



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: ambas são condições relacionadas 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 possuem a 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, pen-drives e outros mídias.

3- Entrada de dados: permitem que os dados sejam inseridos pelos usuários.

- 4- Saída de dados: Envia 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 mouses, 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 enviados.

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 enviá-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 instruções 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 representando em dígitos Hexadecimais.

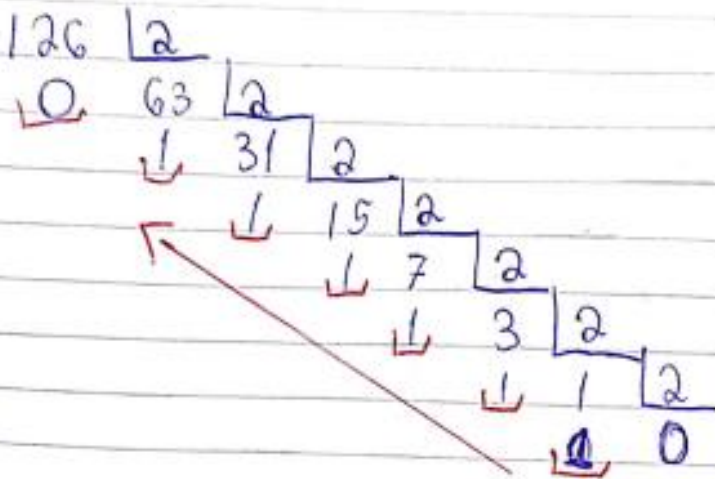
Ex: $120 \rightarrow 78$ (Hex)

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

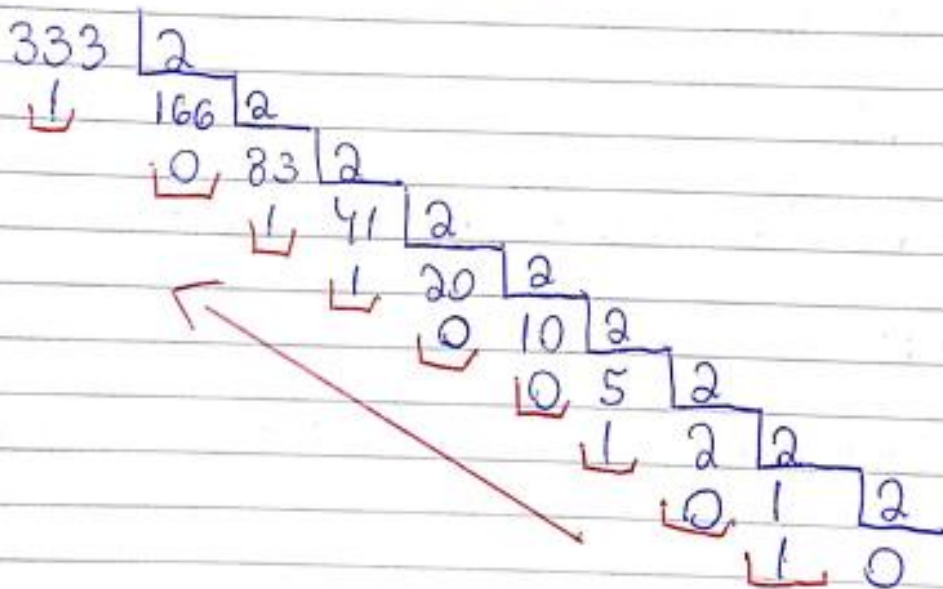
\rightarrow O quociente é 7 e o resto é 8. Não é mais possível continuar a divisão, portanto o número decimal 120 equivale a 78 em Hexadecimal.

(7) Converte para binário:

