

Introdução à Ciência da Computação

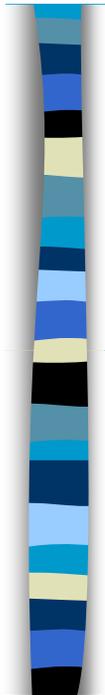
Contextualização Histórica e Noções de Computadores

Prof. Ricardo J. G. B. Campello

Créditos

- ◆ Exceto por algumas adaptações, este material é composto de slides gentilmente cedidos por:
 - Prof. André de Carvalho
 - ◆ Slides baseados nos originais das professoras Rosely Sanches, Sandra Aluísio, Solange Rezende e Renata Fortes
 - Prof. Rudinei Goularte
 - ◆ Slides baseados nos originais da Profa. Rosely Sanches

2



PARTE 1

Contextualização Histórica

3

Histórico do Computador

- O **computador** se desenvolveu paralelamente à necessidade crescente de **cálculos rápidos** e exatos da humanidade
- Os **ancestrais** do computador remontam a mais de **3000** anos

4

Primeiros Métodos de Cálculo

DEDOS

- É quase certo que o primeiro instrumento de cálculo que o homem utilizou foram seus próprios **dedos**

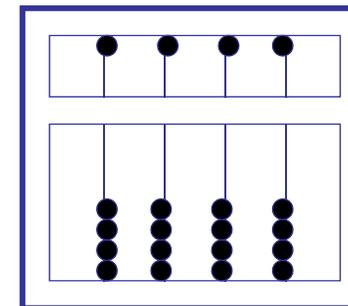


5

Primeiros Métodos de Cálculo

ÁBACO

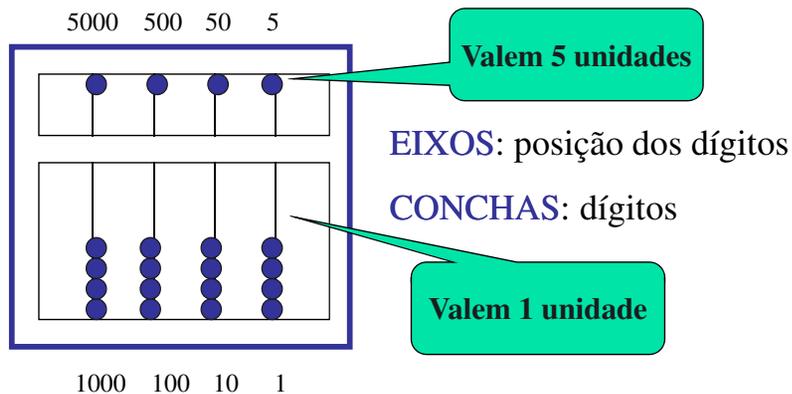
- Instrumento construído de **conchas** móveis se movimentando em **eixos**
- Aperfeiçoado pelos **chineses**



