Aula 00 - (Prof. Evandro)

TJ-RO (Analista Judiciário - Analista de Sistemas) Arquitetura de Computadores - 2024 (Pós-Edital)

Autor:
Equipe Informática e TI, Evandro Dalla Vecchia Pereira

31 de Outubro de 2024

Índice

1) Apresentação do Curso - Evandro 3
2) Apresentação Flashcards 6
3) Sistemas de Numeração - Teoria 8
4) Sistemas de Numeração - Questões Comentadas 14
5) Sistemas de Numeração - Lista de Questões 22
6) Aritmética Binária - Teoria 26
7) Aritmética Binária - Questões Comentadas 32
8) Aritmética Binária - Lista de Questões 38
9) Representação dos Dados - Teoria 42
10) Representação dos Dados - Questões Comentadas 52
11) Representação dos Dados - Lista de Questões 56
12) Proposições x Tabela Verdade - Teoria 59
13) Proposições x Tabela Verdade - Questões Comentadas 68
14) Proposições x Tabela Verdade - Lista de Questões 81
15) Álgebra Booleana e Circuitos Lógicos - Teoria 88
16) Álgebra Booleana e Circuitos Lógicos - Questões Comentadas 96
17) Álgebra Booleana e Circuitos Lógicos - Lista de Questões 103

APRESENTAÇÃO DO CURSO

Iniciamos nosso **Curso Regular de Sistemas Operacionais e Arquitetura de Computadores** em teoria e questões, voltado para provas **objetivas e discursivas** de concurso público. Tais assuntos são cobrados em diversos concursos em que há vagas específicas para a área de TI.

As aulas em PDF possuem por característica essencial a **didática**. Ao contrário do que encontramos em alguns livros, o curso todo se desenvolverá com uma leitura de fácil compreensão e assimilação.

Além disso, teremos videoaulas! Essas aulas destinam-se a complementar a preparação. Quando estiver cansado do estudo ativo (leitura e resolução de questões) ou até mesmo para a revisão, abordaremos alguns pontos da matéria por intermédio dos vídeos. Com outra didática, você disporá de um conteúdo complementar para a sua preparação. Ao contrário do PDF, evidentemente, AS VIDEOAULAS NÃO ATENDEM A TODOS OS PONTOS QUE VAMOS ANALISAR NOS PDFS, NOSSOS MANUAIS ELETRÔNICOS. Por vezes, haverá aulas com vários vídeos; outras que terão videoaulas apenas em parte do conteúdo. Nosso **foco** é sempre o **estudo ativo!**

APRESENTAÇÃO PESSOAL

Meu nome é Evandro Dalla Vecchia Pereira, sou autor do livro "Perícia Digital - Da investigação à análise forense", Mestre em Ciência da Computação (UFRGS), Bacharel em Ciência da Computação (PUCRS), Técnico em Redes de Computadores (Etcom/UFRGS) e em Processamento de Dados (Urcamp). Perito Criminal na área de Perícia Digital desde 2004 no Instituto-Geral de Perícias/RS. Professor de pós-graduação em diversas instituições, nas áreas de Perícia Digital, Perícia Criminal e Auditoria de Sistemas. Lecionei em cursos de graduação de 2006 a 2017, nas instituições PUCRS, Unisinos, entre outras e sou professor em cursos de formação e aperfeiçoamento de Peritos Criminais, Delegados, Inspetores, Escrivães e Policiais Militares.

No Estratégia Concursos leciono desde o começo de 2018, inicialmente na área de Computação Forense e, na sequência, também assumi as áreas de Arquitetura de Computadores e Sistemas Operacionais, tanto na elaboração de materiais escritos como na gravação das videoaulas.

Deixarei abaixo meus contatos para quaisquer dúvidas ou sugestões. Terei o prazer em orientá-los da melhor forma possível nessa caminhada que estamos iniciando.

Instagram: @profevandrodallavecchia

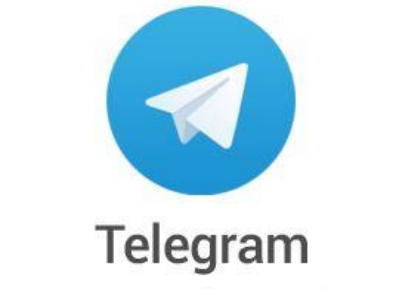
Facebook: https://www.facebook.com/profevandrodallavecchia

PARE TUDO! E PRESTE ATENÇÃO!!

Hoje eu faço parte de uma equipe **SENSACIONAL** de professores! Depois de muita luta conseguimos reunir **um time** de profissionais extremamente QUALIFICADO e sobretudo **COMPROMISSADO** em fazer o melhor pelos alunos. Para tal criamos um conjunto de ações para nos aproximarmos dos alunos, entendermos suas necessidades e evoluirmos nosso material para um patamar ainda mais diferenciado. São 3 as novidades que gostaria de convidá-lo a conhecer:

Nosso podcast alternativo ... livre, descontraído e com dicas rápidas que todo CANETA PRETA raiz deve ouvir. Já temos alguns episódios disponíveis e vários outros serão gravados nas próximas semanas ... acompanhe em:

http://anchor.fm/estrategia-tech



Nosso grupo do Telegram é um local onde ouvimos os alunos e trocamos ideias com eles. Está crescendo a cada dia. A regra do grupo é: só vale falar sobre concursos. Lá divulgamos nossas aulas ao vivo e falamos sobre os concursos abertos, expectativas de novos concursos, revisões de véspera, e por aí vai...

http://t.me/estrategia_ti

Criamos um perfil no Instagram ... e qual o objetivo? Fazer com que os alunos percam tempo nas redes sociais? Claro que não!! Estamos consolidando diversos posts dos professores! São dicas especiais, um patrimônio que deve ser explorado por todos os concurseiros de TI!

http://instagram.com/estrategiaconcursosti

ESTRATÉGIA FLASHCARDS

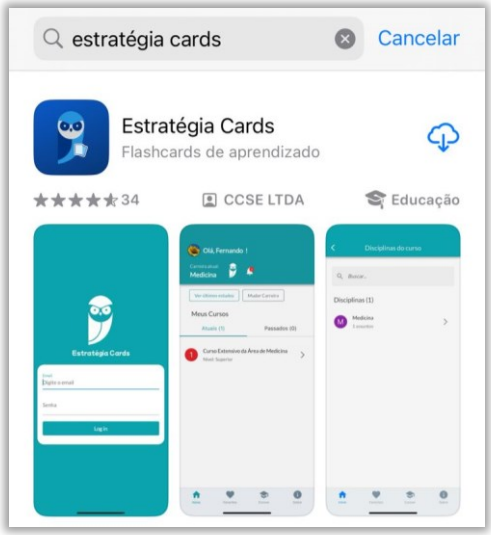
📚 Você tem dificuldade de estudar, memorizar e revisar os conteúdos que estuda em nossas aulas? Então nós temos a ferramenta perfeita para você!

Apresentamos o **Estratégia Cards**: app de flashcards que vai revolucionar sua forma de **estudar** e **revisar** conteúdos de provas de concurso público. Com nossa tecnologia inovadora e interface amigável, você dominará os tópicos mais complexos de maneira eficiente e divertida.

✨ Recursos do Estratégia Cards:

Curadoria de Flashcards	Flashcards criados e revisados por professores especializados em cada área, com qualidade e voltados para concursos públicos.
Flashcards Personalizados	Crie seus próprios flashcards, cobrindo os principais tópicos e matérias dos concursos públicos.
Repetição Espaçada	Técnica de aprendizagem que envolve revisar informações em intervalos crescentes para melhorar a retenção de longo prazo e combater o esquecimento.
Estatísticas Personalizadas	Visualize graficamente o percentual de acertos, erros ou dúvidas dos decks estudados.
Modo Offline	Estude em qualquer lugar, mesmo sem conexão à internet, fazendo o download dos decks.
Estudo por Áudio	*Está dirigindo ou fazendo esteira e quer continuar estudando?* Basta utilizar a opção "Escutar".
Decks Favoritos	Você pode escolher decks específicos como favoritos e visualizá-los em uma aba separada do app.
Opções de Estudo	Você poderá estudar todos os cards de um deck; ou apenas os que você errou; ou apenas os que você não estudou ainda; entre outras opções.

E como eu consigo baixar?



É muito fácil! Basta pesquisar por "Estratégia Cards" na loja oficial do seu smartphone.

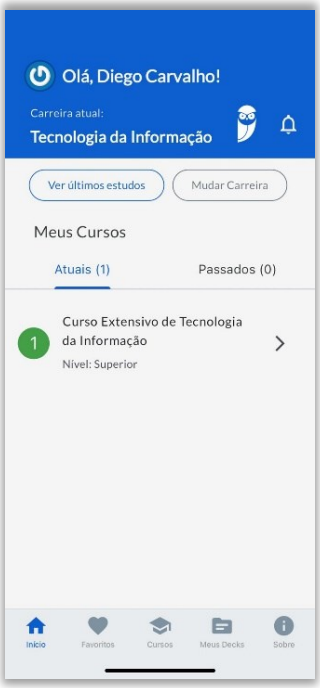
Se você tiver um Android, basta acessar a **Google Play**;

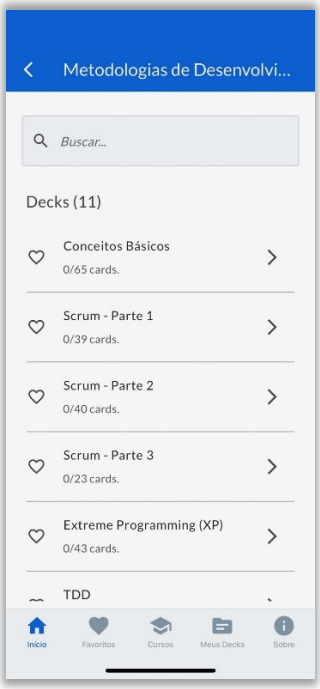
Se for tiver um iPhone, basta acessar a **App Store (iOS)**.

