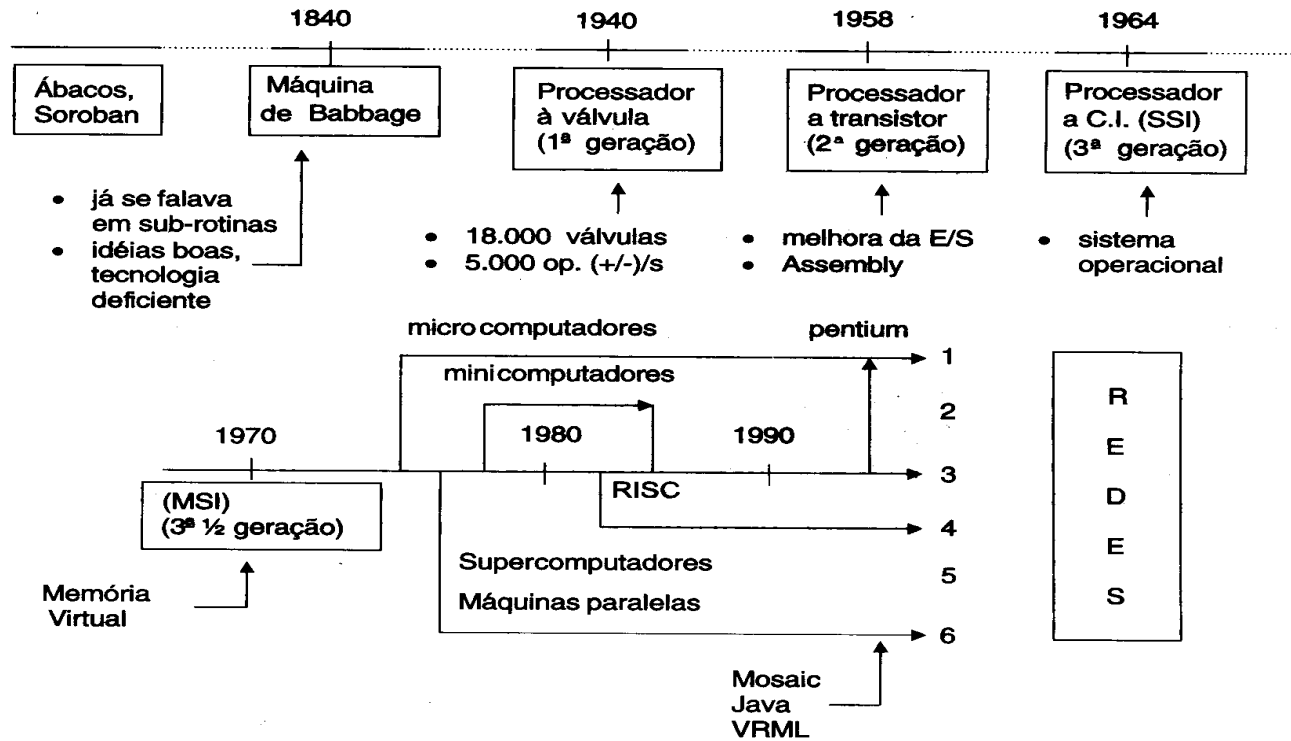
Organização Básica de um Computador



- Processadores (histórico, e evolução)
- Memória (estrutura básica)
- Entrada e Saída (dispositivos básicos)

Linguagem Assembler

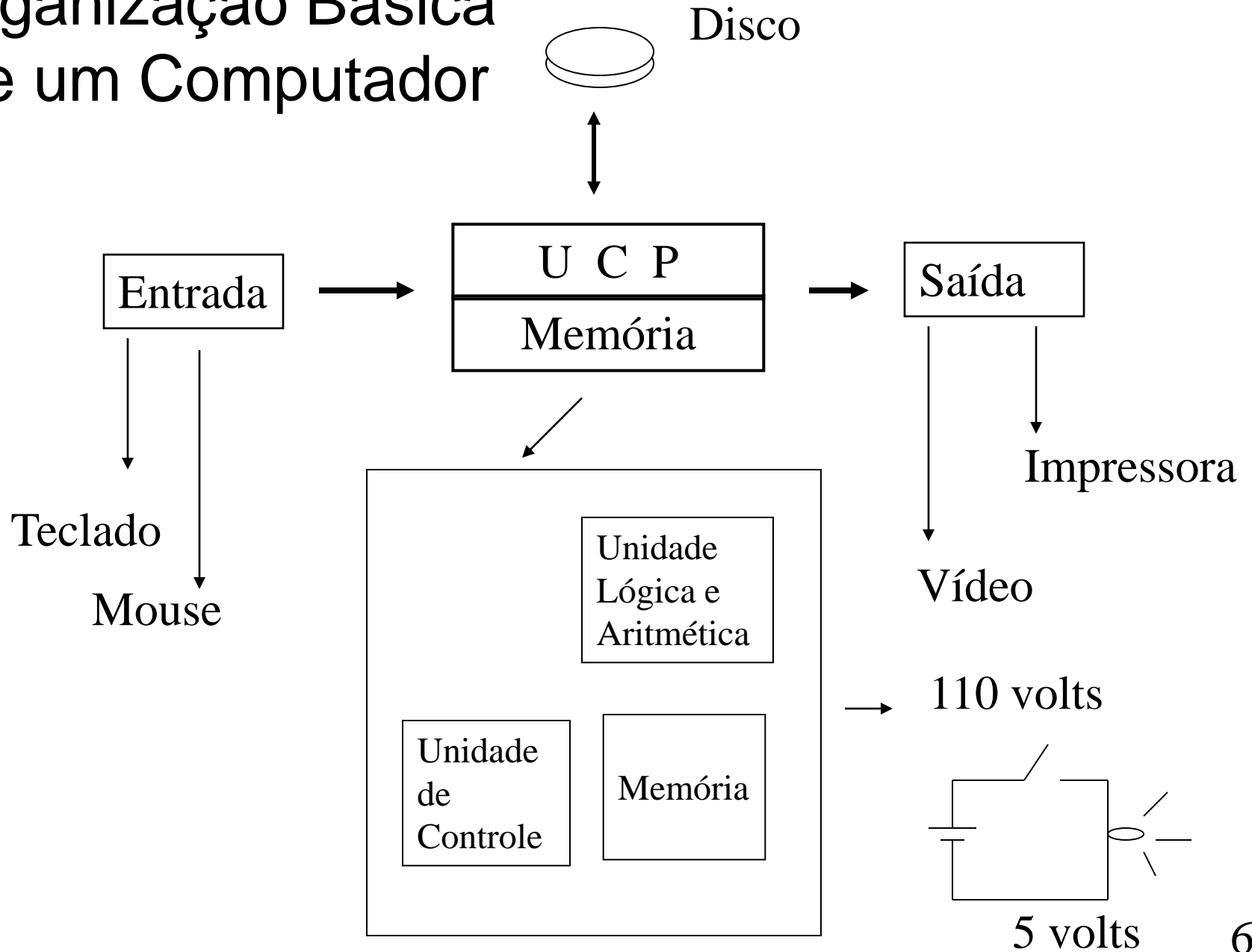
- Representação de instruções
- Modos de Endereçamento
- Arquiteturas do MIPS e 80×86
- Exemplo de programa