6

Primeiros Métodos de Cálculo

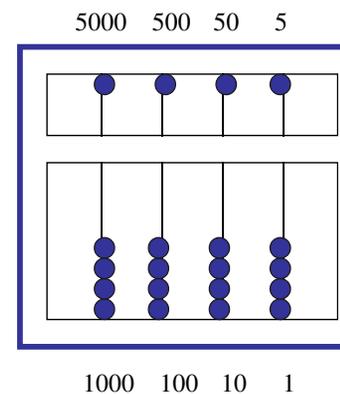
ÁBACO



7

Primeiros Métodos de Cálculo

ÁBACO



Representação do número

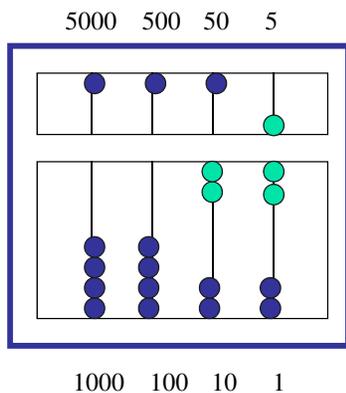
27

$$27 = 20 + 7$$

8

Primeiros Métodos de Cálculo

ÁBACO



Representação do número

27

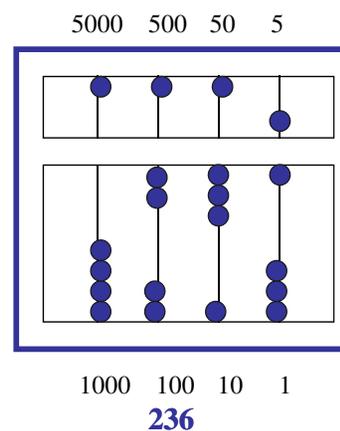
$$27 = 20 + 7$$

9

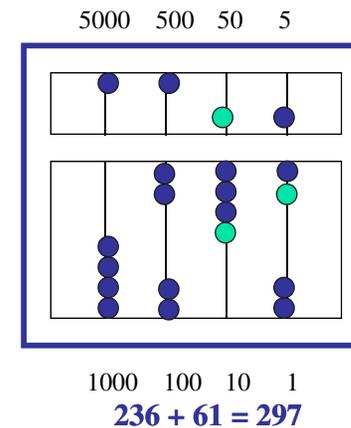
Primeiros Métodos de Cálculo

ÁBACO

SOMA EFETUADA NO ÁBACO $236 + 61 = ?$



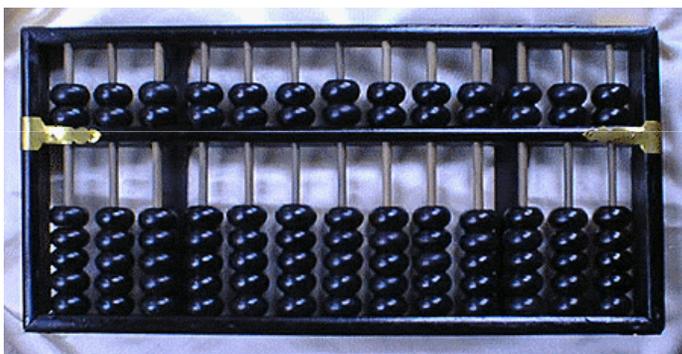
1000 100 10 1
236



1000 100 10 1
 $236 + 61 = 297$

10

Abacus

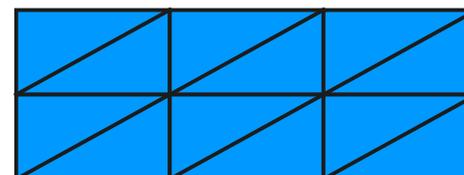


11

Auxílios Manuais nos Cálculos Escritos

MULTIPLICAÇÃO DOS ÁRABES

- O método de **multiplicação** utilizado hoje é uma variação de um método **tabular** desenvolvido pelos **árabes**



12

Auxílios Manuais nos Cálculos Escritos

Multiplicação dos Árabes

É feito o produto de cada dígito do número 217 por 1

É feito o produto de cada dígito do número 217 por 4

$$217 \times 14 = ?$$

	2	1	7
1	2	1	7
4	8	4	2
			8

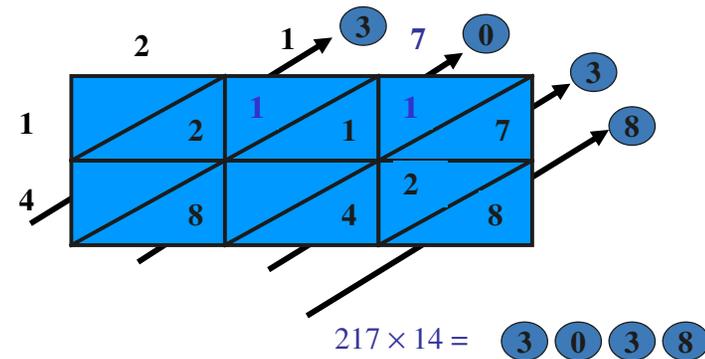
O produto é a **soma** dos dígitos nas diagonais