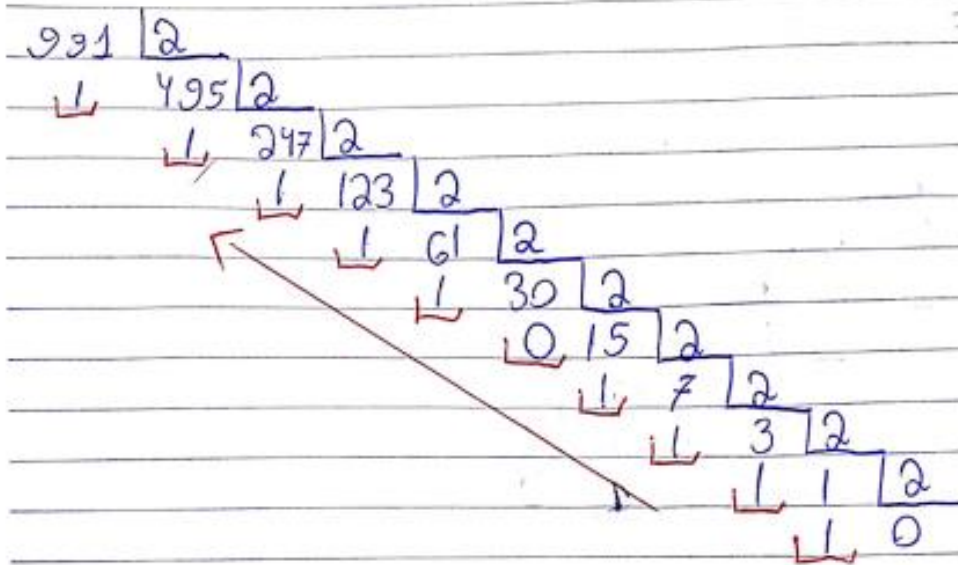
A) $126 = 1111110$



B) $333 = 101001101$



$$7-(C) \ 991 = 1111011111$$



(8) Converta para decimal:

$$A) 10000 = 16$$

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

$$B) 1111000 = 120$$

$$1111000 = 1 \times 2^6 + 1 \times 2^5 + 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 0 \times 2^0 = 120 \text{ (dec)}$$

$$C) 1000001 = 65$$

$$1000001 = 1 \times 2^7 + 0 \times 2^6 + 0 \times 2^5 + 0 \times 2^4 + 0 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 65 \text{ (dec)}$$

Arquitetura de computadores (Lista 2)

* Aula 2

Arquitetura de Computadores

Exercícios:

1) Converta os números de binário para decimal:

A) $10011^0 = 1 \cdot 2^4 + 0 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 19$

B) $111011^0 = 59 \rightarrow (1 \cdot 2^5 + 1 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^1 + 1 \cdot 2^0) = 59$

C) $1110111^0 = 239 \rightarrow (1 \cdot 2^7 + 1 \cdot 2^6 + 1 \cdot 2^5 + 1 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0) = 239$

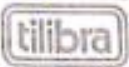
D) $011011101111^0 = 1775 \rightarrow (1 \cdot 2^{10} + 1 \cdot 2^9 + 1 \cdot 2^7 + 1 \cdot 2^6 + 1 \cdot 2^5 + 1 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0) = 1775$

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 o 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 

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

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

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

$\rightarrow 10111 \rightarrow$ (8 bits)
 0001011
 \downarrow
 (Converte e some +1)
 11101000
 $+1$
 $11101001 = -23$

B) 127 (para 8 bits)

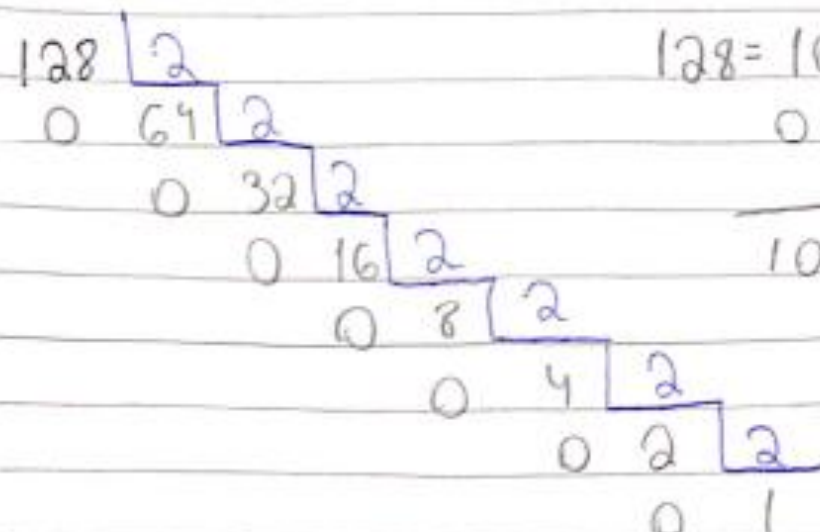
127 | 2
 1 63 2
 1 31 2
 1 15 2
 1 7 2
 1 3 2
 1 1

$\rightarrow 0111111$
 1000000
 $+1$
 $10000001 = -127$

C) 0 (8 bits)