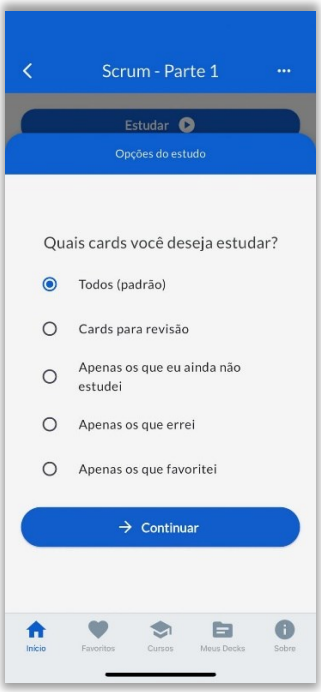
E para acessar?

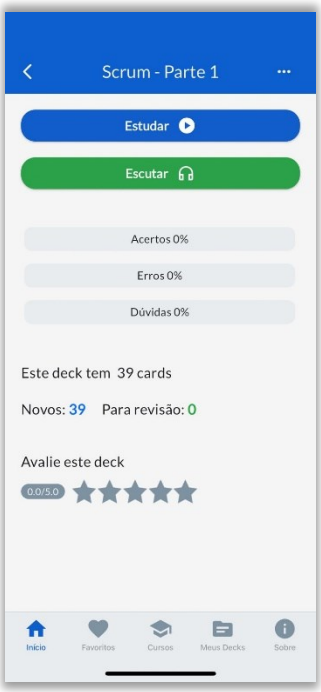
Para acessar, basta ter uma conta no Estratégia Concursos. Em seguida, utilize suas credenciais de login e senha para acessar o aplicativo. Por fim, acessa a carreira de Tecnologia da Informação.

Como utilizar o app:









SISTEMAS DE NUMERAÇÃO

No nosso dia a dia estamos acostumados a utilizar dígitos de 0 a 9, o sistema decimal. Por exemplo, se falamos do número 95 estamos falando de 9 dezenas + 5 unidades:

$$95 = (9 \times 10) + 5$$

Vamos ver um número maior...5893: significa 5 milhares, 8 centenas, 9 dezenas e 3 unidades:

$$5893 = (5 \times 1000) + (8 \times 100) + (9 \times 10) + 3$$

Podemos ver que o sistema decimal tem a base 10. Isso quer dizer que cada dígito no número é multiplicado por 10, elevado a uma potência que corresponde à posição do dígito. Veja:

$$95 = (9 \times 10^1) + (5 \times 10^0)$$

$$5893 = (5 \times 10^3) + (8 \times 10^2) + (9 \times 10^1) + (3 \times 10^0)$$

O mesmo princípio é utilizado para as frações decimais, porém com potências negativas de 10. Por exemplo, a fração decimal 0,368 pode ser representada assim:

$$0{,}368 = (3 \times 10^{-1}) + (6 \times 10^{-2}) + (8 \times 10^{-3})$$

O dígito mais à esquerda é conhecido como dígito mais significativo (valor mais alto) e o mais à direita é o dígito menos significativo (valor mais baixo).

O dígito mais

à esquerda | à direita

mais significativo | menos significativo

Bom, sabemos que a base decimal funciona bem para os seres humanos, mas as máquinas trabalham apenas com bits (0 e 1), ou seja, um sistema binário (base 2). Então devemos ter em mente que jamais haverá o dígito 2, 3, 4, e assim por diante. Vamos ver os primeiros dez dígitos binários: 0, 1, 10, 11, 100, 101, 110, 111, 1000, 1001, 1010. Vejamos o último dessa sequência (1110):

$$(1 \times 2^3) + (1 \times 2^2) + (1 \times 2^1) + (0 \times 2^0)$$

$$8 + 4 + 2 + 0 = 14$$

Pronto! Acabamos de converter o valor 1110_2 em 14_{10} (o número subscrito significa a base, então convertemos 1110 no sistema binário para 14 no sistema decimal). Esse tipo de conversão é bastante cobrado em provas de concurso! Aconselho que você pratique bastante!

E se eu quiser converter um número no sistema decimal para o binário? Temos a seguinte regra:

1. Divide o valor decimal por 2 (o resto será 0 ou 1, não tem como sobrar algo maior que isso);
2. O primeiro resto ficará bem na direita (menos significativo), os seguintes ficarão mais à esquerda;
3. Continua a divisão até que chegue em 1 / 2 = 0 (resto 1).

Vamos converter 23_{10} para binário (dica - é um número ímpar, então deve terminar com 1):

	Quociente	Resto
23 / 2 =	11	1
11 / 2 =	5	1
5 / 2 =	2	1
2 / 2 =	1	0
1 / 2 =	0	1

Agora é só ler de baixo para ↑ cima o resto. Temos como resultado: 101112.

Um outro macete, que é muito útil também para quem estuda as máscaras de redes de computadores é preencher a seguinte tabela (da direita para a esquerda, começa em 1 e vai dobrando...2, 4, 8, 16, 32 etc.):

512	256	128	64	32	16	8	4	2	1
					1	0	1	1	1

Vamos preencher com 1 os valores que queremos somar. Como temos 23 em decimal, temos que começar preenchendo o valor menor que 23, que é 16. Se preenchêssemos o 8, teríamos 24 (16 + 8), então colocamos 0 e vamos para o próximo. Preenchemos o 4 (total = 20). Preenchemos o 2 (total = 22) e preenchemos o 1 (total = 23). Pronto! Agora você escolhe a maneira que acha mais fácil!

Uma outra notação bastante utilizada, principalmente quando queremos analisar alguma informação bruta (*dump* de memória ou uma partição de um disco em que não se identifica um sistema de arquivos, por exemplo), é a hexadecimal (dezesseis dígitos possíveis). Como os números vão de 0 a 9, os caracteres seguintes são A, B, C, D, E, F. Como são 16 valores possíveis, é possível representar 4 bits com um dígito em hexadecimal, afinal $2^4 = 16$.

```
0000000000 33 C0 8E D0 BC 00 7C 8E C0 8E D8 BE 00 7C BF 00 06 B9 00 02 FC F3 A4 50 68 1C 06 CB FB B9 04 00
0000000020 BD BE 07 80 7E 00 00 7C 0B 0F 85 0E 01 83 C5 10 E2 F1 CD 18 88 56 00 55 C6 46 11 05 C6 46 10 00
0000000040 B4 41 BB AA 55 CD 13 5D 72 0F 81 FB 55 AA 75 09 F7 C1 01 00 74 03 FE 46 10 66 60 80 7E 10 00 74
0000000060 26 66 68 00 00 00 00 66 FF 76 08 68 00 00 68 00 7C 68 01 00 68 10 00 B4 42 8A 56 00 8B F4 CD 13
0000000080 9F 83 C4 10 9E EB 14 B8 01 02 BB 00 7C 8A 56 00 8A 76 01 8A 4E 02 8A 6E 03 CD 13 66 61 73 1C FE
00000000a0 4E 11 75 0C 80 7E 00 80 0F 84 8A 00 B2 80 EB 84 55 32 E4 8A 56 00 CD 13 5D EB 9E 81 3E FE 7D 55
00000000c0 AA 75 6E FF 76 00 E8 8D 00 75 17 FA B0 D1 E6 64 E8 83 00 B0 DF E6 60 E8 7C 00 B0 FF E6 64 E8 75
00000000e0 00 FB B8 00 BB CD 1A 66 23 C0 75 3B 66 81 FB 54 43 50 41 75 32 81 F9 02 01 72 2C 66 68 07 BB 00
```

Vamos ver a tabela abaixo que fica mais fácil entender:

Binário	Decimal	Hexadecimal
0000	0	0
0001	1	1
0010	2	2
0011	3	3
0100	4	4
0101	5	5
0110	6	6
0111	7	7
1000	8	8
1001	9	9
1010	10	A
1011	11	B
1100	12	C
1101	13	D
1110	14	E
1111	15	F

Para não haver confusão, geralmente se representa hexadecimal da seguinte forma: 0xFA. E como converter de hexadecimal para binário? Ah, essa é mais fácil. É só lembrar da tabela que acabamos de ver (ou melhor, saber montar uma). Então, sabemos que F = 1111 e que A = 1010, então 0xFA = 11111010. É só concatenar!

E como converter de hexadecimal para decimal? Lembra das bases, que vimos lá no início dessa parte da aula? Então...sabemos que a base agora é 16, logo:

("F" x 16^1) + ("A" x 16^0) → coloquei as "letras" aqui apenas para começar, mas temos que substituir pelo valor de acordo com a sua posição (sabemos que A vem depois do 9, então é equivalente a 10, e o F é o último, ou seja o 15). Então fica assim:

(15 x 16^1) + (10 x 16^0)

240 + 10 = 250

Será que está certo? Vamos conferir com aquele macete que vimos antes (binário x decimal):

512	256	128	64	32	16	8	4	2	1
		1	1	1	1	1	0	1	0

128 + 64 + 32 + 16 + 8 + 2 = 250! Então vimos que funciona!