13

Auxílios Manuais nos Cálculos Escritos

Multiplicação dos Árabes

$$217 \times 14 = ?$$



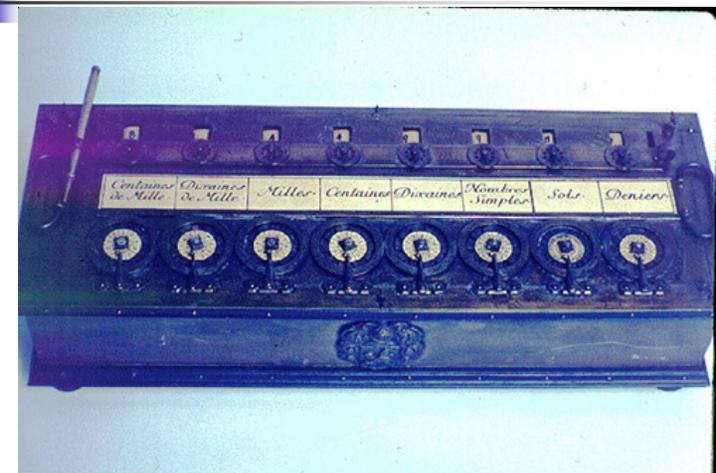
14

Auxílios Mecânicos para Cálculos

- **1642** – Blaise Pascal (filósofo francês) com 19 anos construiu a “**Máquina de Somar**”
 - Pascalina (Pascaline)
 - Para auxiliar seu pai – coletor de impostos
- A máquina constituía-se de **engrenagens mecânicas** para cada dígito
- O resultado era produzido mecanicamente

15

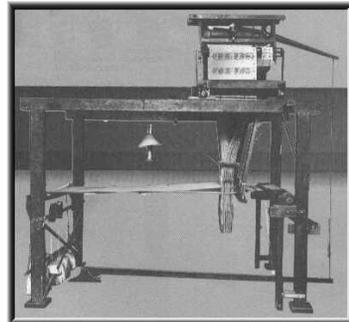
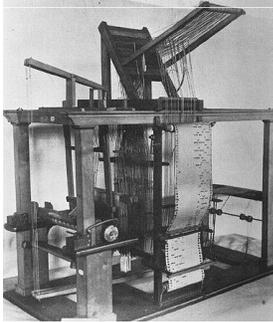
Máquina de Somar – Pascalina



16

Outros Auxílios Mecânicos

- **Séculos 18 e 19** – Teares baseados em papel perfurado



17

Auxílios Mecânicos para Cálculos Charles Babbage (matemático inglês)

- **Década de 1820** – “Engenho Diferencial”
 - Projetado para cálculo automático de tabelas de valores de funções polinomiais, nunca foi concluído
- **Década de 1830** – “Engenho Analítico”
 - Projeto ainda mais complexo, era para ser uma máquina programável via cartões perfurados, com capacidade de armazenamento (memória) para cálculos sequenciais (loops) e até dispositivo de saída (impressão)
 - Babbage é considerado por muitos o “pai do computador”, pois vários elementos estão presentes em computadores modernos

18

Máquina de Tabulação

- Desenvolvida em 1890 por Herman Hollerith para processar dados do censo americano
 - os dados eram perfurados em cartões que podiam ser classificados por meio de pinos que passavam pelos furos.
 - O Censo de 1880 demorou 12 anos para ter seus resultados compilados
 - Censo de 1890, com a máquina de Hollerith, foi processado em 3 anos
 - Montou a *Tabulating Machine Company*, uma das empresas que formou a IBM

19

Gerações Modernas

- **Primeira Geração**
 - Circuitos eletromecânicos e/ou eletrônicos que usavam relés e/ou válvulas como seus principais componentes
 - Operações internas chegando a milissegundos (10^{-3} segundos)
 - Z1 / Z2 / Z3, ENIAC, EDSAC, EDVAC, entre outros

20

Z1, Z2 e Z3

- **Décadas de 1930 e 1940**
 - Computadores eletromecânicos
 - Construídos pelo alemão Konrad Zuse
 - Destruídos na II guerra mundial

21

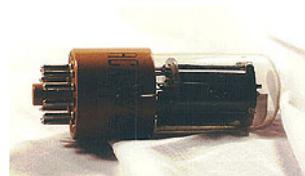
ENIAC - *Electronic Numerical Integrator and Calculator*

- Computador eletrônico / eletromecânico
- Universidade da Pensilvânia
- Levou 3 anos para ser construído: **1943 -1946**
- Possuía milhares de válvulas e relés
- Pesava várias toneladas
- Programação via fios / pinos, podia levar semanas

22

Válvula

- Também chamada de tubo de vácuo
- Dispositivo que pode ser utilizado para amplificar sinais eletrônicos
- Trabalham com voltagens elevadas