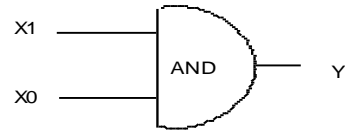
1. Computadores pessoais/ET (Estação de Trabalho) simples/"Transputers"
 2. Computadores de porte médio (CISC)
 3. Computadores de grande porte — Mainframes
 4. ET's RISC: SUN, APOLLO...
 5. Supercomputadores: CRAY, NEC...
 6. Multiprocessadores
- CISC — "Complex Instruction Set Computer"
RISC — "Reduced Instruction Set Computer"
ET — Estação de Trabalho

Figura 1.8 Esquema do desenvolvimento dos computadores.

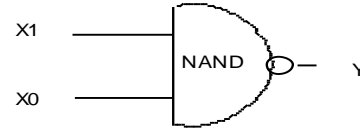
Organização Básica de um Computador



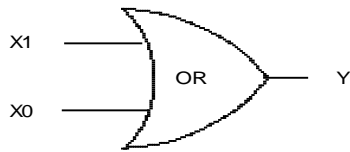
Dispositivos Lógicos



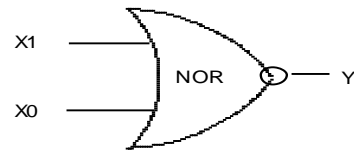
X1	X0	Y
0	0	0
0	1	0
1	0	0
1	1	1



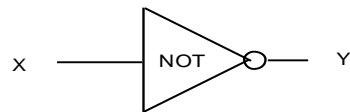
X1	X0	Y
0	0	1
0	1	1
1	0	1
1	1	0



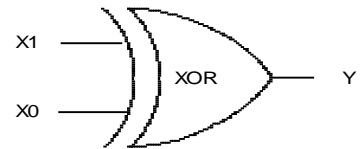
X1	X0	Y
0	0	0
0	1	1
1	0	1
1	1	1



X1	X0	Y
0	0	1
0	1	0
1	0	0
1	1	0



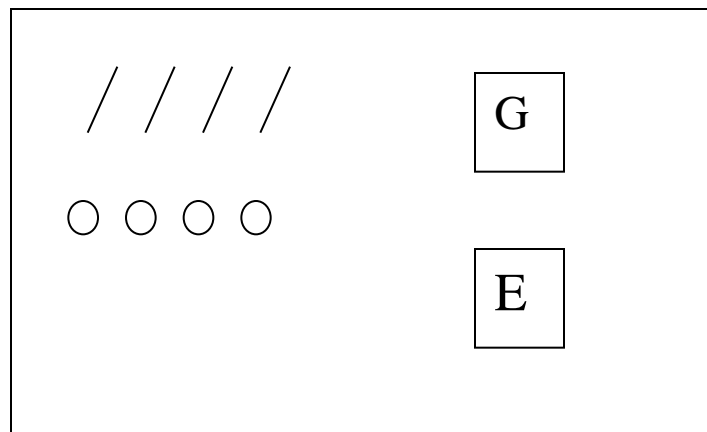
X	Y
0	1
1	0



X1	X0	Y
0	0	0
0	1	1
1	0	1
1	1	0

FIGA-A.WPG

0000	-	0
0001	-	1
0010	-	2
0011	-	3
0100	-	4
0101	-	5
0110	-	6
0111	-	7
1000	-	8
1001	-	9
1010	-	+
1011	-	-
1100	-	*
1101	-	/
1110	-	=
1111	-	?



Se adotarmos o primeiro bit de sinal

0 – positivo

1 – negativo

Total de 8 números positivos e 8 números negativos (cpl/2)

Representação de Números

$$1024 = 1 \cdot 10^3 + 0 \cdot 10^2 + 2 \cdot 10 + 4$$

$$N = A_n \cdot B^{n-1} + A_{n-1} \cdot B^{n-2} + \dots + A_1 \cdot B + A_0$$

$$10101010 \quad n = 8$$

$$N = 1 \cdot 2^7 + 1 \cdot 2^5 + 1 \cdot 2^3 + 1 \cdot 2$$

128 32 8 2 (Notação Posicional)

$$10000000 - 128$$

$$10001000 - 136$$

$$01111111 - 127$$

Operações com Binário

$$\begin{array}{r} 1000 \text{ (8)+} \\ 0100 \text{ (4)} \\ \hline 1100 \text{ (12)?} \end{array}$$

$$\begin{array}{r} 1 \leftarrow 0100 \text{ (4) -} \\ \underline{1000 \text{ (8)}} \\ 1100 \text{ (-4)} \end{array}$$

Representação para números negativos

COMPLEMENTO PARA UM

Trocar os 0's por 1's e vice-versa, do número que se quer complementar.