Por fim, vamos analisar o sistema octal, que, como o nome deixa claro, possui a representação de oito números, do 0 ao 7. Um dos usos do sistema octal é em alguns comandos no Linux. Por exemplo, o comando chmod 750 estrategia.txt atribui ao arquivo "estrategia.txt" as permissões "rwx r-x ---", ou seja, leitura/escrita/execução para o dono do arquivo, leitura/execução para o grupo e nenhuma permissão para os outros. Como $2^3 = 8$, então representamos um dígito na base octal com 3 bits. Vejamos:

Binário	Octal
000	0
001	1
010	2
011	3
100	4
101	5
110	6
111	7

E como converter de octal para decimal? Lembra das bases, que vimos lá no início dessa parte da aula? Então...sabemos que a base agora é 8. Vamos converter 372 para decimal:

(3 x 8^2) + (7 x 8^1) + (2 x 8^0)

192 + 56 + 2 = 250

Ok, mas e se eu tenho um valor em decimal e quero converter na base 8 (octal)? Bom, aí eu devo dividir por 8, até chegar ao resultado 0. Depois pego os restos de trás para frente, lembra? Vamos a um exemplo: converter 77 (base decimal) para a base octal:

	Quociente	Resto
77 / 8 =	9	5
9 / 8 =	1	1
1 / 8 =	0	1

Agora é só ler de baixo para ↑ cima o resto. Temos como resultado: 115.

1. (Quadrix/CRO-AC - 2019) São exemplos de bases utilizadas nos sistemas de computação a binária (base 2) e a octal (base 8).

Comentários:

Embora a octal seja pouco conhecida, é um exemplo, sim. Em comandos Linux temos um uso maior. A base 2 (binária) é a básica, pois representa o bit, a menor unidade de informação. Existe também a hexadecimal. A decimal é aquela que o ser humano utiliza. Portanto, a questão está correta.

Gabarito: Correta

2. (CESPE/IFF - 2018) O computador passou por diversas evoluções nos últimos anos; entretanto, continua utilizando a mesma lógica computacional e, basicamente, a mesma arquitetura. Nos computadores modernos, é empregada a lógica

A) decimal e a arquitetura Von Neumann.

B) binária e a arquitetura Claude E. Shannon.

C) binária e a arquitetura Von Neumann.

D) hexadecimal e a arquitetura Alan Turing.

E) decimal e a arquitetura Claude E. Shannon.

Comentários:

Sabemos que os computadores utilizam os bits (0 e 1), portanto utiliza uma "lógica" binária. E a arquitetura continua sendo a de von Neumann (em alguns componentes é utilizada a de Harvard, como por exemplo memória cache). Portanto, a alternativa C está correta e é o gabarito da questão.

Gabarito: Letra C

QUESTÕES COMENTADAS – SISTEMAS DE NUMERAÇÃO - MULTIBANCAS

1. (FGV/TJ-BA - 2015) O número binário 11111010 é representado na notação hexadecimal como:

A) F8

B) AF

C) FF

D) FA

E) FB

Comentários:

De binário para hexadecimal a maneira mais fácil é separar em grupos de 4 bits e verificar na tabela:

Binário	Decimal	Hexadecimal
0000	0	0
0001	1	1
0010	2	2
0011	3	3
0100	4	4
0101	5	5
0110	6	6
0111	7	7
1000	8	8
1001	9	9
1010	10	A
1011	11	B
1100	12	C

1101	13	D
1110	14	E
1111	15	F

Portanto, a alternativa D está correta e é o gabarito da questão.

Gabarito: Letra D

2. (FGV/IBGE - 2016) Analise as seguintes equações binárias:

111 × 11

111 - 11

110 ÷ 11

O resultado das equações apresentadas é, respectivamente:

A) 10101, 10 e 10;

B) 11101, 101 e 11;

C) 10111, 110 e 1;

D) 10101, 100 e 10;

E) 10100, 11 e 101.

Comentários:

Uma das maneiras é converter antes de aplicar o cálculo. Vamos ver a tabela (só metade dela):

Binário	Decimal	Hexadecimal
0000	0	0
0001	1	1
0010	2	2
0011	3	3
0100	4	4
0101	5	5
0110	6	6
0111	7	7

Convertendo para decimal, temos:

7 x 3 = 21 → 21 / 2= 10 (resto 1); 10 / 2 = 5 (resto 0); 5 / 2 = 2 (resto 1); 2 / 2 = 1 (resto 0); 1 / 2 = 0 (resto 1) → restos de trás para frente → 10101_2

7 – 3 = 4 → 100_2 (só olhar na tabela)

6 ÷ 3 = 2 → 10_2 (só olhar na tabela)

Note que apenas com o cálculo do meio já mataríamos a questão! Mas é bom fazer todos para treinar...no dia da prova não precisa! Apenas se tiver dúvida e se tiver tempo no fim da prova, para conferir.

Portanto, a alternativa D está correta e é o gabarito da questão.

Gabarito: Letra D

3. (CESPE/IFF - 2018) O computador passou por diversas evoluções nos últimos anos; entretanto, continua utilizando a mesma lógica computacional e, basicamente, a mesma arquitetura. Nos computadores modernos, é empregada a lógica

A) decimal e a arquitetura Von Neumann.

B) binária e a arquitetura Claude E. Shannon.

C) binária e a arquitetura Von Neumann.

D) hexadecimal e a arquitetura Alan Turing.

E) decimal e a arquitetura Claude E. Shannon.

Comentários:

Sabemos que os computadores utilizam os bits (0 e 1), portanto utiliza uma "lógica" binária. E a arquitetura continua sendo a de von Neumann (em alguns componentes é utilizada a de Harvard, como por exemplo memória cache). Portanto, a alternativa C está correta e é o gabarito da questão.

Gabarito: Letra C

4. (FCC/AFAP - 2019) A soma do hexadecimal 1C5 com o binário de mais baixa ordem 1101, terá como resultado o decimal

A) 434.

B) 466.

C) 737.

D) 479.

E) 482.

Comentários:

Primeiro vamos converter 1C5 para binário (só olhar na tabela, sabendo que cara caractere é convertido para 4 bits, vou separar com um espaço para facilitar) → 0001 1100 0101. Agora é só somar com 1101 (mais baixa ordem quer dizer mais à direita). Então temos:

0001 1100 0101

\+ 1101

0001 1101 0010

Agora vamos para a velha tabela de guerra:

512	256	128	64	32	16	8	4	2	1
	1	1	1	0	1	0	0	1	0

Calculando: 256 + 128 + 64 + 16 + 2 = 466

Portanto, a alternativa B está correta e é o gabarito da questão.

Gabarito: Letra B

5. (Instituto AOCP/PC-ES - 2019) Diferentes sistemas de numeração: base 10 (decimal), base 2 (binário), base 8 (octal), base 16 (hexadecimal), base 256, entre outros, são usados. Assim, sobre sistemas de numeração e codificação e aritmética computacional, é correto afirmar que o resultado da conversão do número em hexadecimal "3A73" para decimal é

A) 21352.

B) 14963.

C) 1000C.

D) 01011.

E) 1B027.

Comentários:

Primeiro já eliminamos as alternativas C e E, pois possuem dígitos em hexadecimal ("letras"). Eliminamos a alternativa A porque é par. Eliminamos a letra D porque é binário! Só sobrou a alternativa B! Questão rápida de resolver na prova!

Mas vamos calcular para treinar, agora utilizando a fórmula (base 16):

3A73

$(3 \times 16^3) + (10 \times 16^2) + (7 \times 16^1) + (3 \times 16^0)$

12288 + 2560 + 112 + 3 = 14963

Portanto, a alternativa B está correta e é o gabarito da questão.

Gabarito: Letra B

6. (COSEAC/UFF - 2019) O número decimal 111 convertido para o sistema binário é escrito como:

A) 1101.

B) 11011.

C) 110111.

D) 1101111.

E) 1110111.

Comentários:

111 / 2 = 55 (resto 1), 55 / 2 = 27 (resto 1), 27 / 2 = 13 (resto 1), 13 / 2 = 6 (resto 1), 6 / 2 = 3 (resto 0), 3 / 2 = 1 (resto 1), 1 / 2 = 0 (resto 1). ← Analisando os restos de trás para frente temos: 1101111.

Portanto, a alternativa D está correta e é o gabarito da questão.

Gabarito: Letra D

7. (IDECAN/IF-PB - 2019) Um sistema de numeração visa a representar valores de forma consistente através da utilização de símbolos e dando significado ao posicionamento destes. O sistema de numeração mais conhecido é o decimal, que opera com a base 10. No entanto,

existem diversos outros sistemas de numeração, como é o caso do octal, hexadecimal e binário. A respeito dos sistemas de numeração e alguns de seus conceitos e considerando a representação 1E1, analise as afirmativas abaixo.

I. É uma representação pertencente ao sistema de numeração hexadecimal.

II. Convertendo-a para base binária, verificaremos que possui LSB igual a 0.

III. Se convertida para a base decimal, equivale ao número 481.

Assinale

A) se somente as afirmativas I e III estiverem corretas.

B) se somente a afirmativa I estiver correta.

C) se somente as afirmativas I e II estiverem corretas.

D) se somente a afirmativa III estiver correta.

E) se todas as afirmativas estiverem corretas.

Comentários:

(I) 1E1 é uma representação pertencente ao sistema de hexadecimal, pois possui os símbolos "0" a "9" e "A" a "F". (II) LSB significa *Least Significant Bit* (bit menos significativo, o que fica mais à direita). "1" em hexadecimal é igual a 0001 em binário, ou seja, o LSB é 1, e não 0! (III) Vamos ao cálculo (lembrando que a base é 16 - hexadecimal):

$1 \times 16^2 + 14 \times 16^1 + 1 \times 16^0$

$256 + 224 + 1 = 481$

Portanto, a alternativa A está correta e é o gabarito da questão.