23



<http://en.wikipedia.org>

ENIAC

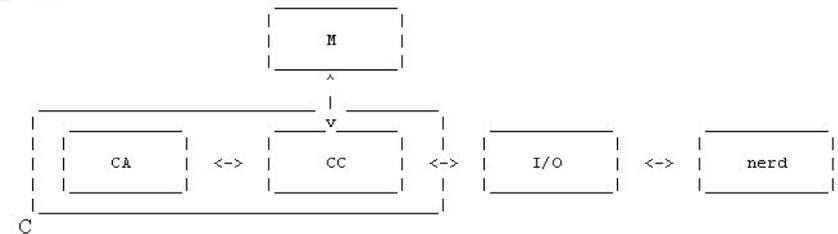
24

Outros Marcos da 1ª Geração

- Computadores Baseados na Arquitetura de Von Neumann:
 - **EDSAC** (1948): Universidade de Cambridge (UK)
 - **EDVAC** (1952): Universidade de Princeton (EUA)
 - ...
- Início da Produção em Escala Comercial (década de 1950)
 - **UNIVAC, IBM 704, ...**
- 1ª Linguagem de Alto Nível: **FORTRAN** (década de 50)

25

Arquitetura de von Neumann



Conceito de “Programa Armazenado”

M → Memória (dados + **programa**)

CA → Controlador Aritmético

CC → Controlador Central

I/O → Dispositivo de Entrada e Saída

26

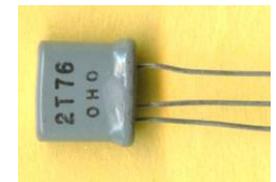
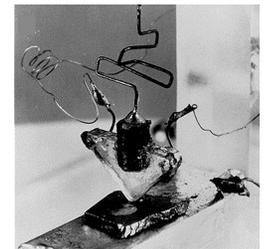
Gerações Modernas

- Segunda Geração (décadas 1950 – 60)
 - Válvulas foram substituídas por transistores
 - Computadores se tornaram “menores”, além de mais rápidos, baratos e confiáveis
 - Operações em micro-segundos
 - 10^{-6} segundos
 - Alavancou a produção de computadores em escala comercial

27

Transistor

- Pode regular o fluxo de corrente ou voltagem
 - Age como uma porta para sinais elétricos
 - Formado por três camadas de material semiconductor



28

Gerações Modernas



- Terceira Geração (décadas de 1960 - 70)
 - Circuitos integrados (SSI e MSI)
 - Circuito eletrônico formado por componentes
 - organizados em um **chip** (“pastilha” de semiconductor) de poucos centímetros ou milímetros quadrados
 - **SSI** - integração em pequena escala
 - menos de 10 componentes por chip
 - **MSI** - integração em média escala
 - 10 a 100 componentes por chip
 - Operações em nano-segundos
 - 10^{-9} segundos

29

Gerações Modernas

- Quarta Geração (década de 1970 em diante)
 - Tecnologia de firmware, integração em escalas superiores (LSI, VLSI, SLSI, ULSI)
 - **Firmware**: programa armazenado em **chip**
 - **LSI** - integração em grande escala (100 a 5.000 componentes)
 - **VLSI** - integração em muito grande escala (5.000 a 50.000)
 - **SLSI** - integração em super grande escala (50.000 a 100.000)
 - **ULSI** - integração em ultra grande escala (> 100.000)
 - Operações em pico-segundos (10^{-12} segundos)
 - Microprocessadores
 - Lei de Moore...

30

Computador Pessoal & Internet

- A partir do final dos anos 1970 (início dos anos 1980 no Brasil)...
- ...computador passa a se transformar em mais um **eletrodoméstico**
- Popularização é alavancada com a **internet** a partir dos anos 1990

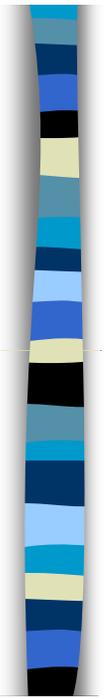


Estado da Arte

- **Robótica e Automação**
 - Telemetria, Controle, ...
- **Conectividade**
 - Redes de Computadores, Internet, ...
- **Software Inteligente**
 - Inteligência Artificial, ...
- **Computação Ubíqua**
 - Mobilidade, Personalização, ...
- **Convergência de Mídias e Tecnologias Digitais**
 - TV, Telefone, Computador, ...
- ...