Ex:

0000 - 0	-0 - 1111
0001 - 1	-1 - 1110
0010 - 2	-2 - 1101
0011 - 3	-3 - 1100
0100 - 4	-4 - 1011
0101 - 5	-5 - 1010
0110 - 6	-6 - 1001
0111 - 7	-7 - 1000

Complemento para um

Vantagens e Desvantagens

Vantagens – Conversão

Desvantagens

- conflitos na representação do 0
- Perde uma representação
- Correção no resultado toda vez que uma operação + ou – resulte em um número < 0

Ex:

	1←	1011 (-4)	
		<u>1110 (-1)</u>	
		1001	
		<u>1</u>	bit de correção
		1010 (-5)	

Complemento para dois

COMPLEMENTO PARA UM + UM

Ex:

0000 - 0	-1 - 1111
0001 - 1	-2 - 1110
0010 - 2	-3 - 1101
0011 - 3	-4 - 1100
0100 - 4	-5 - 1011
0101 - 5	-6 - 1010
0110 - 6	-7 - 1001
0111 - 7	-8 - 1000

Complemento para Dois

Vantagens e Desvantagens

Vantagem

- Não existe conflito para zero
- Total de 16 representações
- Não é necessário correção de 0100 - resultado

Desvantagem

- Conversão

Capacidade de Representação

Em um sistema de Computador que trabalha com 8 bits:

$(2^7 - 1)$	(-2^7)	$(2^7 - 1)$	$- (-2^7 - 1)$
127	-128	127	-127

Complemento 2

Complemento 1

00000000 - 0

(-1) - 11111111

00000001 - 1

(-2) - 11111110

.

.

Cpl/2

.

.

01111111 - 127

(-128) - 10000000

Operações Binárias

00000001 - 1 +

00000011 - 3

00000100 - 4

1← 00000001 - (1) -

11111111 - (-1)

00000000

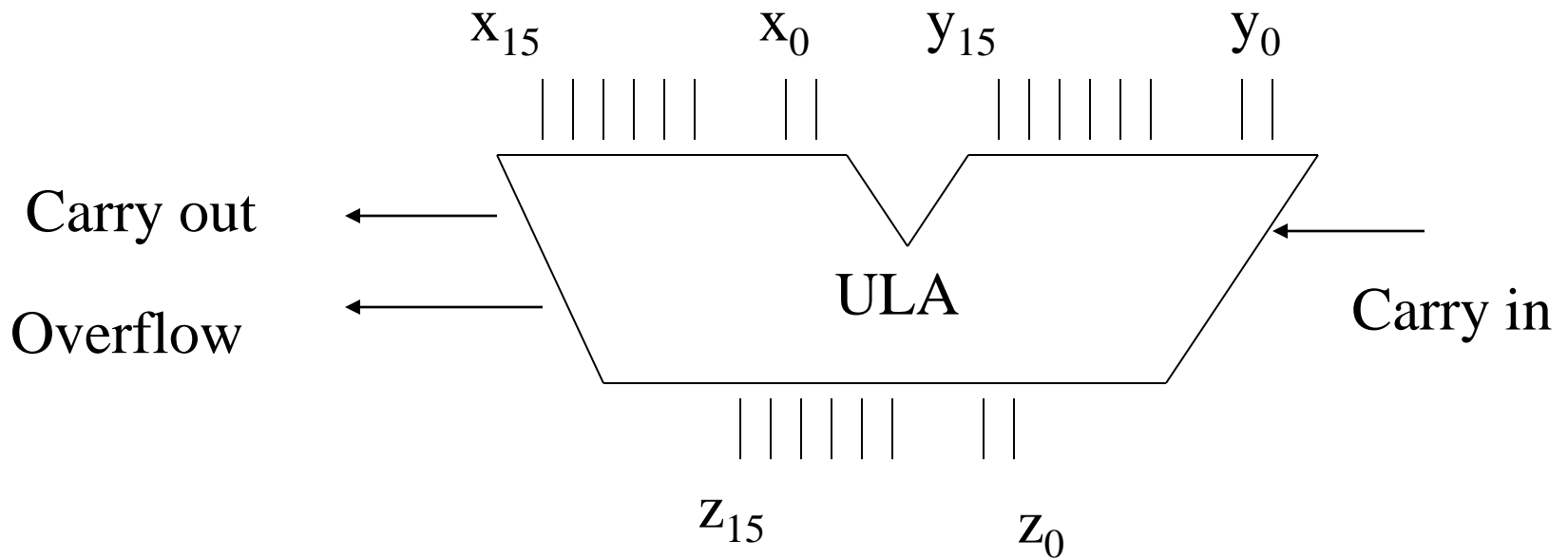
(1 - 1) → (1 +(- 1))

1← 11111111 - (-1) +

11111110 - (-2)

11111101 - (-3)

Unidade Lógica e Aritmética



Operações Aritméticas

C ← C' ←

```

0 000 0000 0000 0010 (2)    +
0 111 1111 1111 1111 (32767)
-----

```

1 000 0000 0000 0001 (32769) ? Estouro

Não existe representação para o número 32769

Overflow = C ⊕ C' 2 + 32767 overflow = 0 ⊕ 1 = 1

1 ← 1 ←

```

1 111 1111 1111 1111 (-1)

```

```

1 111 1111 1111 1111 (-2)
-----

```

```

1 111 1111 1111 1101 (-3)

```

overflow 1 ⊕ 1 = 0 não overflow

Operações Lógicas

x and y = z

1000 0000 0000 0000 (x)

1101 0000 0000 0001 (y)

1000 0000 0000 0000 (z)

x or y = z

1000 0000 0000 0000 (x)

1101 0000 0000 0001 (y)

1101 0000 0000 0001 (z)

not x

1000 0000 0000 0000 (x)

0111 1111 1111 1111 (x')

Comparações

$x = y ?$

subtração de x por y

análise de resultado e carry

Ex: 2-1

0
0000 0000 0000 0010 (2) –
0000 0000 0000 0001 (1)

0000 0000 0000 0001 (# 0)
carry = 0

1
0000 0000 0000 0001 (1) –
0000 0000 0000 0010 (2)

1111 1111 1111 1111 (-1) # 0
carry = 1

Comparar x com y

carry = 0	e	resultado # 0	$x > y$
carry = 0	e	resultado = 0	$x = y$
carry = 0	e	resultado # 0	$x < y$

