

1) Qual a diferença entre organização básica e arquitetura?

A arquitetura é toda a parte em que é visível e interagível, como as instruções informadas a máquina, os números de bits utilizados, mecanismos para interação com a máquina a partir de entrada e saída de dados, métodos e técnicas desenvolvidas para manipulação dos endereçamentos de dados.

Já a organização básica se dá pelas características do circuito lógico, como os sinais de controle, interfaces e todo o hardware para o funcionamento da máquina.

2) Qual a principal vantagem em usar computadores da mesma família de arquitetura?

A compatibilidade dos processadores tem uma vantagem por conta dos aplicativos processados nele, que contêm códigos que funcionam somente com computadores ou processadores da mesma arquitetura.

3) Quais as funções básicas de um computador?

- Processar dados.
- Armazenar dados.
- Transferir ou alocar dados.
- Controlar a entrada e saída de dados.

4) Quais os passos que uma movimentação de dados segue para ser realizada?

Primeiramente temos a entrada dos dados ou transfêrencia dos dados para o controlador, que irá gerenciar o fluxo das informações entre o armazenamento e o processamento se necessário.

5) Quais os passos que uma operação de processamento segue para ser realizada?

Para a operação de processamento de dados, primeiramente temos o controlador acessando a memória de armazenamento e transfêrindo-o para o registrador do processador, dentro do processador tem uma unidade de controle para gerenciar a comunicação entre os registradores e a unidade lógica e aritmética

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

A base decimal tem como base o número 10 sendo por tanto 10 algarismos diferentes o representando, sendo o 0, 1, 2, 3, 4, 5, 6, 7, 8 e o 9, enquanto que no

hexadecimal tem-se a base o número 16 sendo por tanto 16 algarismos diferentes, sendo o 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E e F.

Como por exemplo o número 120 convertido em hexadecimal ficará 78.

7) Converta para Binário:

$$126 = 1111110$$

$$333 = 101001101$$

$$991 = 1111100111$$

8) Converta para decimal:

$$1000001 = 65$$

$$1111000 = 120$$

$$10000 = 16$$

1) Converta os números de binário para decimal:

- a) $10011 = 19$
- b) $111011 = 59$
- c) $11101111 = 239$
- d) $011011101111 = 1775$

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

A melhor forma de representação de números inteiros seria o complemento de dois, pois permite a representação dos valores sem um bit de sinal positivo ou negativo, além de evitar problemas com overflow e underflow associados ao bit de sinal.

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

a) 12 bits 2^{12}

Não sinalizado 0 a 4.095

Sinalizado -2.048 a $+2.047$

b) 20 bits 2^{20}

Não sinalizado 0 a 1.048.575

Sinalizado -524.288 a $+524.287$

c) 24 bits 2^{24}

Não sinalizado 0 a 16.777.215

Sinalizado $-8.388.608$ a $+8.388.607$

4) Demonstre os complemento de dois dos números:

a) 23 (para 8 bits)

23 (dec) = $\begin{array}{r} 00010111 \\ 11101000 \\ 1 \\ 11101001 \end{array}$

b) 127 (para 8 bits)

127 (dec) = $\begin{array}{r} 01111111 \\ 10000000 \\ 1 \\ 10000001 \end{array}$

c) 0 (para 8 bits)

0 (dec) = $\begin{array}{r} 00000000 \\ 11111111 \\ 1 \\ 00000000 \end{array}$

d) 128 (para 8 bits)

128 (dec) = $\begin{array}{r} 10000000 \\ 01111111 \\ 1 \\ 10000000 \end{array}$

e) 3000 (para 16 bits)

3000 (dec) = 0000101110111000
 1111010001000111
 1
 1111010001001000

5) Converta os números da base decimal para: hexa e binário.

- a) 10 (dec) = a (hex) = 1010 (bin)
- b) 64 (dec) = 40 (hex) = 1000000 (bin)
- c) 121 (dec) = 79 (hex) = 1111001 (bin)
- d) 1255 (dec) = 4e7 (hex) = 10011100111 (bin)
- e) 512 (dec) = 200 (hex) = 100000000 (bin)
- f) 497 (dec) = 1f1 (hex) = 111110001 (bin)

6) Converta os números da base hexadecimal para decimal e binário.

- a) 36 (hex) = 54 (dec) = 110110 (bin)
- b) 2000 (hex) = 8.192 (dec) = 10000000000000 (bin)
- c) ABCD (hex) = 43.981 (dec) = 1010101111001101 (bin)
- d) 1204 (hex) = 4.612 (dec) = 1001000000100 (bin)
- e) 3333 (hex) = 13.107 (dec) = 11001100110011 (bin)

7) 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 soma em binário e indique carry e overflow. Usar operações em 8 bits.

36 = 00100100
40 = 00101000
 01001100 (carry)

20 = 00010100
20 = 00010100
 00101000 (carry)

123 = 01111011
100 = 01100100
 11011111 (carry)

240 = 11110000
204 = 11001100
 1 10111100 (overflow)

8) Considere o números decimal 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 carry e overflow. Usar operações em 8 bits.

36 = 00100100
40 = 00101000
11010111 (complemento de dois)
1
11011000
00100100
11111100 = -4 (dec)

20 = 00010100
20 = 00010100
11101011 (complemento de dois)
1
11101100
00010100
1 00000000 (overflow e carry) = 0 (dec)

123 = 01111011
100 = 01100100
10011011 (complemento de dois)
1
10011100
01111011
1 00010111 (overflow e carry) = 23 (dec)

240 = 11110000
204 = 11001100
00110011 (complemento de dois)
1
00110100
11110000
1 00100100 (overflow e carry) = 36 (dec)

1) Converta os números de binário para decimal:

- a) 10011 (bin) = 19 (dec)
- b) 111011.101 (bin) = 59.625 (dec)
- c) 111.11001 (bin) = 7.78125 (dec)

2) Demonstre os complemento de dois dos números:

- a) 123 (para 8 bits);
- b) 1 (para 8 bits);
- c) 118 (para 8 bits);
- d) 2800 (para 16 bits).

123 (dec) = 01111011 (bin)
 10000100
 1
 10000101 = ~123 (dec)

1 (dec) = 00000001 (bin)
 11111110
 1
 11111111 = ~1 (dec)

118 (dec) = 01110110 (bin)
 10001001
 1
 10001010 = ~118 (dec)

2800 (dec) = 0000101011110000 (bin)
 1111010100001111
 1
 1111 0101 0001 0000 = -2800 (dec)

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

- a) 4 + 2 (8 bits);
- b) 120 + 8 (8 bits);
- c) 120 - 5 (8 bits);
- d) 50 - 50 (8 bits);
- e) 50 - 51 (8 bits);
- f) 1000 - 500 (12 bits).

4 (dec) = 00000100
2 (dec) = 00000010
 00000110

120 (dec) = 01111000
8 (dec) = 00001000
 10000000

120 (dec) = 01111000
5 (dec) = 00000101

```

      11111010
        1
      11111011
      01111000
1 01010011 (overflow)
50 (dec) = 00110010
50 (dec) = 00110010
      11001101
        1
      11001110
      00110010
1 00000000 (overflow)

50 (dec) = 00110010
51 (dec) = 00110011
      11001100
        1
      11001101
      00110010
      11111111

1000 (dec) = 001111101000
500 (dec) = 000111110100
      111000001011
        1
      111000001100
      001111101000
1 000111110100 (overflow)

```

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

a) 1 10111001 010110000000000000000000

-1.193265... * 10^{-21}

b) 0 01111000 011000000000000000000000

+1.079597 * 10^{36}