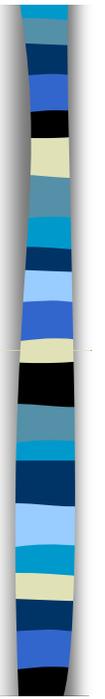
32



PARTE 2

Noções de Computadores

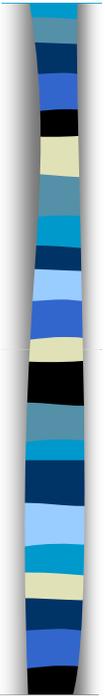
33



Hardware e Software

- **Hardware**
 - As partes físicas, palpáveis de um computador
- **Software**
 - Programas e dados
 - Um programa é um conjunto de instruções
 - Ex.: Sistemas operacionais, aplicativos
- **Hardware & Software**
 - Um computador precisa de ambos
 - Cada um é inútil sem o outro

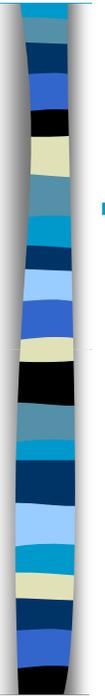
34



Informação Digital

- Computadores digitais armazenam e processam toda informação **digitalmente**
 - Números
 - Textos
 - Figuras e imagens
 - Sons
 - Instruções de um programa
 - ...
 - Enfim, toda informação é digitalizada, representada como **números binários**

35



Informação Digital

- Por exemplo, caracteres são armazenados como números (veremos mais detalhes depois)
 - Incluindo espaços, dígitos, sinais de pontuação, etc.

Oi, Joao!

79 106 44 74 111 97 111 33

Mas como os **números** são representados ?

36

Números Binários

- Toda informação digital é representada e armazenada na memória do computador usando o sistema de números binários
 - Um dígito binário (0 ou 1) é chamado de **bit**
 - *Dispositivos que armazenam e movem informação são mais baratos e confiáveis se eles têm que representar apenas dois valores (biestáveis)*
 - Um bit pode representar dois estados possíveis
 - Como uma luz que está ligada (1) ou desligada (0)
 - Combinações de bits são usadas para armazenar valores mais complexos

37

Combinações de Bits

1 bit	2 bits	3 bits	4 bits
0	00	000	0000 1000
1	01	001	0001 1001
	10	010	0010 1010
	11	011	0011 1011
		100	0100 1100
		101	0101 1101
		110	0110 1110
		111	0111 1111

Cada bit adicional dobra o número de possíveis combinações

38

Combinações de Bits

- Existem 2^N combinações de N bits
 - Assim, N bits são necessários para representar 2^N itens

Quantos itens podem ser representados por:

1 bit ?	$2^1 = 2$ itens
2 bits?	$2^2 = 4$ itens
3 bits?	$2^3 = 8$ itens
4 bits?	$2^4 = 16$ itens
5 bits?	$2^5 = 32$ itens

39

Sistema de Numeração Binária

- É um sistema de **numeração posicional**
 - valor de um dígito é dado pela sua posição no número
- No sistema **decimal** (base 10) o número 2562 tem a seguinte interpretação:

$$\begin{aligned}
 2 \times 1000 \quad (10^3) &= 2000 \\
 5 \times 100 \quad (10^2) &= 500 \\
 6 \times 10 \quad (10^1) &= 60 \\
 2 \times 1 \quad (10^0) &= 2
 \end{aligned}$$

40

Sistema de Numeração Binária

- No sistema **binário** de numeração a base é **2**
- Assim o valor do número 110101 é:

$$\begin{aligned} & 1 \times 2^5 + 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 \\ & = 32 + 16 + 0 + 4 + 0 + 1 \\ & = 53 \end{aligned}$$

- OBS. 1,78 pode ser representado por 178 e -2
 - $1,78 = 178 \times 10^{-2}$

41

Sistema de Numeração Binária

- Um bit pode ser utilizado para representar o sinal
 - Representação sinal-magnitude de 16 bits:
 - 1 bit** para sinal (0 positivo e 1 negativo) e **15 bits** para o número



- Nesse tipo de representação, qual faixa de valores inteiros pode ser representada?