Comparações - exemplos

Ex:

0

0000 0000 0000 0011 (3) –

0000 0000 0000 0010 (2)

0000 0000 0000 0001 (#0) 3 > 2

1

0000 0000 0000 0010 (2) –

0000 0000 0000 0011 (3)

1111 1111 1111 1111 (# 0) 2 < 3

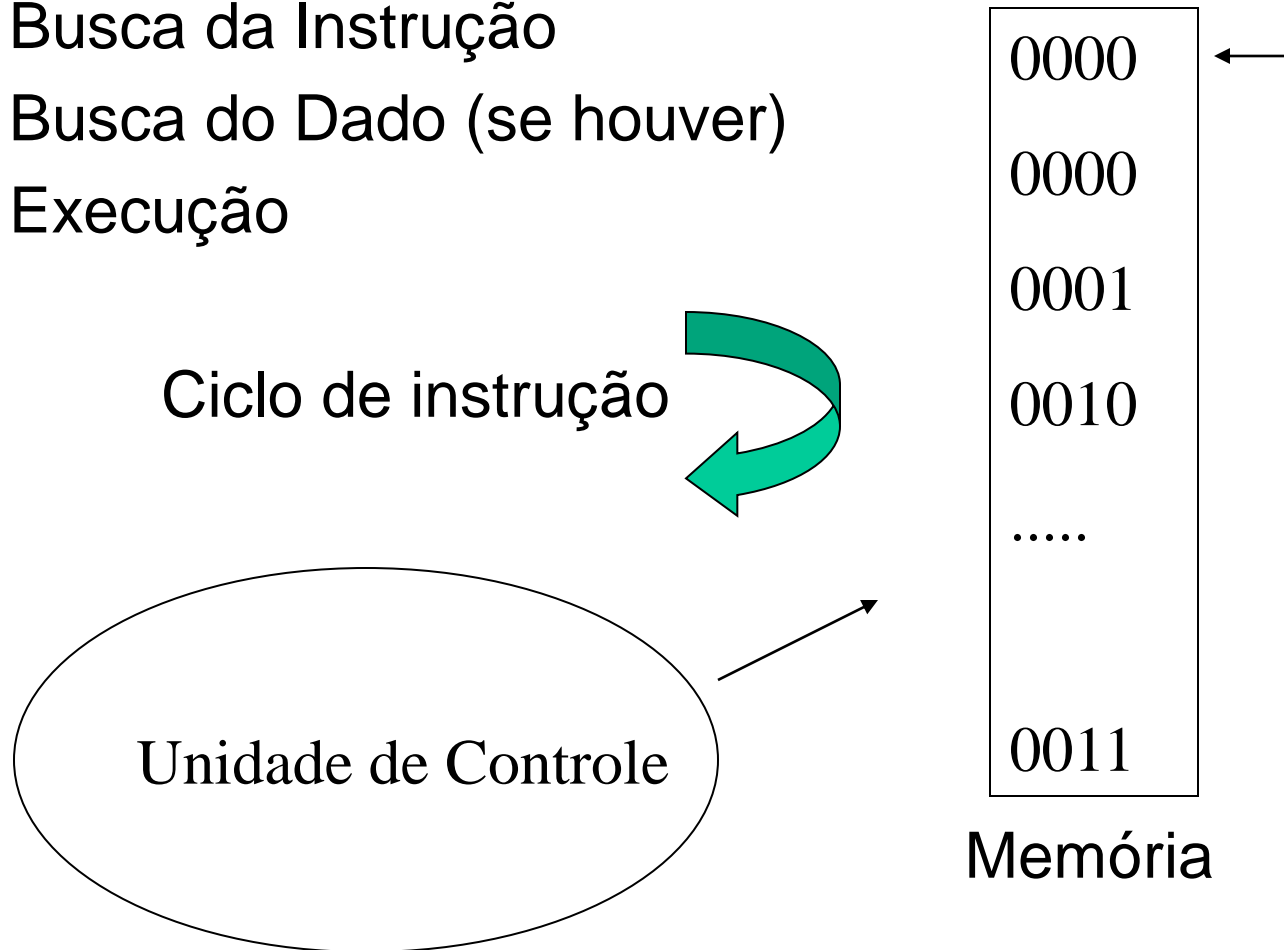
Tabelas de dados e instruções

Tabelas

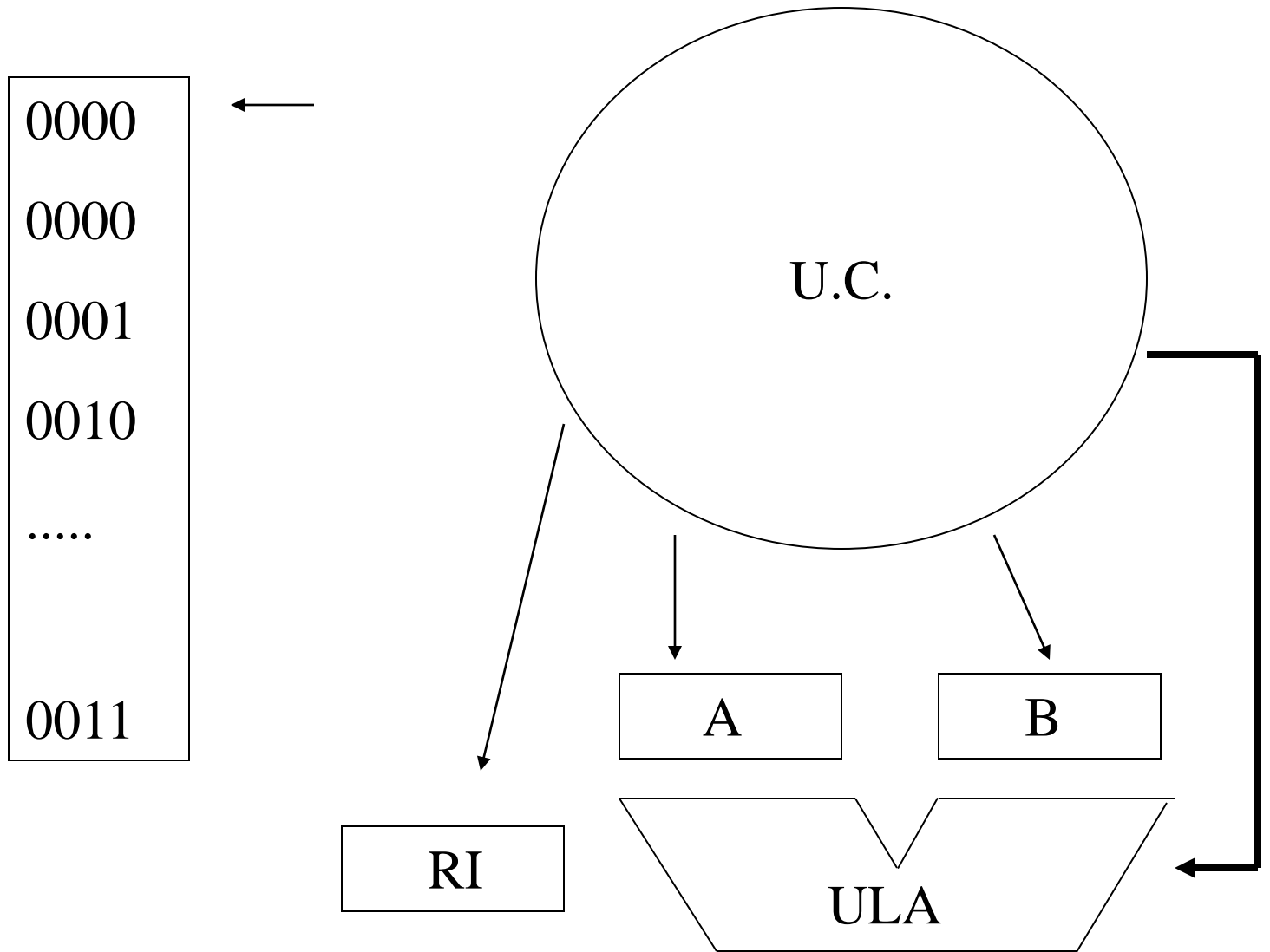
0000	-	0	0000	-	op1
0001	-	1	0001	-	op2
0010	-	2	0010	-	op3
0011	-	3	0011	-	op4
0100	-	4	0100	-	op5