$0 = 00000000$
 $11111111 = 0$
 $+1$
 Complemento

D) 128 (para 8 bits)



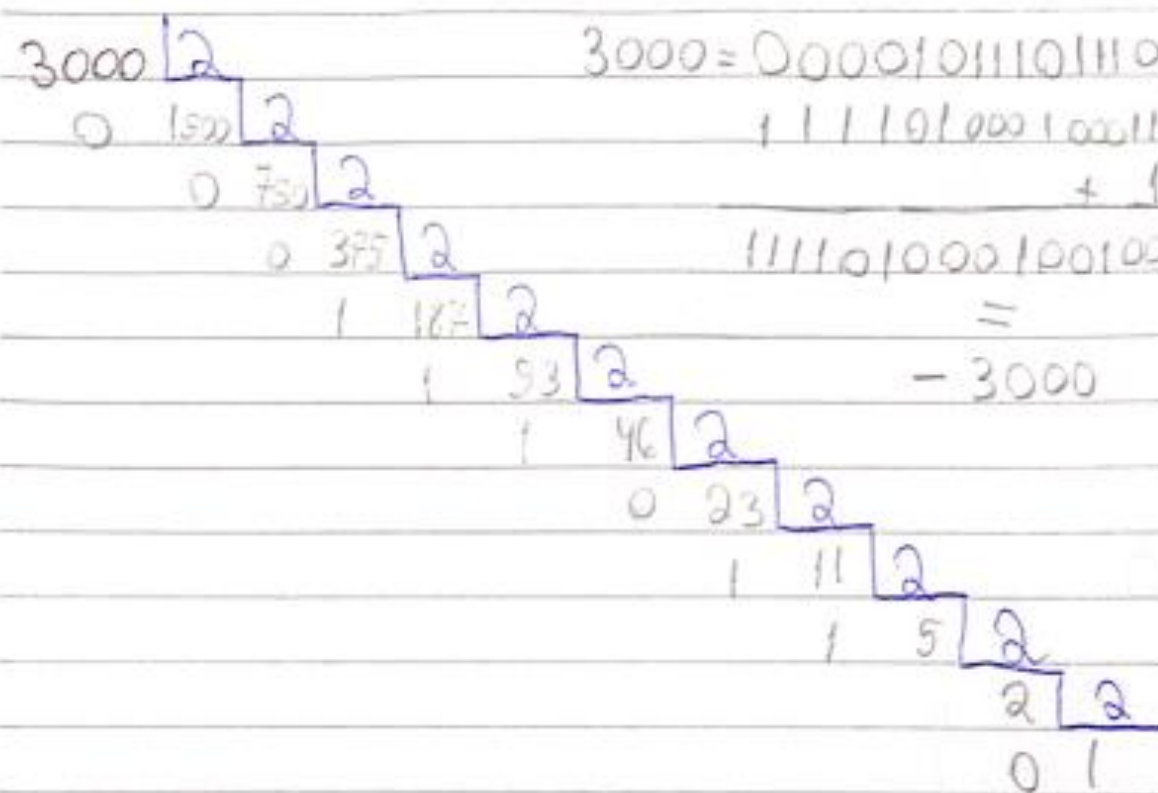
$$128 = 100000000$$

$$01111111$$

$$+ 1$$

$$100000000 = -128$$

E) 3000 (para 16 bits)



$$3000 = 000010111011000$$

$$1111010001000111$$

$$+ 1$$

$$1111010001001000$$

$$=$$

$$-3000$$

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

A) 10

Binário: $10 \begin{array}{l} 2 \\ 0 \end{array} \begin{array}{l} 2 \\ 5 \end{array} \begin{array}{l} 2 \\ 2 \\ 1 \end{array} \begin{array}{l} 2 \\ 2 \\ 0 \end{array} \begin{array}{l} 0 \\ 1 \end{array}$ $10 = 1010$

Hexadecimal: $10 \begin{array}{l} 16 \\ 10 \end{array} \begin{array}{l} 0 \\ 0 \end{array}$ $10 = A$