42

Sistema de Numeração Binária

0 1 1 1 1 1 1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

$2^{14} 2^{13} 2^{12} 2^{11} 2^{10} 2^9 2^8 2^7 2^6 2^5 2^4 2^3 2^2 2^1 2^0$

$$\begin{aligned} & 1 \times 2^{14} + 1 \times 2^{13} + \dots + 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = 32767 \\ & = 2^{15} - 1 \quad (2^{15} = 32768 \rightarrow 0 \text{ a } 32767) \end{aligned}$$

- Logo, com 15 bits mais um bit de sinal, podemos representar os nos. inteiros de -32767 a $+32767$

43

Representando Textos Digitalmente

- Realizado através de um esquema de codificação
- Métodos de codificação mais utilizados:
 - Código **ASCII** (7 bits) - *American Standard Code for Information Interchange*
 - Códigos **ASCII Estendidos** (8 bits ou 1 byte)
 - Código **Unicode** (até 32 bits ou 4 bytes)

44

Representando Textos Digitalmente

■ Código ASCII (7 bits)

- Cada byte armazena um caractere: algarismo, letra, símbolo ou caractere de controle

CARACTERE	CÓDIGO							
0	0	1	1	0	0	0	0	48
9	0	1	1	1	0	0	1	57
A	1	0	0	0	0	0	1	65
a	1	1	0	0	0	0	1	97
P	1	0	1	0	0	0	0	80
+	0	1	0	1	0	1	1	43
7 BITS "MENORES" DE UM BYTE								

45

Representando Textos Digitalmente

■ Código ASCII (7 bits)

- Possibilidade de 2^7 representações diversas

- 128 caracteres

- Alfabeto inglês em letras minúsculas e maiúsculas (52)
- Caracteres decimais numéricos (10)
- Caracteres especiais (33)
- Caracteres de controle não imprimíveis (33)

46

Tabela ASCII

Dec Char	Dec Chr	Dec Chr	Dec Chr
0 NUL (null)	32 Space	64 @	96 `
1 SOH (start of heading)	33 !	65 A	97 a
2 STX (start of text)	34 "	66 B	98 b
3 ETX (end of text)	35 #	67 C	99 c
4 EOT (end of transmission)	36 \$	68 D	100 d
5 ENQ (enquiry)	37 %	69 E	101 e
6 ACK (acknowledge)	38 &	70 F	102 f
7 BEL (bell)	39 '	71 G	103 g
8 BS (backspace)	40 (72 H	104 h
9 TAB (horizontal tab)	41)	73 I	105 i
10 LF (NL line feed, new line)	42 *	74 J	106 j
11 VT (vertical tab)	43 +	75 K	107 k
12 FF (NP form feed, new page)	44 ,	76 L	108 l
13 CR (carriage return)	45 -	77 M	109 m
14 SO (shift out)	46 .	78 N	110 n
15 SI (shift in)	47 /	79 O	111 o
16 DLE (data link escape)	48 0	80 P	112 p
17 DC1 (device control 1)	49 1	81 Q	113 q
18 DC2 (device control 2)	50 2	82 R	114 r
19 DC3 (device control 3)	51 3	83 S	115 s
20 DC4 (device control 4)	52 4	84 T	116 t
21 NAK (negative acknowledge)	53 5	85 U	117 u
22 SYN (synchronous idle)	54 6	86 V	118 v
23 ETB (end of trans. block)	55 7	87 W	119 w
24 CAN (cancel)	56 8	88 X	120 x
25 EM (end of medium)	57 9	89 Y	121 y
26 SUB (substitute)	58 :	90 Z	122 z
27 ESC (escape)	59 ;	91 [123 {
28 FS (file separator)	60 <	92 \	124
29 GS (group separator)	61 =	93]	125 }
30 RS (record separator)	62 >	94 ^	126 ~
31 US (unit separator)	63 ?	95 _	127 DEL

Source: www.LookupTables.com

128 Ç	144 É	161 í	177	193	209	225	241
129 ù	145 æ	162 ó	178	194	210	226	242
130 é	146 Æ	163 ú	179	195	211	227	243
131 â	147 ô	164 ñ	180	196	212	228	244
132 ä	148 ö	165 Ñ	181	197	213	229	245
133 à	149 ò	166 ¨	182	198	214	230	246
134 â	150 û	167 °	183	199	215	231	247
135 ç	151 ù	168 ¸	184	200	216	232	248
136 ê	152	169	185	201	217	233	249
137 ë	153 Ö	170	186	202	218	234	250
138 è	154 Ü	171 ¼	187	203	219	235	251
139 ì	156 £	172 ¾	188	204	220	236	252
140 î	157 ₣	173 ¡	189	205	221	237	253
141 ï	158	174 «	190	206	222	238	254
142 Æ	159 f	175 »	191	207	223	239	255
143 Å	160 á	176	192	208	224	240	255