0101	-	5	0101	-	op6
0110	-	6	0110	-	op7
0111	-	7	0111	-	op8
1000	-	(-8)	1000	-	op9
1001	-	(-7)	1001	-	op10
1010	-	(-6)	1010	-	op11
1011	-	(-5)	1011	-	op12
1100	-	(-4)	1100	-	op13
1101	-	(-3)	1101	-	op14
1110	-	(-2)	1110	-	op15
1111	-	(-1)	1111	-	op16

Ciclo de Instrução

- Busca da Instrução
- Busca do Dado (se houver)
- Execução

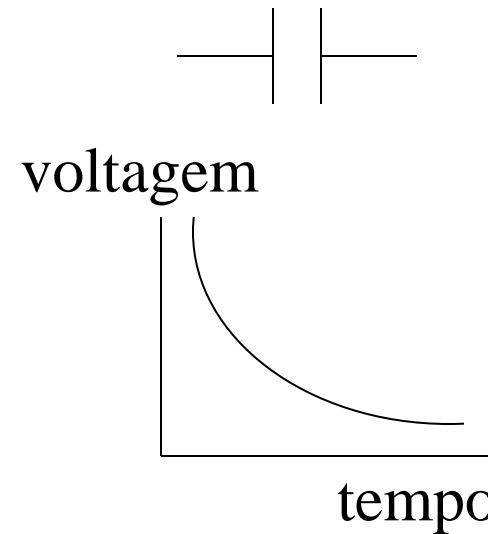
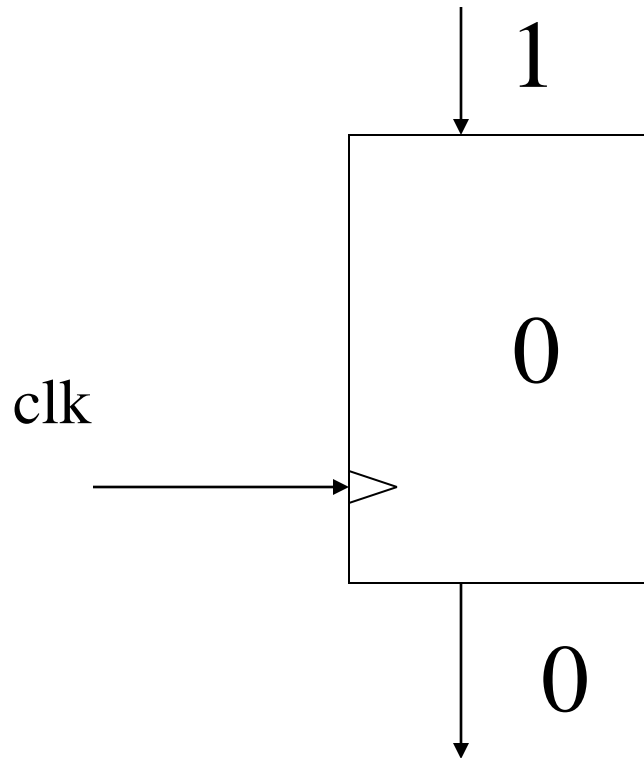