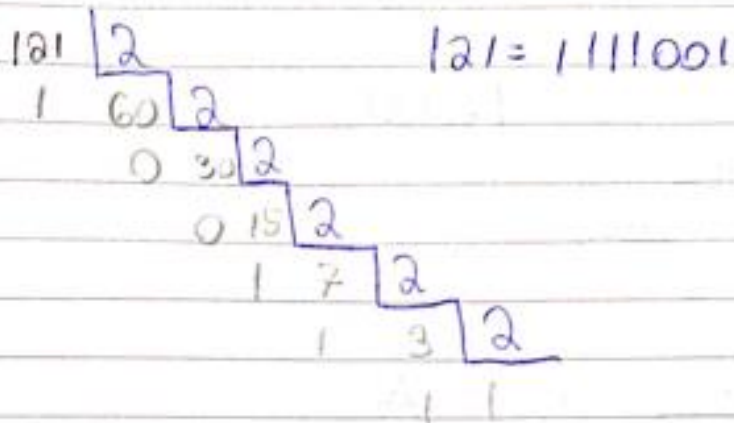
b) 64

Binário: $64 \begin{array}{l} 2 \\ 0 \end{array} \begin{array}{l} 32 \\ 16 \end{array} \begin{array}{l} 2 \\ 8 \end{array} \begin{array}{l} 2 \\ 4 \end{array} \begin{array}{l} 2 \\ 2 \end{array} \begin{array}{l} 2 \\ 0 \end{array} \begin{array}{l} 1 \end{array}$ $64 = 1000000$

Hex: $64 \begin{array}{l} 16 \\ 0 \end{array} \begin{array}{l} 4 \end{array}$ $64 = 40$

C) 121

Binário:

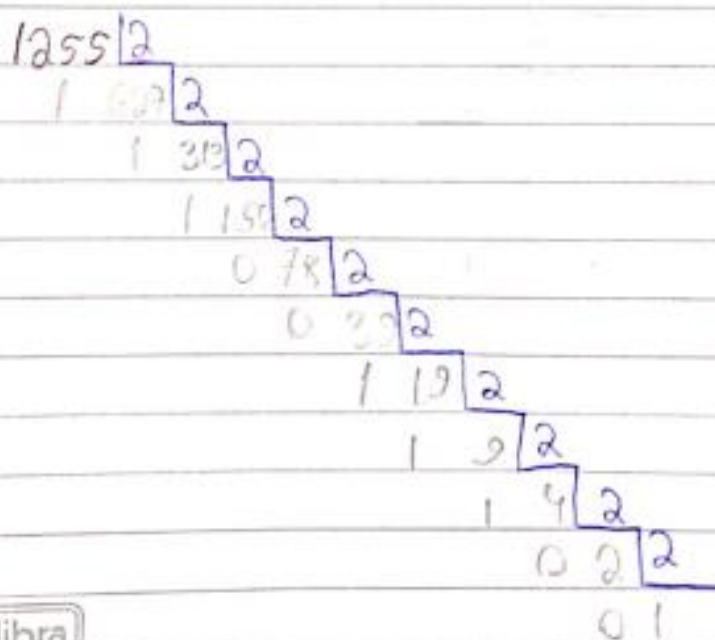


Hexa: 121 | 16 121 = 79

9 7

D) 1255

Binário : 1255 = 10011100111



Hexadecimal: $1255 = 4E7$

$$\begin{array}{r|l} 1255 & 16 \\ \hline 7 & 78 \\ 14 & 4 \end{array}$$

e) 512

Binário: $512 = 1000000000$

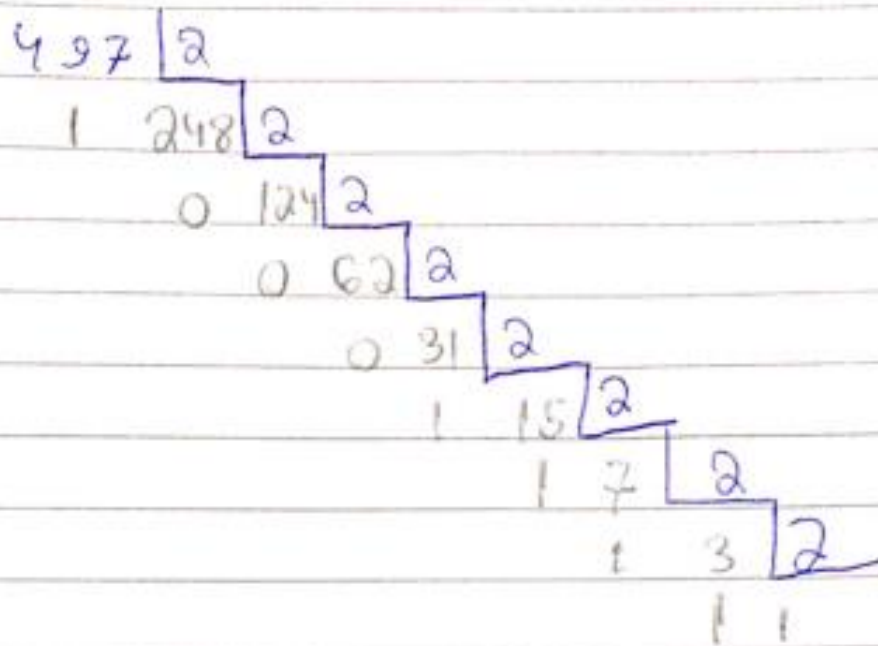
$$\begin{array}{r} 512 \div 2 \\ 0 \quad 256 \div 2 \\ 0 \quad 128 \div 2 \\ 0 \quad 64 \div 2 \\ 0 \quad 32 \div 2 \\ 0 \quad 16 \div 2 \\ 0 \quad 8 \div 2 \\ 0 \quad 4 \div 2 \\ 0 \quad 2 \div 2 \\ 0 \quad 1 \end{array}$$