Source: www.LookupTables.com

■ Exemplo de tabela ASCII estendida (8 bits)

- Atenção: existem vários padrões diferentes
- OBS: <ALT> + Código (no teclado numérico, com ele ativado) → caractere

48

Representando Textos Digitalmente

- Código **Unicode** (www.unicode.org)
 - Desenvolvido para unificar os diversos padrões de codificação existentes
 - eliminar incompatibilidades entre diferentes sistemas
 - Suporta até 32 bits, possibilitando que muito mais caracteres sejam codificados (até 2^{32})
 - Repertório atual contém mais de 100.000 caracteres
 - Abrange caracteres de todas as línguas
 - Ex.: grego, hebreu, japonês, ...
 - Tem sido adotado por diversos fabricantes e desenvolvedores de software

49

Hardware

- Arquitetura básica de um computador
 - **CPU**
 - Unidade de controle
 - Unidade aritmética e lógica
 - **Memória**
 - Memória principal
 - Memória auxiliar (ou secundária)
 - **Dispositivos de Entrada e Saída**

50

CPU

- Unidade Central de Processamento
 - Coordena e executa
 - Instruções dos programas
 - Operações aritméticas e lógicas
 - Divide-se em:
 - Unidade de controle
 - Controla funcionamento do computador
 - Unidade lógica e aritmética (ULA)
 - Realiza operações sobre dados

51

Memória Principal



- Também denominada memória **RAM**
 - *Random Access Memory*
- É muito rápida quando comparada aos dispositivos de memória auxiliar
- Porém, só é capaz de armazenar dados enquanto o computador está ligado

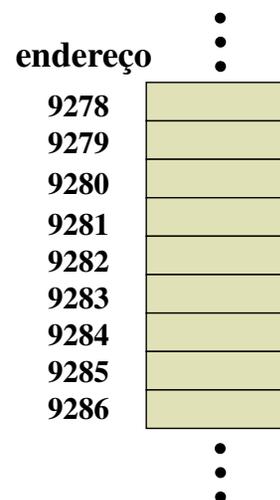
52

Memória Principal

- A **memória RAM** do computador é dividida em unidades de mesmo tamanho, chamadas **PALAVRAS**
- Cada palavra de memória possui um endereço que a identifica unicamente

53

Organização da Memória



O endereço de cada posição de memória (palavra) a identifica unicamente

Os endereços são permanentes (vêm da fábrica) e não podem ser modificados pelo programador

54

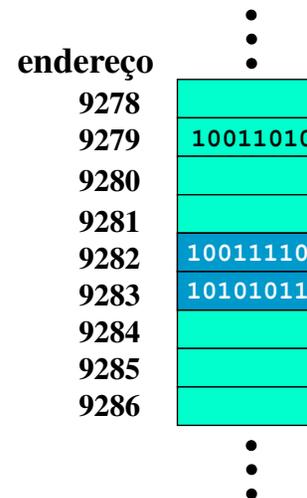
Organização da Memória

- A PALAVRA é formada por um conjunto de **bits**
 - quantidade depende da arquitetura do computador
- Exemplo:
 - Palavra de 4 bits

0	bit 1	bit 2	bit 3	bit 4
1				
2				
3				
...

55

Organização da Memória



Exemplo de palavra de 8 bits (1 **byte**)

Valores grandes (maiores que o tamanho da palavra) podem ser armazenados em posições consecutivas

56

Organização da Memória

- BIT (*binary digit*) - dígitos binários
- BYTE (*binary term*) : Unidade básica da informação
- O byte é composto por 8 BITS



57

Capacidade de Memória

- Todo dispositivo de memória tem uma determinada capacidade de armazenamento
 - Capacidades são expressas em várias unidades:

Medida	No. Bits / Bytes	Armazena
1 byte	8 bits	1 caractere
1 kilobyte (KB)	2^{10} ou 1024 bytes	~ meia página escrita
1 megabyte (MB)	2^{20} ou 1024 KB	~ 500 páginas escritas
1 gigabyte (GB)	2^{30} ou 1024 MB	~ 500.000 páginas escritas
1 terabyte (TB)	2^{40} ou 1024 GB	Muita coisa...
Petabyte		
Exabyte		
Zettabyte		
yottabyte		