Uma organização simplificada

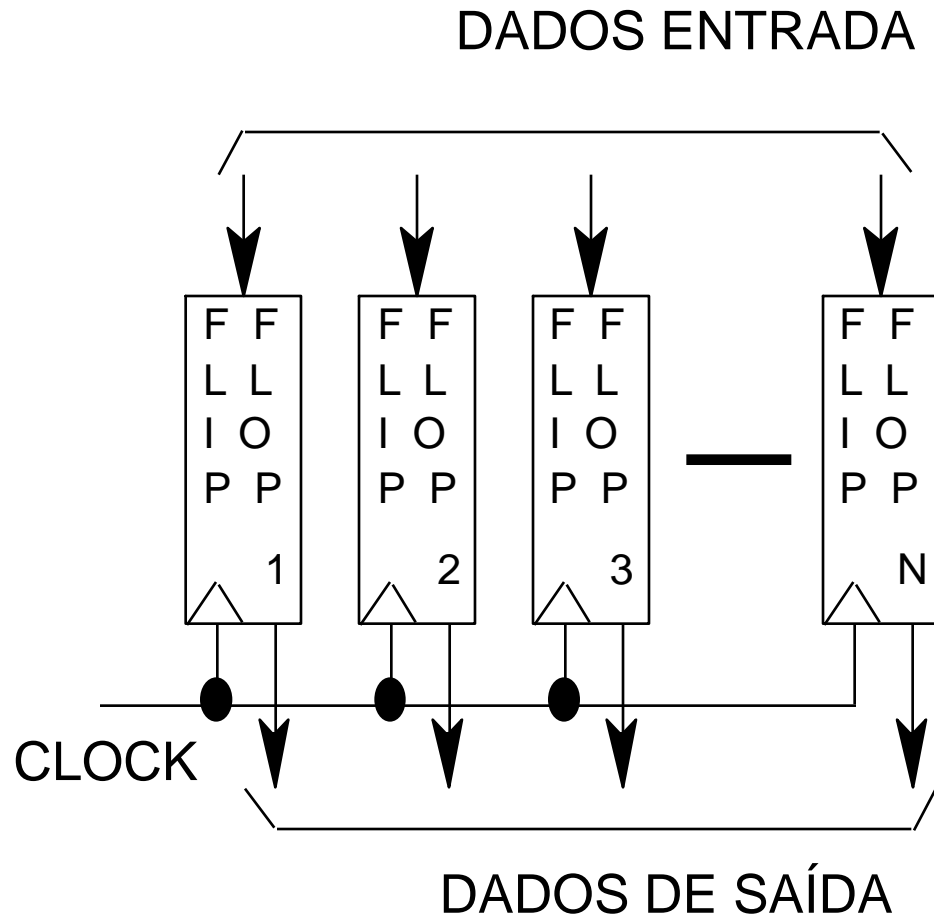


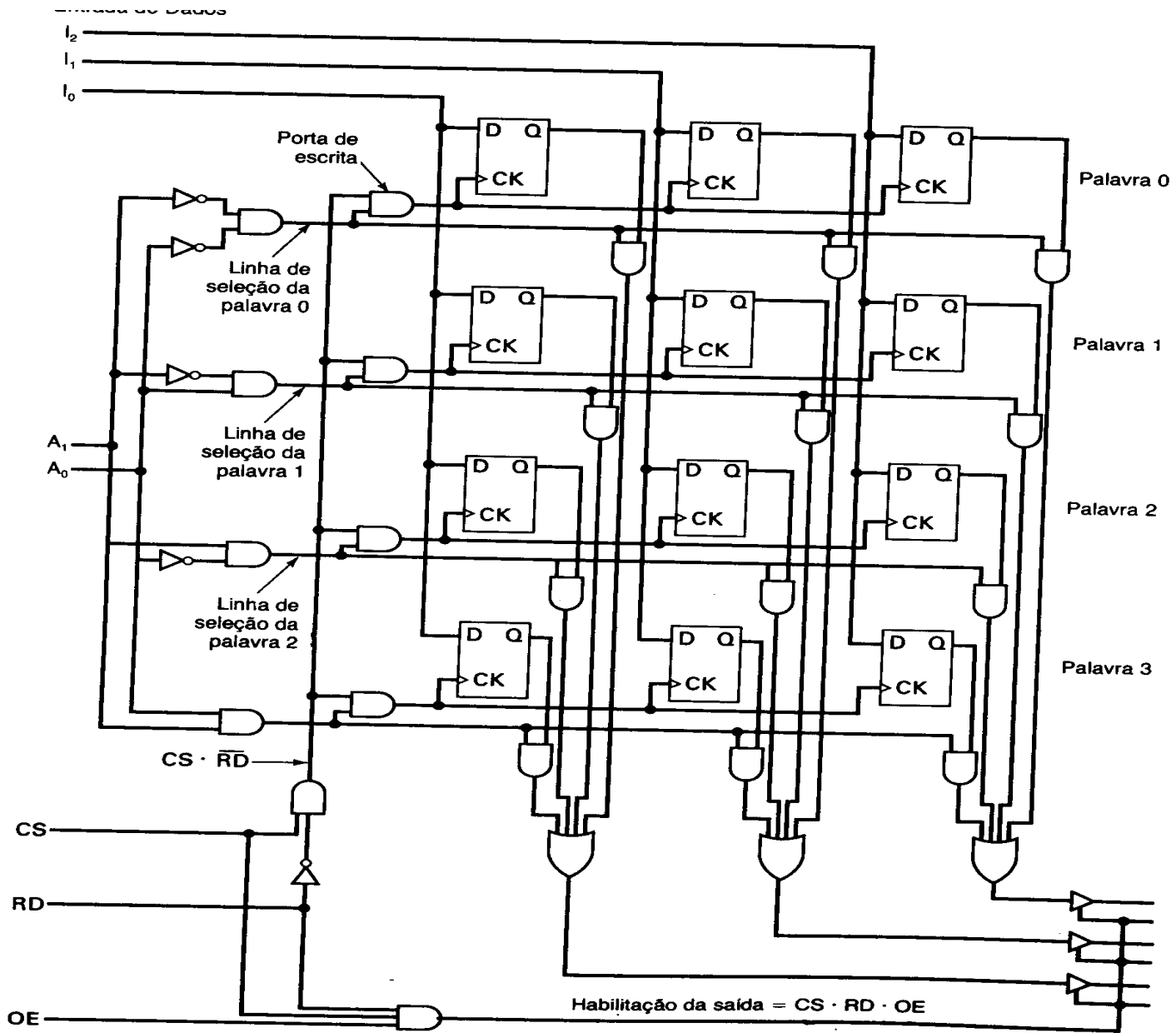
Unidade de Memória

Flip Flop

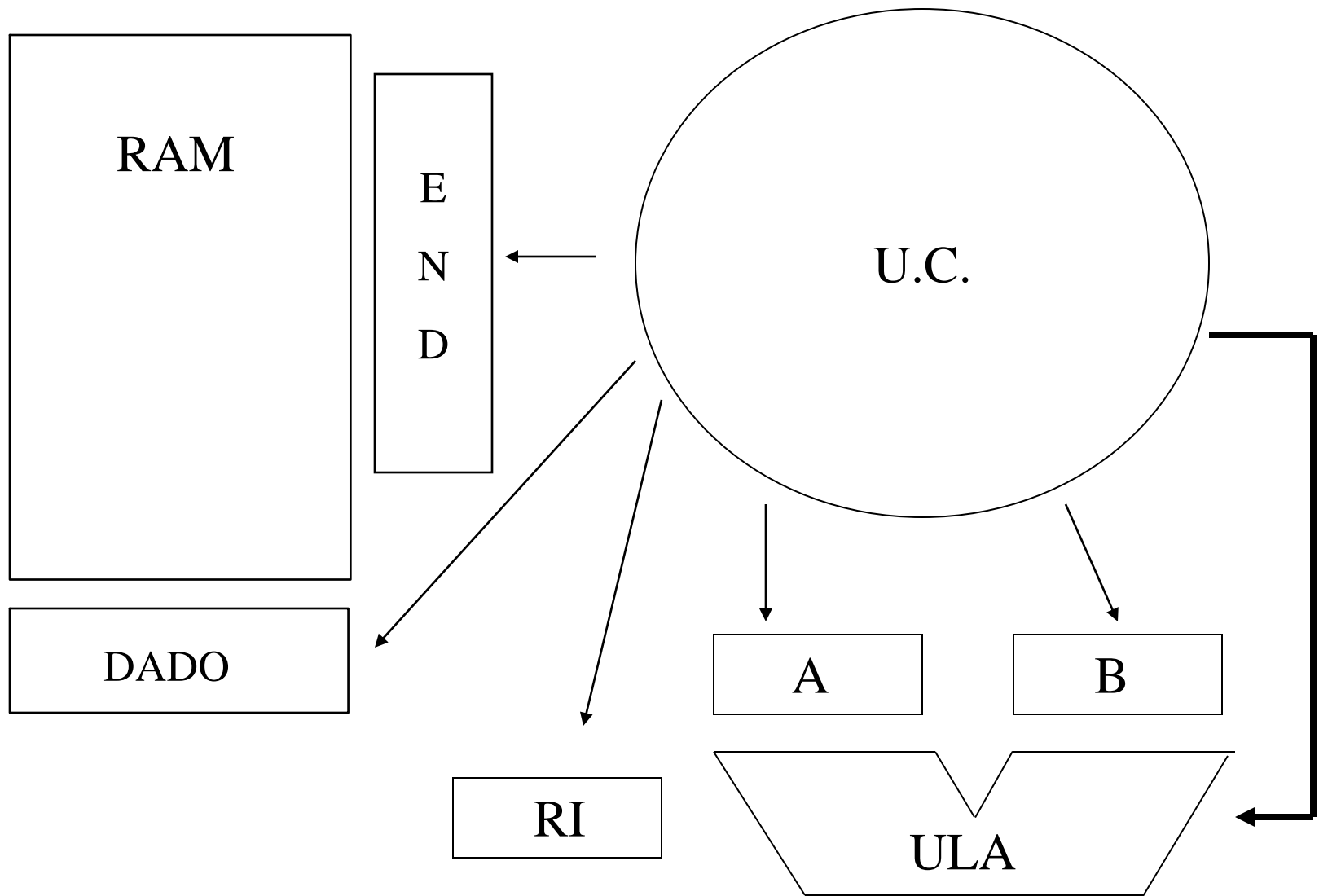


