ENGENHARIA DE COMPUTAÇÃO

ARQUITETURA E ORGANIZAÇÃO BÁSICA DE COMPUTADORES – TEORIA

Exercícios:

Listas 1, 2 e 3

Turma: CP300TIN2

João Pedro de Oliveira Grangeiro - 222507
Leonardo Rossi de Oliveira - 222410
Lucas Camargo Oliveira - 222231
Natã Camargo Oliveira - 210399

Professor: Rafael Rodrigues da Paz

Sorocaba / SP
01/04/23

Arquitetura de computadores (Lista 1)

Aula 1 - Exercícios (grupo 4)

1) Qual a diferença entre organização básica e arquitetura?
R: ambos são conceitos relacionados à área de computação, mas se referem a aspectos diferentes do funcionamento de um sistema operacional. A organização básica se concentra na organização física do sistema, já a arquitetura refere-se à maneira como o Software interage com o Hardware.

2) Qual a principal vantagem em usar computadores da mesma família de arquitetura?
R: as principais vantagens em usar computadores da mesma família de arquitetura é a compatibilidade entre sistemas e programas, ou seja, quando os computadores provem da mesma arquitetura, eles são capazes de executar o mesmo conjunto de instruções e lidar com os mesmos formatos de dados.

3) Quais as principais funções de um computador?
R: 1- Processamento de dados: é a principal função de um computador.
2- Armazenamento de dados: ele pode armazenar uma grande quantidade de informações em discos rígidos, memórias, pendrives e outras mídias.
3- Entrada de dados: permitem que os dados sejam inseridos pelos usuários.

4- Saída de dados: exibem informações em diferentes formatos, através de periféricos de saída.
5- Comunicação: permite a interação com outros dispositivos.
6- Controle de dispositivos externos: os computadores podem controlar dispositivos externos, como robôs, sensores e atuadores.

(4) Quais os passos que uma movimentação de dados segue para ser realizada?
R: 1º Passo: Identificação da fonte - o primeiro passo é identificar a fonte dos dados que serão movidos.
2º Passo: Preparação dos dados - antes de realizar a movimentação dos dados, eles precisam ser preparados.
3º Passo: Transferência de dados - o próximo passo é movê-los para o destino.
4º Passo: Recebimento dos dados - logo após a transferência, os dados são recebidos no destino.
5º Passo: Armazenamento dos dados - por último, os dados são armazenados no dispositivo.

(5) Quais os passos que uma operação de processamento segue para ser realizada?
R: 1º Passo: Identificação da Tarefa - o primeiro passo é identificar qual tarefa será realizada pelo computador.
2º Passo: Preparação dos dados - antes de processar dados, é necessário prepará-los.
3º Passo: Interpretar as instruções - o computador

preciso interpretar as intruções necessárias para realizar a tarefa.

4º passo: Executar as instruções - após o 3º passo, o computador executa a tarefa de acordo com as instruções.

5º Passo: Verificação dos erros - o computador analisa se houveram erros de execução.

6º Passo: Geração de resultados - Por fim, o dispositivo gera os resultados da tarefa.

(6) Como é realizada a conversão de decimal para Hexadecimal? Explique com um exemplo, converta 120 para Hexadecimal.

R: Para realizar essa conversão, devemos dividir o número por 16 repetidamente, anotando o resto de cada divisão e os convertendo em dígitos Hexadecimais.

Ex: 120 -> 78 (Hex)

$$\begin{array}{r|l} 120 & 16 \\ \hline -112 & 7 \\ 8 & \end{array}$$

-> O quociente é 7 e o resto é 8. Não é mais possível continuar a divisão, portanto o número decimal 120 equivale a 78 em Hexadecimal.

(7) Converta para binário:

A) 126 = 1111110

126 | 2
0 63 | 2
1 31 | 2
1 15 | 2
1 7 | 2
1 3 | 2
1 1 | 2
1 0

B) 333 = 101001101

333 | 2
1 166 | 2
0 83 | 2
1 41 | 2
1 20 | 2
0 10 | 2
0 5 | 2
1 2 | 2
0 1 | 2
1 0

7-(C) 991 = 1111011111

991 | 2
1   495 | 2
1   247 | 2
1   123 | 2
1   61 | 2
1   30 | 2
0   15 | 2
1   7 | 2
1   3 | 2
1   1 | 2
1   0

(8) Converta para decimal:

A) 10000 = 16

$\overset{4}{1}\overset{3}{0}\overset{2}{0}\overset{1}{0}\overset{0}{0} = 1 \times 2^4 + 0 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 0 \times 2^0 = 16$

B) 1111000 = 120

$\overset{6}{1}\overset{5}{1}\overset{4}{1}\overset{3}{1}\overset{2}{0}\overset{1}{0}\overset{0}{0} = 1 \times 2^6 + 1 \times 2^5 + 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 0 \times 2^0$
$= 120$ (dec)

C) 1000001 = 65

$\overset{7}{1}\overset{6}{0}\overset{5}{0}\overset{4}{0}\overset{3}{0}\overset{2}{0}\overset{1}{0}\overset{0}{1} = 1 \times 2^7 + 0 \times 2^6 + 0 \times 2^5 + 0 \times 2^4 + 0 \times 2^3 + 0 \times 2^2 + 0 \times 2^1$
$+ 1 \times 2^0 = 65$ (dec).

Arquitetura de computadores (Lista 2)

* Aula 2 Arquitetura de Computadores

Exercícios:

1) Converta os números de binário para decimal:

A) $10011 = 1 \times 2^4 + 0.2^3 + 0.2^2 + 1.2^1 + 1.2^0 = 19$

B) $111011 = 59 \rightarrow (1\times2^5 + 1\times2^4 + 1\times2^3 + 1\times2^1 + 1\times2^0) = 59$

C) $11101111 = 239 \rightarrow (1\times2^7 + 1\times2^6 + 1\times2^5 + 1\times2^3 + 1\times2^2 + 1\times2^1 + 1\times2^0) = 239$

D) $011011101111 = 1775 \rightarrow (1\times2^{10} + 1\times2^9 + 1\times2^7 + 1\times2^6 + 1\times2^5 + 1\times2^3 + 1\times2^2 + 1\times2^1 + 1\times2^0) = 1775$

2) Qual é a melhor forma de representação números inteiros: Sinal de magnitude ou complemento de dois?

R: O complemento de dois é considerado a melhor forma de representação de números inteiros por diversos fatores: 1º fator: tem a propriedade de que a soma e a subtração de números inteiros são realizadas de maneiras semelhantes. 2º fator: permite que a representação de zero seja única. 3º fator: sua representação é facilmente estendida para representar números negativos com precisão arbitrária.

3. Demonstre a faixa de funcionamento dos números inteiros sendo sinalizado e não sinalizado com:

A) 12 bits: Sinalizado: (-2^{11}) a $(2^{11}-1) = -2048$ a 2047

Não sinalizado: $(2^{12})-1 = 0$ a 4095

tilibra

B) 20 bits: Sinalizado: (-2^{19}) a $(+2^{19}-1) = -524288$ a 524287
Não Sinalizado: $(2^{20})-1 = 0$ a 1048575

C) 24 bits: Sinalizado: (-2^{23}) a $(2^{23}-1) = -8388608$ a 8388607
Não Sinalizado: $(2^{24})-1 = 0$ a 16777215

4) Demonstre o complemento de dois dos números:

A) 23 (para 8 bits)

23 | 2
1 11 | 2
1 5 | 2
1 2 | 2
0 1

$10111 \rightarrow 00010111$ (8 bits)

(Converte e some +1)

11101000
+1
11101001 = -23

B) 127 (para 8 bits)

127 | 2
1 63 | 2
1 31 | 2
1 15 | 2
1 7 | 2
1 3 | 2
1 1

01111111
10000000
+1
10000001 = -127

C) 0 (8 bits)

0 = 00000000
11111111 = 0
+1
Complemen

D) 128 (para 8 bits)

128 | 2
0 64 | 2
0 32 | 2
0 16 | 2
0 8 | 2
0 4 | 2
0 2 | 2
0 1

128 = 10000000
01111111
+ 1
10000000 = -128

C) 3000 (para 16 bits)

3000 | 2
0 1500 | 2
0 750 | 2
0 375 | 2
1 187 | 2
1 93 | 2
1 46 | 2
0 23 | 2
1 11 | 2
1 5 | 2
1 2 | 2
0 1

3000 = 0000101110111000
1111010001000111
+ 1
1111010001001000
=
- 3000

5) Converta os números da base decimal para: Hexa e Binário.