Memória Auxiliar

- Historicamente, dispositivos magnéticos e óticos
 - Fitas Magnéticas, HDs, Discos Óticos (CDs, DVDs, ...)
 - Podem armazenar grandes quantidades de dados
 - Mas acesso aos dados é muito mais lento que em RAM
 - Dados devem primeiro ser transferidos para a memória principal antes de serem processados
 - Dados não são perdidos quando computador é desligado
 - Memórias não voláteis
- Mais recentemente, incluem também os “Pen-Drives”
 - Memórias “flash” (eletrônicas)

59

Unidades de Entrada e Saída

- **Entrada**
 - Teclado, mouse, microfone, leitor ótico (código de barras), tela sensível ao toque, etc.
- **Saída**
 - Monitor de Vídeo, Impressora, Speakers, etc.
- **Nota**
 - Dispositivos de memória secundária podem ser considerados unidades de Entrada e Saída (E/S)

60

Software

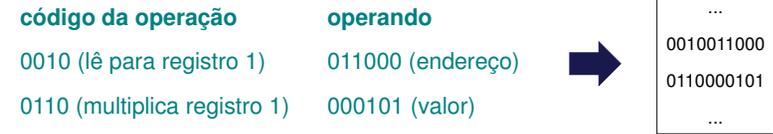
- O Software inclui
 - PROGRAMAS que, quando executados, realizam a tarefa desejada
 - DOCUMENTOS que descrevem a operação e o uso dos programas
 - ESTRUTURAS DE DADOS que possibilitam que os programas manipulem adequadamente a informação

61

Programação e Níveis de Linguagem

■ Linguagem de Máquina

- É o conjunto das instruções primitivas projetadas para um computador
 - Uma CPU somente pode compreender instruções que sejam expressas em termos de sua LINGUAGEM DE MÁQUINA
- Um programa escrito em linguagem de máquina consiste de uma série de números binários e é muito difícil de ser entendido pelas pessoas
- Exemplo: Cada instrução é constituída de 2 partes:



Programação e Níveis de Linguagem

■ Linguagem de Baixo Nível

- Os programas são escritos em uma notação que está próxima da linguagem de máquina
- Exemplo:

código da operação	operando	significado
LD	F0	lê F0
MPI	05	multiplica 5

63

Programação e Níveis de Linguagem

◆ Linguagem de Alto Nível

- Permitem escrever programas em uma notação próxima à maneira natural de expressar o problema que se deseja resolver
- Exemplo: $A = 2; B = 10; C = -5; D = 28;$
 $RESULT = D - ((A + B) / C)$
- Algumas Linguagens: FORTRAN, BASIC, COBOL, PASCAL, C, C++, Java, LISP, PROLOG, ...

64

Programação e Níveis de Linguagem

■ **COMPILADOR**

- Traduz os comandos simbólicos de uma linguagem de alto nível para linguagem de máquina

■ **MONTADOR (ASSEMBLER)**

- Traduz os comandos simbólicos de uma linguagem de baixo nível para linguagem de máquina

■ **INTERPRETADOR**

- Lê e executa uma declaração do programa por vez
- A execução do programa requer que o interpretador da linguagem esteja sendo executado no computador

65

Exercícios

- Mostre em detalhes a conversão para decimal dos seguintes números binários:
 - 10001001, 01101010 e 00111101
- Sejam os números 16, 17, 235, 418 (base 10):
 - Para cada um deles, quantos bits são necessários para representá-los na base 2 (dado que também devemos representar todos os seus predecessores inteiros não-negativos)? Justifique
 - Existe uma forma matemática (e não por tentativa e erro) de responder essa questão? Explique

66

Exercícios

- Utilizando 1 byte completo, mostre e explique a conversão para binário
 - dos caracteres ASCII: "a", "H", "5", "?", "!"
 - dos números 115, 35, -10 (usando bit de sinal)

67

Agradecimentos

- Prof. André de Carvalho
 - ◆ Slides baseados nos originais das professoras Rosely Sanches, Sandra Aluísio, Solange Rezende e Renata Fortes
- Prof. Rudinei Goularte
 - ◆ Slides baseados nos originais da Profa. Rosely Sanches

68