Registrador com Clock





Detalhes de uma organização



Capacidade de Memória

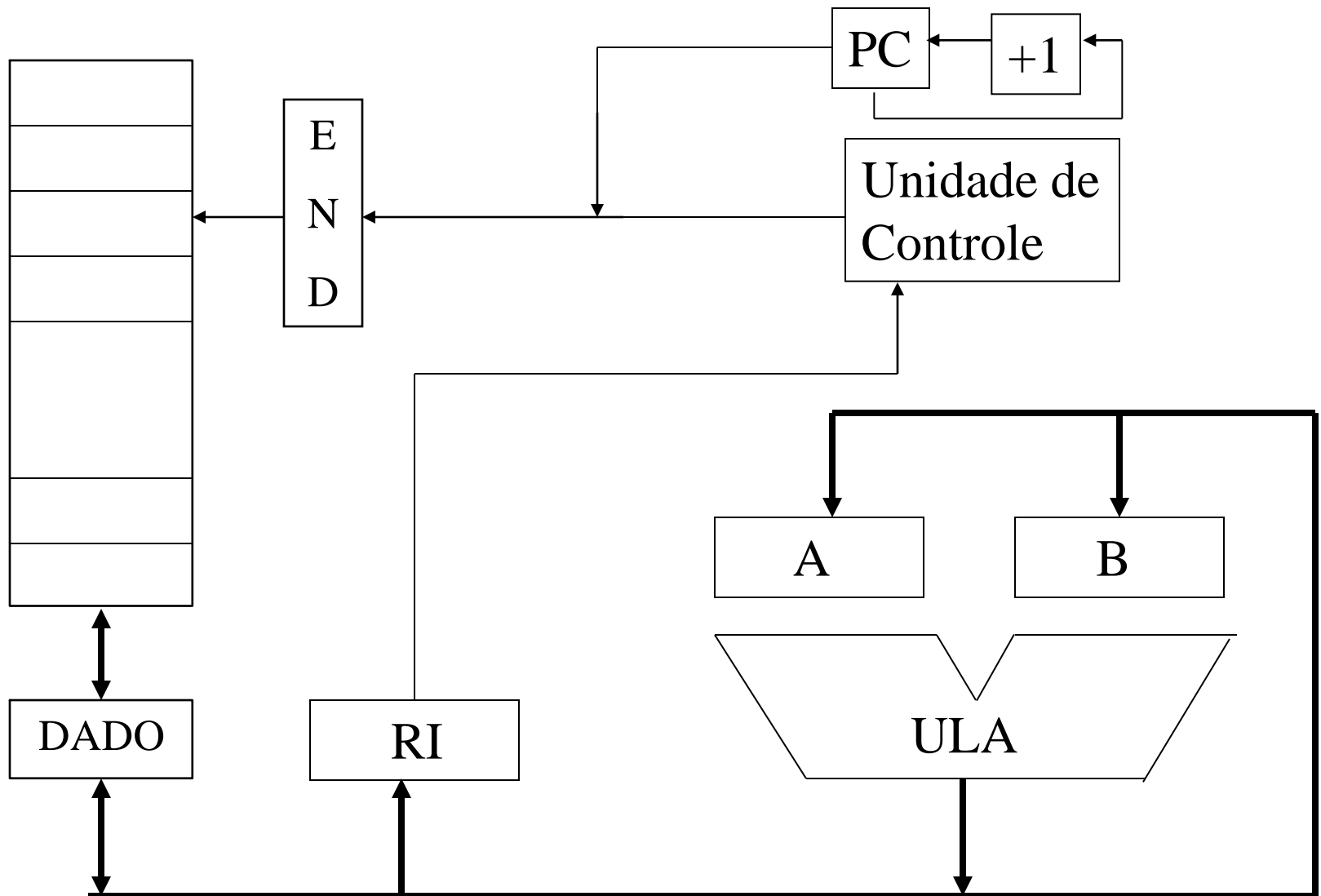
Registrador de Endereço:

- 4 bits (16 posições de memória)
- 8 bits (256 posições de memória)
- 16 bits (65536 posições de memória)
- 32 bits...
- 64 bits...

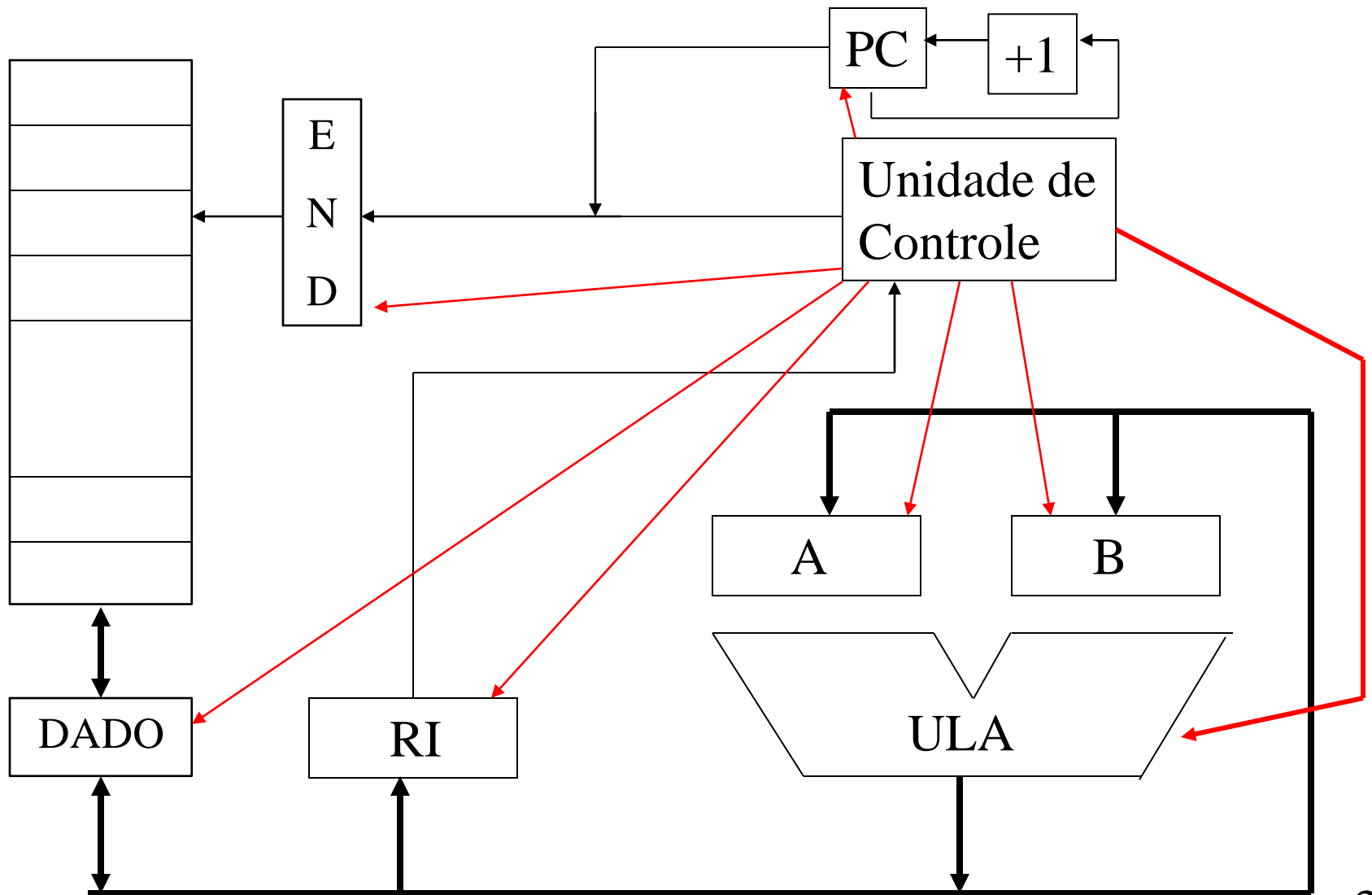
Registrador de Dados:

- 4 bits (tabela com até 16 operações)
- 8 bits (tabela com até 256 operações)
- 16, 32 bits...

Estrutura detalhada de um Sistema



Linhas de Controle em um Sistema



Exemplo de controle de um Sistema

11100001	LD	A,[X]	$A \leftarrow [X]$
11100010	LD	B,[Y]	$B \leftarrow [Y]$
11100011	ADD	A,B	$A \leftarrow A+B$
11100100	LD	[Z],A	$[Z] \leftarrow A$