A) 10

Binário: 10 |2   10 = 1010
0 5 |2
1 2 |2
0 1

Hexadecimal: 10 |16   10 = A
10 0

b) 64

Binário: 64 |2   64 = 1000 000
0 32 |2
0 16 |2
0 8 |2
0 4 |2
0 2 |2
0 1

Hexa: 64 |16 = 64 = 40
0 4

C) 121

Binário:

121 | 2
1 60 | 2
0 30 | 2
0 15 | 2
1 7 | 2
1 3 | 2
1 1

121 = 1111001

Hexa: 121 | 16
9 7

121 = 79

D) 1255

Binário : 1255 = 10011100111

1255 | 2
1 627 | 2
1 313 | 2
1 156 | 2
0 78 | 2
0 39 | 2
1 19 | 2
1 9 | 2
1 4 | 2
0 2 | 2
0 1

tilibra

Hexadecimal: 1255 = 4E7

$$\begin{array}{l} 1255 \,\underline{|16} \\ \quad 7 \quad 78 \,\underline{|16} \\ \qquad\quad 14 \quad 4 \end{array}$$

e) 512

Binário: 512 = 1000000000

$$\begin{array}{l} 512 \,\underline{|2} \\ \; 0 \quad 256 \,\underline{|2} \\ \qquad 0 \quad 128 \,\underline{|2} \\ \qquad\quad 0 \quad 64 \,\underline{|2} \\ \qquad\qquad 0 \quad 32 \,\underline{|2} \\ \qquad\qquad\quad 0 \quad 16 \,\underline{|2} \\ \qquad\qquad\qquad 0 \quad 8 \,\underline{|2} \\ \qquad\qquad\qquad\quad 0 \quad 4 \,\underline{|2} \\ \qquad\qquad\qquad\qquad 0 \quad 2 \,\underline{|2} \\ \qquad\qquad\qquad\qquad\quad 0 \quad 1 \end{array}$$

Hexa: 512 = 200

$$\begin{array}{l} 512 \,\underline{|16} \\ \; 0 \quad 32 \,\underline{|16} = 200 \\ \qquad 0 \quad 2 \end{array}$$

f) 497

Binário: 497 = 111110001

497 | 2
1 248 | 2
0 124 | 2
0 62 | 2
0 31 | 2
1 15 | 2
1 7 | 2
1 3 | 2
1 1

Hex: 497 = 1F1

497 | 16
1 31 | 16 = 1F1
15 1

6) Converta números da base Hexadecimal para decimal e binário:

A) 36 : "3" = $3 \times 16 = 48$ e "6" = 6 = $48 + 6 = 54$ decimal
Binário: 00110110

B) 2000 : Hexa = 2000
Dec : "2" = $2 \times 16 = 32^3 = 8192$
Bi : 10000000000000

C) ABCD: Decimal: A = $10 \times 16^3 = 40960$
B = $11 \times 16^2 = 2816$
C = $12 \times 16^1 = 192$
D = 13
= 43981
Binário: 1010101111001101

D) 1204 : Decimal 4612
Binário: 1001011000100

E) 13333 : Decimal 13123
Binário: 0011001100110011

7) Considere os números decimais apresentados abaixo:

A) 36 e 40 B) 20 e 20
C) 123 e 100 D) 240 e 204

Efetue a soma em binário e indique carry e overflow. Usar operações de 8 bits.

A) 36 e 40: 36 = 00100100

40 = 00101000

soma = 01001100

* Não houve carry e nem overflow, estão todos os bits dentro do limite de 8 bits.

B) 20 e 20: 20 = 00010100 + 00010100

Soma = 00101000

x Não houve nada, estão dentro dos limites

C) 123 e 100: 123 = 01111011

100 = 01100100

7 6 5 4 3 2 1 0

soma = 11011111

* há carry no bit 8, pois ultrapassa o limite

* Não há overflow, porque o último bit não sofreu alteração.

D) 240 e 204: 240 = 11110000

204 = 11001100

soma = 11011100

* Há carry no bit 8, pois 112 + 128 resultam em 240 e ultrapassam o limite.

* há overflow no bit 8, que também ultrapassa o limite de 8 bits.

8) Considere os números decimais apresentados nas letras abaixo:

A) 36 e 40 B) 20 e 20 C) 123 e 100

D) 240 e 204

Efetue a subtração em binário e indique se houve carry e overflow. Usar operações de 8 bits.

