Utilizando 2 bits podemos representar 4 diferentes valores, como apresentado na tabela abaixo:

|  |  |  |  |
| --- | --- | --- | --- |
| 2-Bits | |  | Valor em  Decimal |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 2 |
| 1 | 1 | 3 |

1. Se forem utilizados 3 e 4 bits teremos, respectivamente, 8 e 16 valores. Complete as tabelas com as sequencias de bits.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 3-Bits | | |  | Valor em  Decimal |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 2 |
| 0 | 1 | 1 | 3 |
| 1 | 0 | 0 | 4 |
| 1 | 0 | 1 | 5 |
| 1 | 1 | 0 | 6 |
| 1 | 1 | 1 | 7 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 4-Bits | | | |  | Valor em  Decimal |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 1 | 0 | 2 |
| 0 | 0 | 1 | 1 | 3 |
| 0 | 1 | 0 | 0 | 4 |
| 0 | 1 | 0 | 1 | 5 |
| 0 | 1 | 1 | 0 | 6 |
| 0 | 1 | 1 | 1 | 7 |
| 1 | 0 | 0 | 0 | 8 |
| 1 | 0 | 0 | 1 | 9 |
| 1 | 0 | 1 | 0 | 10 |
| 1 | 0 | 1 | 1 | 11 |
| 1 | 1 | 0 | 0 | 12 |
| 1 | 1 | 0 | 1 | 13 |
| 1 | 1 | 1 | 0 | 14 |
| 1 | 1 | 1 | 1 | 15 |

1. Quantos valores podemos representar para cada um dos casos abaixo? Qual a regra que permite saber o número de valores representáveis? Todos os números são escritos a partir de potências do número 2, potencializado pela quantidade da sequência de bits.
   1. 8 bits: 256
   2. 32 bits: 4.294.967.295
   3. 64 bits: 18.446.744.073.709.551.615

Regra geral: 2^Tam. De Bits

1. Qual o valor decimal representado para os valores abaixo, assumindo representação de número positivos apenas?

a) 101010 = 42

b) 111000 = 56

c) 10001000 = 136

d) 10000001 = 129

1. Seja uma arquitetura de 7 bits, quais as faixas de valores inteiros representáveis assumindo que os números são representados em:
   1. Binário sem sinal [0; 127]
   2. Sinal magnitude [-63; 63]
   3. Complemento de dois [-64; 63]
2. Realize as operações abaixo, representados em complemento de 2 em uma arquitetura de 5 bits. Sinalize os casos de overflow.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1. 11101 | + | 01001 = | 6. 10101 | - 01111 = |
| 2. 10101 | + | 11011 = | 7. 10011 | - 11101 = |
| 3. 10001 | + | 00111 = | 8. 10010 | - 00101 = |
| 4. 01101 | + | 01010 = | 9. 11101 | - 01110 = |
| 5. 01111 | + | 11111 = | 10. 01111 | - 11111 = |

1. Em cada uma dessas representações (Binário Sem Sinal, Sinal Magnitude e Complemento de Dois), mostre como seriam representados os números (em uma arquitetura de 7 bits):

Respectivamente: binário sem sinal, sinal magnitude e complemento de dois

a) a. 100 0110010, 0110010, 0110010

b) b. -49 Não existe, 1110001, 100111

c) c. Zero 0000000, 0000000 ou 10000000, 0000000

1. Um computador possui uma memória máxima de 32GB. Quantos bits são necessários para que se possa endereçar esta memória (ou seja, quantos bits são necessários para podermos numerar cada byte da memória)? Explique sua resposta.

Cada byte equivale a 8 bits, deve-se considerar também que 1GB são 1 bilhão de bytes que equivale a 8 bilhões de bits. Agora, sabe-se que a memória do computador possui 32 bilhões de bytes e, multiplicado por 8, saberemos quantos bits serão necessários.

32.000.000.000 x 8 = 256.000.000.000

Serão necessários 256 bilhões de bits para numerar cada byte da memória.

1. Pesquise e explique os conceitos abaixo:
2. O que são as tabelas ASCII e Unicode? Para que servem? Como estão organizadas? O que tem em comum e o que as diferencia?

**ASCII** – É o Código de Padrão Americano para o Intercâmbio de Informação, sua função é padronizar a forma de como os computadores representam letras, números, acentos, sinais diversos e alguns códigos de controle. É baseado no alfabeto romano e é possível representar 256 caracteres diferentes atualmente.

**Unicode** – O Unicode fornece um único conjunto de caracteres contendo os idiomas de todo o mundo e um número pequeno de formatos e esquemas de codificação conhecidos da máquina para adequar as necessidades de aplicativos e protocolos existentes. Possui formas de codificação como UTF-8, UTF-16 e UTF-32, a tabela define códigos para os caracteres de todas as linguagens mais utilizadas do mundo, suportando mais de 100.000 caracteres diferentes atualmente.

A tabela Unicode foi projetada com base no código ASCII, tanto que o encoding mais usado da tabela Unicode, UTF-8, é 100% compatível com a tabela ASCII. A principal diferença é o tamanho, o Unicode, como foi retratado, representa muito mais caracteres do que ASCII.

1. O que são codificações UFT-8 e UTF-16? Como se relacionam com os conceitos anteriores?

**UTF-8** – É a codificação de caracteres mais comum da World Wide Web. Cada caractere é representado por um a quatro bytes. UTF-8 é compatível com versões anteriores do ASCII e pode representar qualquer caractere Unicode padrão. É muito utilizado com HTML, totalmente compatível com ASCII e possui tamanho variável de bytes.

**UTF-16** - É uma codificação de caracteres capaz de codificar todos os 1.112.064 pontos de código não substitutos de Unicode. A codificação é de comprimento variável , pois os pontos de código são codificados com uma ou duas unidades de código de 16 bits . O UTF-16 surgiu de uma codificação anterior de 16 bits de largura fixa conhecida como UCS-2 uma vez que ficou claro que mais de 65.536 pontos de código eram necessários.

O UTF-8 é o encoding mais usado da tabela Unicode e é compatível com o ASCII, enquanto o UTF-16 é a única codificação da web incompatível com o ASCII.

1. Pesquise e explique como são representadas imagens no computador. Como são representados os elementos das imagens? O que são pixels? O que são formatos de arquivos? Exemplifique.

As imagens são representadas apenas pela combinação de 3 cores base: vermelho, verde e azul. Este sistema é chamado de RGB (do inglês red, green e blue) e podemos colocar uma porcentagem de cada tonalidade para formar uma cor. Na computação, imagem é uma matriz de pixels de várias dimensões. Se ampliarmos bem uma imagem vamos ver diversos quadrinhos coloridos que são chamados de pixel, uma imagem é formada com diversos destes pontos. Um pixel é a menor unidade de uma imagem digital que pode ser exibida em uma tela. Quando combinados, os pixels dão vida às imagens e telas que conhecemos hoje.

Formato de arquivo é o termo que se usa para identificar a forma que um arquivo é armazenado no computador, e reconhecido por ele. Cada programa existente no seu computador serve para abrir alguns formatos específicos. Há diversos arquivos, como o png, que é reconhecido como uma fotografia, ou o pdf que geralmente é um documento de texto que não é permitida alteração, entre outros como jpg, txt, etc...

1. 4. Realize as conversões de base indicadas:

a. 7610 → (1312322)4

b. 1245 → (1245)10

c. 101010 → (2A)16

d. 1AA16 → ()10

e. 7010 → (1101101100010)2