Gabarito: Letra A

8. (CCV-UFC/UFC - 2019) Em um sistema operacional Linux, um arquivo possui as seguintes permissões: -rwxr--r--. Qual valor representa essa permissão no modo octal?

A) 655

B) 666

C) 711

D) 744

E) 777

Comentários:

Linux não faz parte da aula, mas vale a pena ver uma funcionalidade para o sistema octal na informática.

Vamos ignorar o primeiro "-", pois significa zero e nem é mostrado nas alternativas (zero à esquerda). No restante temos as permissões de leitura (r), escrita (w) e execução (x), para o dono do arquivo, grupo e outros. Quando aparece uma letra, temos o valor 1, se aparece o "-", temos o valor zero. Então temos (já separado de 3 em 3, para dono, grupo e outros: "rwx r-- r--", ou seja, 111 100 100. Agora é só converter de binário para octal. Nem precisa fazer cálculo, é só montar a tabela:

Binário	Octal
000	0
001	1
010	2
011	3
100	4
101	5
110	6
111	7

Portanto, a alternativa D está correta e é o gabarito da questão.

Gabarito: Letra D

9. (Quadrix/CRO-AC - 2019) São exemplos de bases utilizadas nos sistemas de computação a binária (base 2) e a octal (base 8).

Comentários:

Embora a octal seja pouco conhecida, é um exemplo, sim. Em comandos Linux temos um uso maior. A base 2 (binária) é a básica, pois representa o bit, a menor unidade de informação. Existe também a hexadecimal. A decimal é aquela que o ser humano utiliza. Portanto, a questão está correta.

Gabarito: Correta

10. (IDECAN/IF-PB - 2019) Considerando o número 31, em base decimal, assinale a alternativa que indica o valor que representa este número nas bases Hexadecimal e Octal, respectivamente.

A) 3A e 42

B) 2E e 35

C) A e 8

D) 1F e 37

E) 3C e 12

Comentários:

Da base decimal para qualquer outra, temos que dividir...então vamos lá...

31 (base 10) para a base hexadecimal (16):

	Quociente	Resto
31 / 16 =	1	15 (F)
1 / 16 =	0	1

Resultado = 1F

31 (base 10) para a base octal (8):

	Quociente	Resto
31 / 8 =	3	7
3 / 8 =	0	3

Resultado = 37

Portanto, a alternativa D está correta e é o gabarito da questão.

Gabarito: Letra D

LISTA DE QUESTÕES – SISTEMAS DE NUMERAÇÃO - MULTIBANCAS

1. (FGV/TJ-BA - 2015) O número binário 11111010 é representado na notação hexadecimal como:

A) F8

B) AF

C) FF

D) FA

E) FB

2. (FGV/IBGE - 2016) Analise as seguintes equações binárias:

   111 × 11

   111 - 11

   110 ÷ 11

   O resultado das equações apresentadas é, respectivamente:

A) 10101, 10 e 10;

B) 11101, 101 e 11;

C) 10111, 110 e 1;

D) 10101, 100 e 10;

E) 10100, 11 e 101.

3. (CESPE/IFF - 2018) O computador passou por diversas evoluções nos últimos anos; entretanto, continua utilizando a mesma lógica computacional e, basicamente, a mesma arquitetura. Nos computadores modernos, é empregada a lógica

A) decimal e a arquitetura Von Neumann.

B) binária e a arquitetura Claude E. Shannon.

C) binária e a arquitetura Von Neumann.

D) hexadecimal e a arquitetura Alan Turing.

E) decimal e a arquitetura Claude E. Shannon.

4. (FCC/AFAP - 2019) A soma do hexadecimal 1C5 com o binário de mais baixa ordem 1101, terá como resultado o decimal

A) 434.

B) 466.

C) 737.

D) 479.

E) 482.

5. (Instituto AOCP/PC-ES - 2019) Diferentes sistemas de numeração: base 10 (decimal), base 2 (binário), base 8 (octal), base 16 (hexadecimal), base 256, entre outros, são usados. Assim, sobre sistemas de numeração e codificação e aritmética computacional, é correto afirmar que o resultado da conversão do número em hexadecimal "3A73" para decimal é

A) 21352.

B) 14963.

C) 1000C.

D) 01011.

E) 1B027.

6. (COSEAC/UFF - 2019) O número decimal 111 convertido para o sistema binário é escrito como:

A) 1101.

B) 11011.

C) 110111.

D) 1101111.

E) 1110111.

7. (IDECAN/IF-PB - 2019) Um sistema de numeração visa a representar valores de forma consistente através da utilização de símbolos e dando significado ao posicionamento destes. O sistema de numeração mais conhecido é o decimal, que opera com a base 10. No entanto, existem diversos outros sistemas de numeração, como é o caso do octal, hexadecimal e binário. A respeito dos sistemas de numeração e alguns de seus conceitos e considerando a representação 1E1, analise as afirmativas abaixo.

I. É uma representação pertencente ao sistema de numeração hexadecimal.

II. Convertendo-a para base binária, verificaremos que possui LSB igual a 0.

III. Se convertida para a base decimal, equivale ao número 481.

Assinale

A) se somente as afirmativas I e III estiverem corretas.

B) se somente a afirmativa I estiver correta.

C) se somente as afirmativas I e II estiverem corretas.

D) se somente a afirmativa III estiver correta.

E) se todas as afirmativas estiverem corretas.

8. (CCV-UFC/UFC - 2019) Em um sistema operacional Linux, um arquivo possui as seguintes permissões: -rwxr--r--. Qual valor representa essa permissão no modo octal?

A) 655

B) 666

C) 711

D) 744

E) 777

9. (Quadrix/CRO-AC - 2019) São exemplos de bases utilizadas nos sistemas de computação a binária (base 2) e a octal (base 8).

10. (IDECAN/IF-PB - 2019) Considerando o número 31, em base decimal, assinale a alternativa que indica o valor que representa este número nas bases Hexadecimal e Octal, respectivamente.

A) 3A e 42

B) 2E e 35

C) A e 8

D) 1F e 37

E) 3C e 12

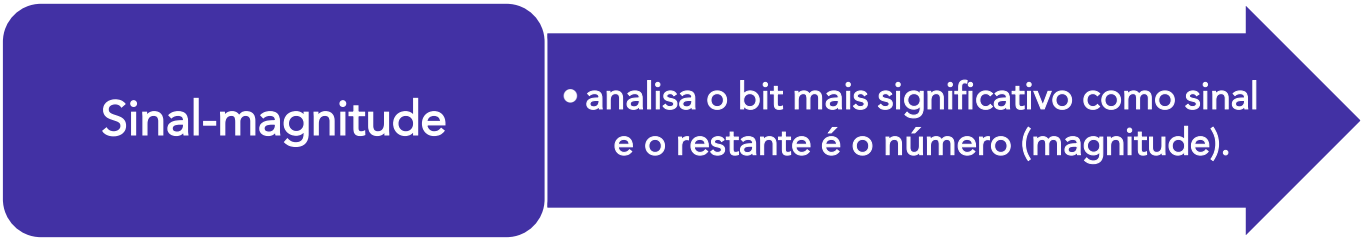
GABARITO

1- D
2- D
3- C
4- B
5- B
6- D
7- A
8- D
9- Correta
10- D

ARITMÉTICA BINÁRIA

Existem diferentes formas de representar números inteiros positivos e negativos, todas envolvendo o bit mais significativo (mais à esquerda) como um bit de sinal. Ou seja, se o bit mais significativo for 0, o número é positivo, se for 1 é negativo.

A forma mais simples de representação é a denominada sinal-magnitude, que simplesmente analisa o bit mais significativo como sinal e o restante é o número (magnitude). Por exemplo, utilizando 8 bits (um para o sinal e sete para o número em si):



+15 = 00001111

-15 = 10001111

Essa representação possui diversas desvantagens, sendo a principal delas a dupla representação do valor 0:

$+0_{10}$ = 00000000

-0_{10} = 10000000

Com essa dupla representação do 0 fica um pouco mais difícil testar se um valor é igual a 0, e essa é uma operação realizada com frequência nos computadores. Por causa desse e de outros problemas, a representação sinal-magnitude quase não é utilizada na implementação da parte inteira da ULA (Unidade Lógica e Aritmética) do processador.

O esquema mais utilizado é a representação em complemento de dois. O bit mais significativo continua sendo utilizado para o sinal, sendo mais fácil testar se um número é positivo ou negativo (0 = positivo, 1 = negativo). Mas o modo como os outros bits são representados é diferente.

Existem explicações em livros, com fórmulas e tudo mais, mas para concurso temos que focar em saber como representar um número rapidamente! Vamos lá...

- Números positivos: ficam da mesma forma que na representação sinal-magnitude. Ex. com 8 bits: 2 = 00000010 em sinal-magnitude, e 00000010 em complemento de dois;

- Números negativos: inverte-se todos os bits do número positivo e soma-se 1. Ex. com 8 bits: -2 → o valor 2 positivo é 00000010 → inverte-se todos os bits: 11111101 → soma-se 1: 11111110
    - Obs.: a simples inversão de todos os bits se chama complemento de um. Ao se adicionar um bit 1, o nome dado é complemento de dois.

Um macete é fazer o seguinte: pega o valor positivo e começa a ver bit a bit da direita para a esquerda até encontrar um bit 1. Até encontrar esse primeiro bit 1 deixa como está, depois dele (o próximo à esquerda) em diante altera tudo! Vamos ver novamente com o valor 2:

Valor positivo: 00000010 ← começa da direita para esquerda até achar o primeiro bit 1, deixa como está tudo que está à direita do primeiro bit 1 e ele também! O resto altera tudo: 11111110. Pronto, acabamos de representar o -2 em complemento de dois. Fica a seu critério escolher como fazer na prova, depois de treinar ver como acha mais fácil e rápido.

