



Exercícios de Revisão

- Arquitetura de Computadores.** Caracterize a arquitetura de von Neumann: apresente seus módulos componentes e suas funcionalidades.
- Representação de dados.**
 - O que representa a tabela ASCII?
 - Qual o valor ASCII correspondente aos seguintes símbolos: 'A', 'B', 'a', 'b', '0' e '9' ?
 - Qual o valor da tabela ASCII nas posições 0, 8, 9 e 123?
 - Problema: caso você possua um byte com o código ASCII de um caracter que você saiba que codifica um dígito (ou seja, um “caracter” entre 0 e 9, não um número entre 0 e 9), como você poderia obter o valor numérico correspondente? Por exemplo, se o caracter em questão for o caracter '5', como obter o número 5 correspondente?
- Representação de dados.** Bases numéricas. Complete a tabela de conversão de bases numéricas.

<i>Decimal</i>	<i>Binário</i>	<i>Octal</i>	<i>Hexadecimal</i>
			139
		343	
	1000.0000.0000		
32010			

- Representação de dados.** Bases numéricas. Complete a tabela de conversão de bases numéricas.

	<i>Valor</i>	<i>E</i>	<i>Sinal</i>	<i>Expoente</i>	<i>Mantissa</i>
a)	1.0				
b)	-1.0				
c)	10.5				
d)	-10.5				
e)	0.25				
f)	-0.25				
g)	0.4375				
h)	4.375				

Onde:

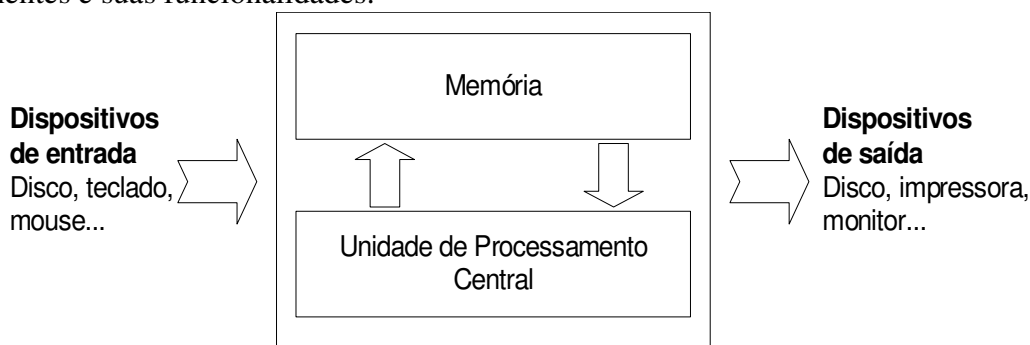
E : corresponde ao expoente utilizado na notação científica normalizada binária

Expoente : corresponde a representação binária de armazenado $127+E$ (pois $Expoente - 127 = E$)

Mantissa : corresponde ao valor de xxxx na mantissa binária normalizada para o formato $1,xxx * 2^{Expoente-127}$

Soluções

1. **Arquitetura de Computadores.** Caracterize a arquitetura de von Neumann: apresente seus módulos componentes e suas funcionalidades.



A arquitetura de von Neumann possui os módulos apresentados na figura acima. esta arquitetura é um modelo de computador que possui uma unidade de armazenamento única, a memória, onde são mantidas, simultaneamente, as instruções de um programa em execução e os dados que este programa manipula. A unidade central de processamento (CPU) é responsável pela execução das operações, provendo meios tanto para realizar o controle da execução (como o cadenciamento das instruções) como para efetivar as operações (tais as operações matemáticas requeridas) Dispositivos de entrada e saída permitem a comunicação deste sistema computacional com o meio externo.

2. Representação de dados.

2.1. O que representa a tabela ASCII?

A tabela ASCII (American Standard Code for Information Interchange) corresponde a norma de codificação de caracteres mais conhecida e utilizada em informática. Esta norma, na sua origem, utiliza 7 bits (números 0 a 127) para codificar, em linguagem binária, caracteres. Para aproveitar os 8 bits de um byte, normalmente é utilizada a versão estendida da tabela ASCII, permitindo a codificação de 256 caracteres. Deve-se notar que nem todos os “caracteres” codificados por esta tabela podem ser “visíveis”. Alguns destes caracteres possuem funções específicas, por exemplo, o código 0_{10} representa o caracter NULL, o código 13_{10} corresponde à ENTER e o código 27_{10} a ESCAPE.