A) 36 - 40

36 → 00100100 | 11011011
40 → 00101000 | +1
11011100

11011100 | 36
00101000 | -40
1 00000100 → -4

→ Carry

B) 20 - 20
20 → 00010100 → 11101011
+1
11101100

00010100 | 20
11101100 | -20
100000000 | 0
→ Carry

C) 123 − 100

{ 123 → 01111011
{ 100 → 01100100 → 10011011

+ 1

10011100

01111011 } 123
1 10011100 } −100
00010111 → 23
Carry

D) 240 − 204

{ 240 → 11110000
{ 204 → 11001100 → 00110011

+ 1

00110100

11110000 } 240
00110100 } −204
1 00100100 } 36
Carry

Arquitetura de computadores (Lista 3)

* Aula 3 Arquitetura de Computadores

Exercícios:

1) Converta os números de binário para decimal:

a) $\overset{4}{1}\overset{3}{0}\overset{2}{0}\overset{1}{1}\overset{0}{1} = 1\times2^4 + 1\times2^1 + 1\times2^0 = 19$

b) $111011.101 = \overset{5\,4\,3\,2\,1\,0}{111011} \;.\; \overset{-1\,-2\,-3}{101}$

$111011 \rightarrow (1\times2^5 + 1\times2^4 + 1\times2^3 + 1\times2^1 + 1\times2^0) = 59$

$101 \rightarrow (1\times2^{-1} + 0\times2^{-2} + 1\times2^{-3}) = 0{,}625$

R: $111011.101 = 59.625$

c) $111.11001 = \overset{2\,1\,0}{111} \;.\; \overset{-1\,-2\,-3\,-4\,-5}{11001}$

$111 \rightarrow (1\times2^2 + 1\times2^1 + 1\times2^0) = 7$

$11001 \rightarrow (1\times2^{-1} + 1\times2^{-2} + 1\times2^{-5}) = 0{,}78125$

R: 7.78125

2) Demonstre o complemento de 2 dos números:

A) 123 (8 bits)

123	2						
1	61	2					
	1	30	2				
		0	15	2			
			1	7	2		
				1	3	2	
					1	2	2
							1

123 → 01111011

10000100

\+ 1

10000101

B) 1 (8 bits)

1 = 00000001
11111110
+ 1
11111111

C) 118 (8 bits)

118 = 01110110
10001001
+ 1
10001010

D) 2800 (16 bits)

2800 = 0000101011110000
1111010100001111
+ 1
1111010100010000

3) Faça contas com os números inteiros (converte em binário). e indique se ocorreu ou não overflow.

A) 4 + 2 (8 bits)

4 → 00000100 } 4
2 → 00000010 } +2
00000110 } 6

B) 120 + 8 (8 bits)

120 → 01111000
8 → 00001000
10000000

$$\begin{array}{r} 120 \\ +\ 8 \\ \hline 128 \end{array}$$

C) 120 − 5 (8 bits)

120 → 01111000
5 → 00000101 → 1111010
+ 1
1111011

01111000
11111011
1 01110011
↳ Carry

$$\begin{array}{r} 120 \\ -\ 5 \\ \hline 115 \end{array}$$

D) 50 − 50 (8 bits)

50 → 00110010 → 1100110 1
+ 1
11001110

00110010
11001110
1 00000000
↳ Carry

$$\begin{array}{r} 50 \\ -\ 50 \\ \hline 0 \end{array}$$

e) 50 − 51 (8 bits)

50 → 00110010
51 → 00110011 → 11001100
+ 1
11001101

00110010 } 50
11001101 } −51
1 11111111 } −1
↳ Carry

f) 1000 − 500 (12 bits)

1000 → 001111101000
500 → 000111110100 → 111000001011
+ 1
111000001100

001111101000 } 1000
111000001100 } − 500
1 000111110100   500
↳ Carry

4) Converta os números binários em ponto flutuante conforme o padrão IEEE 754

A)	1	10111001	01011000000000000000000
	1	185 - 127 = 58	$\frac{1}{4} + \frac{1}{16} + \frac{1}{32}$ -> 0,34375

$\rightarrow -1^S (1,M) . 2^E \rightarrow -1^1 . (1,34375) 2^{58}$

R: $-3,8730957 \times 10^{17}$

B)	0	01111000	01100000000000000000000
	0 (+)	120 - 127 = -7	$\frac{1}{4} + \frac{1}{8}$ -> 0,375

$\rightarrow -1^S . (1,M) . 2^E \rightarrow 1^0 . (1,375) . 2^{-7} \rightarrow 0,010742187$

R: 0,0107421875