Importante salientar que em complemento de dois só há uma representação do ZERO, que é 00000000 (não existe o zero "negativo"). Veja a tabela abaixo com representações em sinal-magnitude e complemento de dois para 4 bits. Sugiro que você tente "calcular" para ver como chegar nos números negativos da representação complemento de dois. Os examinadores adoram cobrar isso!

Decimal	Sinal-magnitude	Complemento de dois
+8	Não tem!	Não tem!
+7	0111	0111
+6	0110	0110
+5	0101	0101
+4	0100	0100
+3	0011	0011
+2	0010	0010
+1	0001	0001
+0	0000	0000
-0	1000	Não tem!
-1	1001	1111
-2	1010	1110
-3	1011	1101
-4	1100	1100
-5	1101	1011

-6	1110	1010
-7	1111	1001
-8	Não tem!	1000

Como chegar no complemento de dois nós vimos, mas e se aparecer na prova um número que já está em complemento de dois, como podemos saber que número em decimal se trata? Aí existe uma "caixa" de valores para conversão, praticamente igual a um macete que já vimos (aquele usado para calcular máscaras de rede). O que muda é que o valor mais à esquerda é negativo. Vamos usar novamente o valor -2, com quatro bits (1110 em complemento de dois):

-8	4	2	1
1	1	1	0

-8 +4 +2 = -2

Vamos ver um exemplo com 8 bits agora, que geralmente é cobrado em prova. Se tiver uma questão que mostra o valor 10000111 em complemento de dois e peça que você marque a alternativa que representa esse valor em decimal. Como fazer? De cara você vê que é um número negativo, então pode começar a fazer a "caixa" de conversão:

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

-128 +4 +2 +1 = -121

Pronto, a resposta é -121! E se essa mesma questão tivesse "falado" em sinal-magnitude? Bom, aí não precisaria essa "caixa", e só saber que "111" é 7, então a resposta seria -7. Muito cuidado! Porque uma das alternativas poderia ser exatamente -7, quando cobrado em complemento de dois. Fique atento!

Outra forma é aquele esquema de contar da direita para a esquerda até encontrar o primeiro bit 1, afinal o negativo do negativo é positivo! Então devemos encontrar 121...vejamos:

10000111 ← já encontramos de primeira, agora é só mudar o resto: 01111001, mas que valor é esse? Agora podemos usar a "caixa" usada para calcular máscaras de rede. Note que agora o valor mais à esquerda não será negativo:

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

+64 +32 +16 +8 +1 = 121

Bom, se a gente fez o complemento de dois de um número negativo (começava com 1) e chegamos a 121, então aquele valor era -121! Veja que há algumas formas de resolver, tente ver qual a que você acha mais fácil, praticando com questões!

Agora vamos ver como "funcionam" a adição e a subtração em complemento de dois. A **adição** acontece como se os dois números não tivessem sinal. Se o resultado for positivo, o número obtido será um número positivo em complemento de dois (que é o mesmo quando não é utilizado sinal). Se o resultado for negativo, esse número estará na forma complemento de dois, ou seja, tem que usar aquela "caixa" para sabermos o valor em decimal. Vamos a alguns exemplos de adição com 4 bits:

```
    1
  1001 = -7 (coloque na "caixa" para verificar, ou faça o complemento de 2 para ver que dá 7)
+ 0101 = 5 (não precisa de "caixa", pois é positivo ☺)
  1110 = -2 (coloque na "caixa" para verificar, ou faça o complemento de 2 para ver que dá 2)

  0011 = 3
+ 0100 = 4
  0111 = 7
```

Os dois exemplos acima não tiveram um **bit de carry** (aquele "vai um") além do **tamanho** da **palavra**. Abaixo veremos dois exemplos em que isso acontece (marcado em vermelho e tachado). Esse bit simplesmente é **ignorado**!

```
 1 1
  1100 = -4
+ 0100 = 4
 ~~1~~0000 = 0

 1 1
  1100 = -4
+ 1111 = -1
 ~~1~~1011 = -5
```

Em qualquer adição o resultado pode extrapolar o tamanho da palavra, o que é chamado de **overflow** (um estouro do tamanho limite "para cima"). Quando ocorre um overflow, a ULA (Unidade Lógica e Aritmética) deve sinalizar para que tal valor não seja utilizado, afinal é um valor incorreto!

Regra para detectar o overflow: se dois números positivos são somados ou se dois números negativos são somados, e o resultado tiver sinal oposto, então acaba de ocorrer um *overflow*! Resumindo: positivo + positivo tem que dar positivo! E negativo + negativo tem que dar negativo!

Caso contrário, tem algo errado, nesse caso... *overflow*! Vejamos dois exemplos (com e sem bit de *carry*):

```
 1
  1001 = -7
+ 1010 = -6
 10011 = 0 → overflow - negativo mais negativo deu positivo! Obs.: (-7) + (-6) não poderia dar 3!)
```

```
   1
  0101 = 5
+ 0100 = 4
  1001 = 0 → overflow - positivo mais positivo deu negativo! Obs.: 5 + 4 não poderia dar (-7)!
```

Agora vamos para a **subtração**. A regra é a seguinte: aplique o complemento de dois no subtraendo (aquele número que está "embaixo", ao lado do sinal de subtração – apenas para ficar mais fácil de explicar, porque nem sempre vem assim) e some esse valor com o minuendo (o outro valor). Vejamos como ficaria 2 – 7 → (0010) – (0111) → aplicando o complemento de 2 em 0111 temos 1001. Agora vamos somar 0010 com 1001:

```
  0010 = 2
+ 1001 = -7
  1011 = -5
```

Vamos ver outros exemplos:

```
 1 1
  0101 = 5
+ 1110 = -2
 10011 = 3
```

Ok, e o *overflow*? Continua a mesma regra! Vejamos:

```
 1
  1010 = -6
+ 1100 = -4
 10110 = overflow - negativo mais negativo deu positivo! Obs.: -6 -4 não poderia dar 6!
```

1. (AOCP/Prefeitura de Novo Hamburgo-RS - 2020) Efetuando a divisão binária de 110 por 11, qual é o valor que um programador obtém?

A) 111

B) 01

C) 00

D) 11

E) 10

Comentários:

Como são números pequenos, dá para fazer a conversão mental para decimal, realizar a divisão, e depois converter o resultado para binário. Aquela tabela que vimos já deve estar em sua cabeça! Ou você deve saber criá-la rapidinho na sua prova. Então, 110 → 6 e 11 → 3. Dividindo 6 por 3, temos 2, que em binário é 10. Portanto, a alternativa E está correta e é o gabarito da questão.

Gabarito: Letra E

2. (CESPE/EBSERH - 2018) Na aritmética computacional, a representação conhecida como sinal e magnitude é utilizada para fazer a distinção entre números positivos e negativos.

Comentários:

O nome já deixa claro: SINAL e MAGNITUDE (o número em si). O primeiro bit é o sinal (0 = positivo, 1 = negativo), os demais bits representam o número (magnitude). Ex. com 8 bits: 10000011 = -3. Portanto, a questão está correta.

Gabarito: Correta

QUESTÕES COMENTADAS – ARITMÉTICA BINÁRIA - MULTIBANCAS

1. (CESPE/Banco da Amazônia - 2010) Considerando a aritmética em sinal e magnitude, se ambos os números têm o mesmo sinal, somam-se as suas magnitudes. Nesse caso, o sinal do resultado é o mesmo das parcelas individuais.

Comentários:

Em aritmética em sinal-magnitude, o primeiro bit é o sinal (0 = positivo, 1 = negativo). A questão quer dizer que se somarmos dois negativos teremos um número negativo, e se somarmos dois positivos teremos como resultado um número positivo. É exatamente o que acontece! Portanto, a questão está correta.

Gabarito: Correta

2. (FGV/IBGE - 2017) O número inteiro –2 (menos dois) tem a seguinte representação em 16 bits, usando complemento a 2:

A) 1000000000000010;

B) 1111111111111110;

C) 0111111111111110;

D) 1010101010101010;

E) 0101010101010101.

Comentários:

O nosso velho macete de guerra: vamos pegar o valor 2 (decimal) e transformar em binário (16 bits): 0000000000000010 → da direita para a esquerda até encontrar o 1° bit 1 → inverte o resto → 1111111111111110. Portanto, a alternativa B está correta e é o gabarito da questão.

Gabarito: Letra B

3. (FCM/IF Baiano - 2017) No contexto da álgebra computacional, pela regra de complemento a 2, é possível manipular tanto números positivos, quanto números negativos. Dado o número binário 0000010100110010 (sinal magnitude), o seu hexadecimal, após o complemento a 2 resultante, será

A) FACA

B) CAFE

C) FACE

D) CADA

E) FADA

Comentários:

Vamos usar aquele macete da direita para a esquerda até achar o primeiro bit 1 → 0000010100110010 → vamos inverter o resto: 1111101011001110 → agora vamos separar de quatro em quatro bits: 1111 1010 1100 1110 → olhando na tabela hexadecimal (você tem que saber montar a sua!): FACE.

Portanto, a alternativa C está correta e é o gabarito da questão.

Gabarito: Letra C

4. (CESPE/EBSERH - 2018) Na aritmética computacional, a representação conhecida como sinal e magnitude é utilizada para fazer a distinção entre números positivos e negativos.

Comentários:

O nome já deixa claro: SINAL e MAGNITUDE (o número em si). O primeiro bit é o sinal (0 = positivo, 1 = negativo), os demais bits representam o número (magnitude). Ex. com 8 bits: 10000011 = -3. Portanto, a questão está correta.