2.2. Qual o valor ASCII correspondente aos seguintes símbolos: 'A', 'B', 'a', 'b', '0' e '9' ?

Tanto na tabela ASCII original como na estendida estes caracteres encontram-se nas posições: 65, 66, 97, 98, 48 e 57.

2.3. Qual o valor da tabela ASCII nas posições 0, 8, 9 e 123?

Os valores nestas posições são os caracteres representando NULL, BACKSPACE, TABULAÇÃO e {

2.4. Problema: caso você possua um byte com o código ASCII de um caracter que você saiba que codifica um dígito (ou seja, um “caracter” entre 0 e 9, não um número entre 0 e 9), como você poderia obter o valor numérico correspondente? Por exemplo, se o caracter em questão for o caracter '5', como obter o número 5 correspondente?

Basta realizar a subtração do caracter em questão do caracter '0'. Em outras palavras, subtrair o código ASCII do dígito '0' do código ASCII desejado.

3. Representação de dados. Bases numéricas. Complete a tabela de conversão de bases numéricas.

<i>Decimal</i>	<i>Binário</i>	<i>Octal</i>	<i>Hexadecimal</i>
313	0001 0011 1001	471	139
227	1110 0011	343	E3
2048	1000 0000 0000	4000	800
32010	111 1101 0000 1010	76412	7D0A



4. **Representação de dados.** Bases numéricas. Complete a tabela de conversão de bases numéricas.

	<i>Valor</i>	<i>E</i>	<i>Sinal</i>	<i>Expoente</i>	<i>Mantissa</i>
a)	1.0	0	0	01111111	000000000000000000000000
b)	-1.0	0	1	01111111	000000000000000000000000
c)	10.5	3	0	10000010	010100000000000000000000
d)	-10.5	3	0	10000010	010100000000000000000000
e)	0.25	-2	0	01111101	000000000000000000000000
f)	-0.25	-2	0	01111101	000000000000000000000000
g)	0.4375	-2	0	01111101	110000000000000000000000
h)	4.375	2	0	10000001	000110000000000000000000

Exemplos de resolução

- a) Decimal: 1.0 Binário: 1.0 Binário normalizado: 1.0×2^0
Bit de Sinal: 0 (o número é positivo)
Expoente em excesso: $0 + 127 = -127 = 01111111$ em binário
Mantissa: 1.000... Como o valor 1 a esquerda do ponto é subentendido, é armazenado 32 zeros
Mantissa armazenada: 23 zeros (o valor 1 a esquerda do ponto é subentendido)
- b) Decimal: -1.0 Binário: -1.0 Binário normalizado: -1.0×2^0
Bit de Sinal: 1 (o número é negativo)
Expoente em excesso: $0 + 127 = -127 = 01111111$ em binário
Mantissa: 1.000... Como o valor 1 a esquerda do ponto é subentendido, é armazenado 32 zeros
Mantissa armazenada: 23 zeros (o valor 1 a esquerda do ponto é subentendido)
- c) Decimal: 10.5 Binário: 1010.1 Binário normalizado: 1.0101×2^3
Bit de Sinal: 0 (o número é positivo)
Expoente em excesso: $3 + 127 = 130$ decimal ou 10000010 em binário
Mantissa: 1.01010... Completar com zeros até atingir 23 bits
Mantissa armazenada: 01010 (completar os bits restantes com zeros)
- f) Decimal: -0.25 Binário: 0.01 Binário normalizado: 1.0×2^{-2}
Bit de Sinal: 1 (o número é negativo)
Expoente em excesso: $2 + 127 = 129$ decimal ou 01111101 em binário
Mantissa: 1.00... Completar com zeros até atingir 23 bits
Mantissa armazenada: 23 zeros (o valor 1 a esquerda do ponto é subentendido)
- g) Decimal: 0.4375 Binário: 0.011101 Binário normalizado: 1.1101×2^{-2}
Bit de Sinal: 1 (o número é negativo)
Expoente em excesso: $-2 + 127 = 125$ decimal ou 01111101 em binário
Mantissa: 1.1101... Completar com zeros até atingir 23 bits
Mantissa armazenada: 1101 (completar bits restantes com zeros)
- h) Decimal: 4.375 Binário: 100.011001 Binário normalizado: 1.00011001×2^2
Bit de Sinal: 1 (o número é negativo)
Expoente em excesso: $2 + 127 = 129$ decimal ou 10000001 em binário
Mantissa: 1.1101... Completar com zeros até atingir 23 bits
Mantissa armazenada: 1101 (completar bits restantes com zeros)