Decimal: $512 = 200$

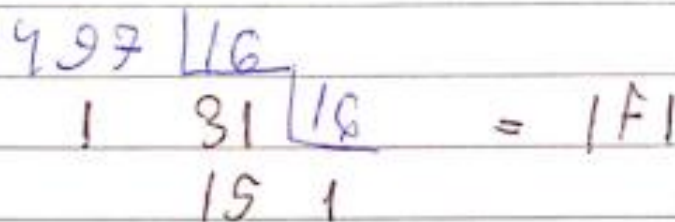
$$\begin{array}{r|l} 512 & 16 \\ \hline 0 & 32 \\ 0 & 2 \end{array} = 200$$

9) 497

Binário: $497 = 111110001$



Hex: $497 = 1F1$



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 : Hex = 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: 101010111001101

^{3 2 1 0}
D) 1204: Decimal: 4612
Binário: 1001011000100

^{3 2 1 0}
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

* Não houve modo, estão dentro dos limites

C) 123 e 100: $123 = 01111011$

$100 = 01100100$

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.

3) Considere os números decimais apresentados nas linhas 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. Use operações de 8 bits.

A) 36 - 40

$$\begin{array}{r} 36 \rightarrow 00100100 \quad | \quad 11011011 \\ 40 \rightarrow 00101000 \quad | \quad \underline{+1} \\ \hline 11011100 \end{array}$$

$$\begin{array}{r} 11011100 \quad | \quad -36 \\ \underline{00101000} \quad | \quad -40 \\ \hline 10000100 \rightarrow -4 \end{array}$$

→ Carry

B) 20 - 20

$$\begin{array}{r} 20 \rightarrow 00010100 \rightarrow 11010111 \\ \hline 00010100 \quad \left\{ \begin{array}{l} 20 \\ -20 \\ 0 \end{array} \right. \quad \begin{array}{r} 11010111 \\ \underline{+1} \\ 1101100 \end{array} \end{array}$$

$$\begin{array}{r} 1101100 \end{array}$$

→ Carry

C) $123 - 100$

$$\begin{cases} 123 \rightarrow 01111011 \\ 100 \rightarrow 01100100 \rightarrow 10011011 \\ \quad \quad \quad + 1 \\ \hline 10011100 \end{cases}$$

$$\begin{array}{r} 01111011 \quad \left\{ \begin{array}{l} 123 \\ -100 \\ \hline 0001011 \end{array} \right. \rightarrow 23 \\ \text{Carry} \end{array}$$

D) $240 - 204$

$$\begin{cases} 240 \rightarrow 11110000 \\ 204 \rightarrow 11001100 \rightarrow 00110011 \\ \quad \quad \quad + 1 \\ \hline 00110100 \end{cases}$$

$$\begin{array}{r} 11110000 \quad \left\{ \begin{array}{l} 240 \\ -204 \\ \hline 10010000 \end{array} \right. \rightarrow 36 \\ \text{Carry} \end{array}$$

Arquitetura de computadores (Lista 3)

* Aula 3 Arquitetura de computadores

Exercícios:

1) Converta os números de binário para decimal:

$$a) 10011 = 1 \times 2^4 + 1 \times 2^1 + 1 \times 2^0 = 19$$

$$b) 111011.101 = 111011 \cdot 101$$

$$111011 \rightarrow (1 \times 2^5 + 1 \times 2^4 + 1 \times 2^3 + 1 \times 2^1 + 1 \times 2^0) = 59$$