Gabarito: Correta

5. (CESGRANRIO/Petrobras - 2018) Alguns sistemas computacionais costumam representar números negativos em complemento a 2, o que facilita as operações de subtração, já que é possível implementá-las como uma soma ao complemento.

Em um sistema que representa números com 8 bits e usa o complemento a 2, a operação 0A + F8, em que os números estão representados em hexadecimal, gera como resultado

A) -3

B) -2

C) 1

D) 2

E) um erro de overflow

Comentários:

Primeiro vamos transformar em binário, sabendo que cada caractere em hexadecimal representa 4 bits (vou separar para ficar melhor de visualizar): 0000 1010 + 1111 1000. Vamos ao cálculo:

```
  1 1 1 1 1
   00001010 = 10
+ 11111000 = -8
 100000010 = 2
```

Fazendo o cálculo conseguimos chegar no resultado 2, tranquilamente. O 10 também é tranquilo de converter (nem precisava), mas vamos ver como chegamos ao (-8) no subtraendo. Para variar um pouco não vamos usar a "caixa", vamos aplicar o complemento de 2 ao valor "11111000" → da direita para a esquerda mantém tudo até o primeiro bit 1 e inverte o resto: 00001000 = 8. Se aplicamos o complemento de 2 e chegamos a 8, isso quer dizer que o número era -8!

Portanto, a alternativa D está correta e é o gabarito da questão.

Gabarito: Letra D

6. (CEPS-UFPA/UFPA - 2018) Em complemento a 2, a representação em binário da operação 13 – 6 é

A) 1101 + 1010.

B) 1101 – 0111.

C) 1101 + 1001.

D) 1100 – 1001.

E) 1100 – 1111.

Comentários:

Pessoal, confesso que essa questão está "estranha", pois com 4 bits, sendo um para o sinal, não tem como representar o valor 13! Mas abstraindo isso, vamos ver o que o examinador quis que entendêssemos...

Em complemento a 2, quando temos a subtração de dois números positivos, mantemos o minuendo (o primeiro) e somamos com o complemento a 2 do subtraendo. Vou "inventar" um bit a mais porque com 4 não é possível! Primeiro vamos fazer o complemento a 2 do valor 6: 00110 → 11010. Agora vamos à adição:

```
 1 1
  01101 = 13
+ 11010 = -6
 100111 = 7
```

Note que analisando apenas os quatro bits a resposta é a alternativa A (que é a resposta da banca). No dia da prova você deve fazer assim, mesmo com um erro grosseiro desses. Portanto, a alternativa A é o gabarito da questão.

Gabarito: Letra A

7. (FGV/AL-RO - 2018) Considere a valor hexadecimal FFFFFFFF.

Dado que este valor binário está representado na notação de complemento para dois, assinale o valor decimal desse número.

A) 0

B) 1

C) -1

D) 65.535

E) 4.294.967.295

Comentários:

O valor mostrado é 11111111, então sabemos que é negativo! Agora vamos aplicar o complemento de 2 para vermos qual a magnitude (valor): 11111111 → 00000001, ou seja 1 em decimal. Então a resposta é -1. Portanto, a alternativa C está correta e é o gabarito da questão.

Gabarito: Letra C

8. (Instituto UniFil/Prefeitura de Cunha Porã-SC - 2020) O sistema binário usado pelos computadores é constituído de dois dígitos [0,1]. A combinação desses dígitos leva o computador a criar várias informações: letras, palavras, textos, cálculos. Considerando o tema, números binários, assinale a alternativa que representa o resultado da operação da soma binária: 101101 + 11100011.

A) 010010010

B) 111001001

C) 100010000

D) 101010001

Comentários:

Vamos ao cálculo:

```
  1 1     1 1 1 1
      101101
+11100011
100010000
```

Portanto, a alternativa C está correta e é o gabarito da questão.

Gabarito: Letra C

9. (AOCP/Prefeitura de Novo Hamburgo-RS - 2020) Um programador necessita calcular o complemento de 1 do número binário 1 0 1 1 0 0 1 0. Sendo assim, assinale a alternativa que apresenta o complemento de 1 do binário dado.

A) 0 1 0 0 1 1 0 1

B) 1 1 1 1 1 1 1 1

C) 1 1 0 0 0 0 1 1

D) 1 0 0 0 0 0 0 1

E) 0 1 1 1 1 1 1 0

Comentários:

Para calcular o complemento de um basta inverter todos os bits. Para calcular o complemento de dois, após inverter todos os bits é só adicionar o bit 1 (ou aquele bizu que vimos na aula, mas o foco agora é complemento de um!). Invertendo todos os bits do valor proposto no enunciado temos: 01001101. Portanto, a alternativa A está correta e é o gabarito da questão.

Gabarito: Letra A

10. (AOCP/Prefeitura de Novo Hamburgo-RS - 2020) Efetuando a divisão binária de 110 por 11, qual é o valor que um programador obtém?

A) 111

B) 01

C) 00

D) 11

E) 10

Comentários:

Como são números pequenos, dá para fazer a conversão mental para decimal, realizar a divisão, e depois converter o resultado para binário. Aquela tabela que vimos já deve estar em sua cabeça! Ou você deve saber criá-la rapidinho na sua prova. Então, 110 → 6 e 11 → 3. Dividindo 6 por 3, temos 2, que em binário é 10. Portanto, a alternativa E está correta e é o gabarito da questão.

Gabarito: Letra E

LISTA DE QUESTÕES – ARITMÉTICA BINÁRIA - MULTIBANCAS

1. (CESPE/Banco da Amazônia - 2010) Considerando a aritmética em sinal e magnitude, se ambos os números têm o mesmo sinal, somam-se as suas magnitudes. Nesse caso, o sinal do resultado é o mesmo das parcelas individuais.
2. (FGV/IBGE - 2017) O número inteiro –2 (menos dois) tem a seguinte representação em 16 bits, usando complemento a 2:

A) 1000000000000010;

B) 1111111111111110;

C) 0111111111111110;

D) 1010101010101010;

E) 0101010101010101.

3. (FCM/IF Baiano - 2017) No contexto da álgebra computacional, pela regra de complemento a 2, é possível manipular tanto números positivos, quanto números negativos. Dado o número binário 0000010100110010 (sinal magnitude), o seu hexadecimal, após o complemento a 2 resultante, será

A) FACA

B) CAFE

C) FACE

D) CADA

E) FADA

4. (CESPE/EBSERH - 2018) Na aritmética computacional, a representação conhecida como sinal e magnitude é utilizada para fazer a distinção entre números positivos e negativos.
5. (CESGRANRIO/Petrobras - 2018) Alguns sistemas computacionais costumam representar números negativos em complemento a 2, o que facilita as operações de subtração, já que é possível implementá-las como uma soma ao complemento.

Em um sistema que representa números com 8 bits e usa o complemento a 2, a operação 0A + F8, em que os números estão representados em hexadecimal, gera como resultado

A) -3

B) -2

C) 1

D) 2

E) um erro de overflow

6. (CEPS-UFPA/UFPA - 2018) Em complemento a 2, a representação em binário da operação 13 – 6 é

A) 1101 + 1010.

B) 1101 – 0111.

C) 1101 + 1001.

D) 1100 – 1001.

E) 1100 – 1111.

7. (FGV/AL-RO - 2018) Considere a valor hexadecimal FFFFFFFF.

Dado que este valor binário está representado na notação de complemento para dois, assinale o valor decimal desse número.

A) 0

B) 1

C) -1

D) 65.535

E) 4.294.967.295

8. (Instituto UniFil/Prefeitura de Cunha Porã-SC - 2020) O sistema binário usado pelos computadores é constituído de dois dígitos [0,1]. A combinação desses dígitos leva o computador a criar várias informações: letras, palavras, textos, cálculos. Considerando o tema,

números binários, assinale a alternativa que representa o resultado da operação da soma binária: 101101 + 11100011.

A) 010010010

B) 111001001

C) 100010000

D) 101010001

9. (AOCP/Prefeitura de Novo Hamburgo-RS - 2020) Um programador necessita calcular o complemento de 1 do número binário 1 0 1 1 0 0 1 0. Sendo assim, assinale a alternativa que apresenta o complemento de 1 do binário dado.

A) 0 1 0 0 1 1 0 1

B) 1 1 1 1 1 1 1 1

C) 1 1 0 0 0 0 1 1

D) 1 0 0 0 0 0 0 1

E) 0 1 1 1 1 1 1 0

10. (AOCP/Prefeitura de Novo Hamburgo-RS - 2020) Efetuando a divisão binária de 110 por 11, qual é o valor que um programador obtém?

A) 111

B) 01

C) 00

D) 11

E) 10

GABARITO

1- Correta
2- B
3- C
4- Correta
5- D
6- A
7- C
8- C
9- A
10- E

Representação Numérica, Aritmética Binária e Representação dos Dados

Existem diferentes formas de representar números inteiros positivos e negativos, todas envolvendo o bit mais significativo (mais à esquerda) como um bit de sinal. Ou seja, se o bit mais significativo for 0, o número é positivo, se for 1 é negativo.

A forma mais simples de representação é a denominada sinal-magnitude, que simplesmente analisa o bit mais significativo como sinal e o restante é o número (magnitude). Por exemplo, utilizando 8 bits (um para o sinal e sete para o número em si):

+15 = 00001111

-15 = 10001111

Essa representação possui diversas desvantagens, sendo a principal delas a dupla representação do valor 0:

$+0_{10}$ = 00000000

-0_{10} = 10000000

Com essa dupla representação do 0 fica um pouco mais difícil testar se um valor é igual a 0, e essa é uma operação realizada com frequência nos computadores. Por causa desse e de outros problemas, a representação sinal-magnitude quase não é utilizada na implementação da parte inteira da ULA (Unidade Lógica e Aritmética) do processador.

O esquema mais utilizado é a representação em complemento de dois. O bit mais significativo continua sendo utilizado para o sinal, sendo mais fácil testar se um número é positivo ou negativo (0 = positivo, 1 = negativo). Mas o modo como os outros bits são representados é diferente.

Existem explicações em livros, com fórmulas e tudo mais, mas para concurso temos que focar em saber como representar um número rapidamente! Vamos lá...

- Números positivos: ficam da mesma forma que na representação sinal-magnitude. Ex. com 8 bits: 2 = 00000010 em sinal-magnitude, e 00000010 em complemento de dois;
- Números negativos: inverte-se todos os bits do número positivo e soma-se 1. Ex. com 8 bits: -2 → o valor 2 positivo é 00000010 → inverte-se todos os bits: 11111101 → soma-se 1: 11111110
    - Obs.: a simples inversão de todos os bits se chama complemento de um. Ao se adicionar um bit 1, o nome dado é complemento de dois.

Um macete é fazer o seguinte: pega o valor positivo e começa a ver bit a bit da direita para a esquerda até encontrar um bit 1. Até encontrar esse primeiro bit 1 deixa como está, depois dele (o próximo à esquerda) em diante altera tudo! Vamos ver novamente com o valor 2:

Valor positivo: 00000010 começa da direita para esquerda até achar o primeiro bit 1, deixa como está tudo que está à direita do primeiro bit 1 e ele também! O resto altera tudo: 11111110. Pronto, acabamos de representar o -2 em complemento de dois. Fica a seu critério escolher como fazer na prova, depois de treinar ver como acha mais fácil e rápido.

Importante salientar que em complemento de dois só há uma representação do ZERO, que é 00000000 (não existe o zero "negativo"). Veja a tabela abaixo com representações em sinal-magnitude e complemento de dois para 4 bits. Sugiro que você tente "calcular" para ver como chegar nos números negativos da representação complemento de dois. Os examinadores adoram cobrar isso!

Decimal	Sinal-magnitude	Complemento de dois
+8	Não tem!	Não tem!
+7	0111	0111
+6	0110	0110
+5	0101	0101
+4	0100	0100
+3	0011	0011
+2	0010	0010
+1	0001	0001
+0	0000	0000
-0	1000	Não tem!
-1	1001	1111
-2	1010	1110
-3	1011	1101
-4	1100	1100
-5	1101	1011
-6	1110	1010
-7	1111	1001
-8	Não tem!	1000

Como chegar no complemento de dois nós vimos, mas se aparecer na prova um número que já está em complemento de dois, como podemos saber que número em decimal se trata? Aí existe uma "caixa" de valores para conversão, praticamente igual a um macete que já vimos (aquele usado para calcular máscaras de rede). O que muda é que o valor mais à esquerda é negativo. Vamos usar novamente o valor -2, com quatro bits (1110 em complemento de dois):

-8	4	2	1
1	1	1	0

-8 +4 +2 = -2

Vamos ver um exemplo com 8 bits agora, que geralmente é cobrado em prova. Se tiver uma questão que mostra o valor 10000111 em complemento de dois e peça que você marque a alternativa que representa esse valor em decimal. Como fazer? De cara você vê que é um número negativo, então pode começar a fazer a "caixa" de conversão:

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

-128 +4 +2 +1 = -121

Pronto, a resposta é -121! E se essa mesma questão tivesse "falado" em sinal-magnitude? Bom, aí não precisaria essa "caixa", e só saber que "111" é 7, então a resposta seria -7. Muito cuidado! Porque uma das alternativas poderia ser exatamente -7, quando cobrado em complemento de dois. Fique atento!

Outra forma é aquele esquema de contar da direita para a esquerda até encontrar o primeiro bit 1, afinal o negativo do negativo é positivo! Então devemos encontrar 121...vejamos:

10000111 já encontramos de primeira, agora é só mudar o resto: 01111001, mas que valor é esse? Agora podemos usar a "caixa" usada para calcular máscaras de rede. Note que agora o valor mais à esquerda não será negativo:

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

+64 +32 +16 +8 +1 = 121

Bom, se a gente fez o complemento de dois de um número negativo (começava com 1) e chegamos a 121, então aquele valor era -121! Veja que há algumas formas de resolver, tente ver qual a que você acha mais fácil, praticando com questões!

Agora vamos ver como "funcionam" a adição e a subtração em complemento de dois. A adição acontece como se os dois números não tivessem sinal. Se o resultado for positivo, o número obtido será um número positivo em complemento de dois (que é o mesmo quando não é utilizado sinal). Se o resultado for negativo, esse número estará na forma complemento de dois, ou seja, tem que usar aquela "caixa" para sabermos o valor em decimal. Vamos a alguns exemplos de adição com 4 bits:

```
    1
  1001 = -7 (coloque na "caixa" para verificar, ou faça o complemento de 2 para ver que dá 7)
+ 0101 = 5 (não precisa de "caixa", pois é positivo ☺)
  1110 = -2 (coloque na "caixa" para verificar, ou faça o complemento de 2 para ver que dá 2)

  0011 = 3
+ 0100 = 4
  0111 = 7
```

Os dois exemplos acima não tiveram um bit de *carry* (aquele "vai um") além do tamanho da palavra. Abaixo veremos dois exemplos em que isso acontece (marcado em vermelho e tachado). Esse bit simplesmente é ignorado!

```
 1 1
  1100 = -4
+ 0100 = 4
 ~~1~~0000 = 0

 1 1
  1100 = -4
+ 1111 = -1
 ~~1~~1011 = -5
```

Em qualquer adição o resultado pode extrapolar o tamanho da palavra, o que é chamado de *overflow* (um estouro do tamanho limite "para cima"). Quando ocorre um overflow, a ULA (Unidade Lógica e Aritmética) deve sinalizar para que tal valor não seja utilizado, afinal é um valor incorreto!

Regra para detectar o *overflow*: se dois números positivos são somados ou se dois números negativos são somados, e o resultado tiver sinal oposto, então acaba de ocorrer um *overflow*! Resumindo: positivo + positivo tem que dar positivo! E negativo + negativo tem que dar negativo! Caso contrário, tem algo errado, nesse caso...*overflow*! Vejamos dois exemplos (com e sem bit de *carry*):

```
  1
  1001 = -7
+ 1010 = -6
 ~~1~~0011 = 0 → overflow - negativo mais negativo deu positivo! Obs.: (-7) + (-6) não poderia dar 3!)

   1
  0101 = 5
```

+ 0100 = 4
  1001 = 0 → *overflow* - positivo mais positivo deu negativo! Obs.: 5 + 4 não poderia dar (-7)!

Agora vamos para a subtração. A regra é a seguinte: aplique o complemento de dois no subtraendo (aquele número que está "embaixo", ao lado do sinal de subtração – apenas para ficar mais fácil de explicar, porque nem sempre vem assim) e some esse valor com o minuendo (o outro valor). Vejamos como ficaria 2 – 7 → (0010) – (0111) → aplicando o complemento de 2 em 0111 temos 1001. Agora vamos somar 0010 com 1001:

  0010 = 2
+ 1001 = -7
  1011 = -5

Vamos ver outros exemplos:

 1 1
  0101 = 5
+ 1110 = -2
 10011 = 3

Ok, e o *overflow*? Continua a mesma regra! Vejamos:

  1
  1010 = -6
+ 1100 = -4
 10110 = *overflow* - negativo mais negativo deu positivo! Obs.: -6 -4 não poderia dar 6!

Underflow: É a condição na qual o resultado de uma operação aritmética é menor do que o valor mínimo representável dentro de um sistema numérico. É o oposto do *overflow*! Por exemplo, em operações com números em ponto flutuante, um *underflow* ocorre quando o resultado é menor do que o menor valor representável nesse sistema. Essa condição pode ocorrer em cálculos que resultam em números muito pequenos, levando à perda de precisão ou ao arredondamento para zero.

Aritmética Computacional

Em binário e hexadecimal há a multiplicação e a divisão pela base, da mesma forma como ocorre em decimal.

Para ficar mais fácil de entender, vamos começar pela base decimal, onde temos 10 valores possíveis (0 a 9). Se multiplicarmos por 10, "movemos a vírgula" para a direita. Se multiplicarmos por 100 (10 x 10) então movemos duas casas para a direita (se não houver casas decimais, então acrescenta-se zero). Com a divisão por 10, a "vírgula é movida" para a esquerda. Exemplos:

1035 / 10 = 103,5

1035 / 100 = 10,35

1035 x 10 = 10350

1035 x 100 = 103500

Agora vamos ver alguns exemplos para a base binária (2 valores possíveis, 0 ou 1):

$101010,11_2 \times 2 = 1010101,1_2$

$101010,11_2 \times 4 = 10101011_2$

$101010,11_2 / 2 = 10101,011_2$

$101010,11_2 / 4 = 1010,1011_2$

E, para finalizar, uns exemplos para a base hexadecimal (16 valores possíveis, 0 a F):

$F,A5_{16} \times 16 = FA,5_{16}$

$F,A5_{16} \times 256 = FA5_{16}$

$FA5_{16} / 16 = FA,5_{16}$

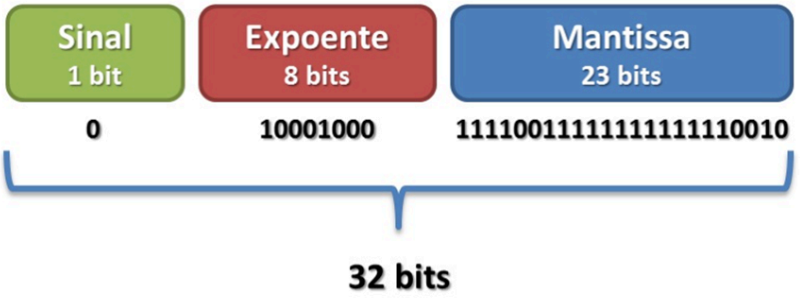
$FA5_{16} / 256 = F,A5_{16}$

Obs.: 4 = 2 x 2 e 256 = 16 x 16, por isso "move a vírgula duas casas".