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

$$R: 111011.101 = 59.625$$

$$c) 111.11001 = 111 \cdot 11001$$

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

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

$$R: 7.78125$$

2) Demonstre o complemento de 2 dos números:

A) 123 (8 bits)

$$123 \mid 2 \quad 123 \rightarrow 01111011$$

$$1 \ 61 \ 2 \quad 10000100$$

$$1 \ 30 \ 2 \quad + 1$$

$$0 \ 15 \ 2 \quad 10000101$$

$$1 \ 7 \ 2$$

$$1 \ 3 \ 2$$

$$1 \ 2 \ 2$$

1

B) 1 (8 bits)

$$\begin{array}{r}
 1 = 00000001 \\
 \quad 11111110 \\
 \quad \quad + 1 \\
 \quad 11111111
 \end{array}$$

C) 118 (8 bits)

$$\begin{array}{r}
 118 = 01110110 \\
 \quad 10001001 \\
 \quad \quad + 1 \\
 \quad 10001010
 \end{array}$$

D) 2800 (16 bits)

$$\begin{array}{r}
 2800 = 0000101011100000 \\
 \quad 1111010100001111 \\
 \quad \quad + 1 \\
 \quad 11110101000010000
 \end{array}$$

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

A) $4 + 2$ (8 bits)

$$\begin{array}{r}
 4 \rightarrow 000000100 \} 4 \\
 2 \rightarrow 00000010 \} + 2 \\
 \quad 000000110 \} 6
 \end{array}$$

B) $120 + 8$ (8 bits)

$$\begin{array}{r}
 120 \rightarrow 01111000 \\
 8 \rightarrow 00001000 \\
 \hline
 10000000
 \end{array}
 \left.
 \begin{array}{l}
 \\
 + \\
 \\
 \end{array}
 \right\}
 \begin{array}{l}
 120 \\
 8 \\
 \hline
 128
 \end{array}$$

C) $120 - 5$ (8 bits)

$$\begin{array}{r}
 120 \rightarrow 01111000 \\
 5 \rightarrow 00000101 \rightarrow 1111010 \\
 \hline
 1111011
 \end{array}$$

$$\begin{array}{r}
 01111000 \\
 1111011 \\
 \hline
 10110011
 \end{array}
 \left.
 \begin{array}{l}
 \\
 -5 \\
 \\
 \end{array}
 \right\}
 \begin{array}{l}
 120 \\
 -5 \\
 \hline
 115
 \end{array}$$

\hookrightarrow Carry

D) $50 - 50$ (8 bits)

$$\begin{array}{r}
 50 \rightarrow 00110010 \rightarrow 11001101 \\
 \hline
 11001110
 \end{array}$$

$$\begin{array}{r}
 00110010 \\
 11001110 \\
 \hline
 10000000
 \end{array}
 \left.
 \begin{array}{l}
 50 \\
 -50 \\
 \\
 \end{array}
 \right\}
 \begin{array}{l}
 50 \\
 -50 \\
 \hline
 0
 \end{array}$$

\hookrightarrow Carry

e) 50 - 51 (8 bits)

50 \rightarrow 00110010

51 \rightarrow 00110011 \rightarrow 11001100
 $+ 1$
 11001101

00110010 } 50

11001101 } -51

11111111 } -1

\hookrightarrow Carry

f) 1000 - 500 (12 bits)

1000 \rightarrow 001111101000

500 \rightarrow 000111101000 \rightarrow 111000001011
 $+ 1$
 111000001100

001111101000 } 1000

111000001100 } -500

100011110100 500

\hookrightarrow Carry

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

A)

1	10111001	010110000000000000000000
---	----------	--------------------------

 \ominus \leftarrow $\frac{1}{4} + \frac{1}{16} + \frac{1}{32} \rightarrow 0,34375$
 $\rightarrow -1^5 (1, m) \cdot 2^E \rightarrow -1^5 \cdot (1,34375) 2^{58}$

R: $-3,8730957 \times 10^{12}$

B)

0	01111000	011000000000000000000000
---	----------	--------------------------

 \oplus $\frac{1}{4} + \frac{1}{8} \rightarrow 0,375$
 $\rightarrow -1^5 (1, m) \cdot 2^E \rightarrow 1^0 \cdot (1,375) \cdot 2^{-7} \rightarrow 0,0107421875$

R: $0,0107421875$