Ponto Flutuante

Ponto flutuante é uma forma de representação numérica utilizada em computação para representar números reais (números com casas decimais) de forma aproximada. Um número é representado por dois componentes principais: a mantissa (ou fração) e o expoente.

Geralmente utiliza-se o padrão IEEE 754, que é um formato amplamente adotado para representação de números em ponto flutuante em computadores. Neste padrão, um número em ponto flutuante é representado por três campos: o bit de sinal, o expoente e a mantissa (ou fração). Um exemplo com 32 bits:



Por exemplo, para representar -128 em ponto flutuante (padrão IEEE 754), de precisão simples (32 bits), a representação é a seguinte:

- Bit de sinal: 1 (indica número negativo);
- Expoente: 10000000 (para representar -128, o expoente é -7 em decimal, que em binário é 10000000);
- Mantissa: 00000000000000000000000 (a mantissa para números inteiros é sempre 0).

Então, a representação em ponto flutuante de -128 seria:

1 10000000 00000000000000000000000

Sendo que o primeiro bit é o bit de sinal, os próximos 8 bits são o expoente e os 23 bits seguintes são a mantissa.

Representação dos Dados

Toda informação recebida pelo computador, sejam dados a serem processados ou instruções de um programa, necessitam ser compreendidos pela máquina, possibilitando a correta interpretação e o devido processamento.

O computador é um equipamento eletrônico, que armazena e transfere as informações sob a forma eletrônica (um valor de voltagem ou corrente, ex.: 2V a 5V). Na memória secundária (HDs, CDs etc.) as informações são armazenadas sob a forma magnética (ex.: HD), ótica (ex.: CD) ou memória *flash* (ex.: pen drive).

Se um computador representasse eletricamente todos os símbolos representados na linguagem humana (“a” até “z”, “A” até “Z”, números etc.), seriam necessárias centenas de valores de voltagem (ou corrente) diferentes, o que seria algo muito complexo e teria uma baixa confiabilidade!

Visto tudo isso, optou-se por uma máquina binária (apenas dois valores possíveis para cada “símbolo”), o que tornou muito mais simples e confiável projetar um circuito. Esses valores possíveis são o 0 e 1. Além disso, com uma máquina binária, é mais fácil aplicar a lógica booleana (Verdadeiro ou Falso). E como podemos representar a letra “a”, por exemplo? Através de uma sequência de 0’s e 1’s, definida em uma tabela pré-definida (depende qual a codificação utilizada).

Depois dessa introdução, concluímos que a menor unidade de informação armazenável em um computador é um dígito binário (*binary digit*, o famoso bit!). As informações manipuladas por um computador são codificadas em grupos ordenados de bits, tendo algum significado útil.

Quantos bits são necessários para representar um caractere? Depende da codificação utilizada! Por que isso? Porque a língua inglesa não precisa representar o “ç”, por exemplo, mas a língua portuguesa precisa! E o mandarim? Vixe...é necessária a representação de dezenas de milhares de símbolos diferentes! A fórmula é simples: 2^x = quantidade de caracteres diferentes que podem ser representados. Vamos ver alguns exemplos:

2^7 = 128, ou seja, com 7 bits é possível representar 128 caracteres diferentes.

2^8 = 256, ou seja, com 8 bits é possível representar 256 caracteres diferentes.

2^{16} = 65536, ou seja, com 16 bits é possível representar 65536 caracteres diferentes.

Vamos ver alguns dos tipos de codificação de caracteres mais conhecidos:

- ASCII (*American Standard Code for Information Interchange*): utiliza 7 bits (valores 0 a 127, em decimal). Abaixo é mostrado um pedaço da tabela ASCII, note que "A" é representado pelo valor 65 (em decimal) e "a" pelo valor 97;

Bin	Oct	Dec	Hex	Sinal
0010 0000	040	32	20	(espaço)
0010 0001	041	33	21	!
0010 0010	042	34	22	"
0010 0011	043	35	23	#
0010 0100	044	36	24	$
0010 0101	045	37	25	%
0010 0110	046	38	26	&
0010 0111	047	39	27	'
0010 1000	050	40	28	(

Bin	Oct	Dec	Hex	Sinal
0100 0000	100	64	40	@
0100 0001	101	65	41	A
0100 0010	102	66	42	B
0100 0011	103	67	43	C
0100 0100	104	68	44	D
0100 0101	105	69	45	E
0100 0110	106	70	46	F
0100 0111	107	71	47	G
0100 1000	110	72	48	H

Bin	Oct	Dec	Hex	Sinal
0110 0000	140	96	60	`
0110 0001	141	97	61	a
0110 0010	142	98	62	b
0110 0011	143	99	63	c
0110 0100	144	100	64	d
0110 0101	145	101	65	e
0110 0110	146	102	66	f
0110 0111	147	103	67	g
0110 1000	150	104	68	h

- ASCII Estendido: utiliza 8 bits para representar caracteres. Na verdade, utiliza os mesmos 128 caracteres da tabela ASCII e foram acrescidos do 128 ao 255, conforme podemos ver no pedaço da tabela abaixo (destaque para "ç" = 135 em decimal);

Decimal	Binário	Hex	Referência
131	10000011	83	â
132	10000100	84	ä
133	10000101	85	à
134	10000110	86	å
135	10000111	87	ç
136	10001000	88	ê
137	10001001	89	ë
138	10001010	8A	è

- UNICODE: utiliza 16 bits para representar caracteres, ou seja, pode representar 65536 símbolos! Essa codificação pretende ser universal, possibilitando representar qualquer linguagem conhecida no mundo!

Um agrupamento de bits bastante conhecida e utilizada é o byte, que são 8 bits tratados de forma individual, como unidade de armazenamento e transferência. Você já está acostumado a dizer que o arquivo que você acabou de gravar possui 500 bytes, por exemplo. Agora vamos ver uma tabela das grandezas utilizadas para abreviar valores:

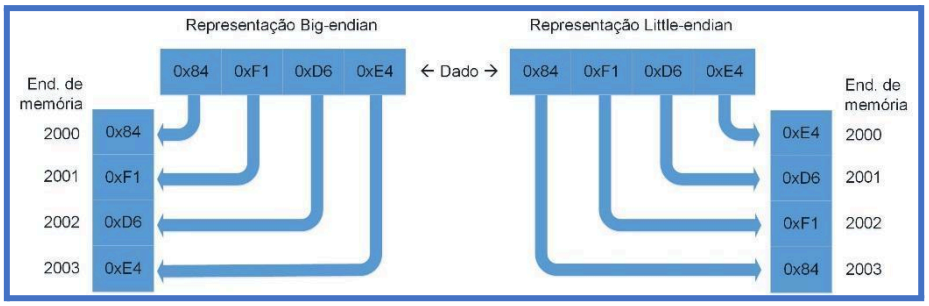
Nome da unidade	Valor em potência de 2	Valor em unidades
1K (1 quilo)	2^{10}	1024
1M (1 mega)	2^{20} = 1024K	1.048.576
1G (1 giga)	2^{30} = 1024M	Mais que 1 bilhão
1T (1 tera)	2^{40} = 1024G	Mais que 1 trilhão
1P (1 peta)	2^{50}	Mais que 1 quatrilhão
1Ex (1 exa)	2^{60}	...
1Z (1 zeta)	2^{70}	...

Na verdade, 1K seria 1000, segundo o Sistema Internacional de Unidades, mas os concursos ainda levam em consideração a tabela acima. Claro que se na prova aparecer que 1K = 1000, 1M = 1000000, e assim por diante, tudo bem! Você tem que pescar do examinador o que ele realmente quer!

Um fator relevante na análise de dados em memória é a ordem de sua representação. Essa ordem varia de acordo com a arquitetura de computador utilizada. Existem dois sistemas, o *big endian* e o *little endian*, conforme explicado abaixo.

- *Little endian*: o byte menos significativo é armazenado no endereço de memória menor;
- *Big endian*: o byte mais significativo é armazenado no endereço de memória menor.

Por exemplo, em uma palavra de memória de 32 bits (4 bytes) o valor 0x84F1D6E4 armazenado nos endereços de memória 2000 a 2003 pode ser representado da seguinte forma:



De uma forma simplista, podemos dizer que o sistema *big endian* é como nós estamos acostumados a escrever um valor, da esquerda para a direita, byte a byte. E o sistema *little endian* é o contrário, da direita para a esquerda. Ou seja, *big endian* representaria o valor como 0x84F1D6E4 e o *little endian* como 0xE4D6F184.

QUESTÕES COMENTADAS - REPRESENTAÇÃO DOS DADOS-MULTIBANCAS

1. (IF-RJ/IF-RJ - 2010) Desde o advento da computação, vários códigos de caracteres foram desenvolvidos para representação interna de informação das máquinas. Assinale a alternativa que representa o código de 16 bits, por símbolo, que pretende codificar em um único código símbolos de qualquer linguagem conhecida no mundo, solucionando o problema dos demais códigos atuais, onde ocorre a necessidade de diversos conjuntos de códigos ou versões.

A) BCD

B) EBCDIC

C) ASCII

D) UNICODE

E) ASCIII

Comentários:

Quando "fala" em 16 bits já fique atento em UNICODE! O EBCDIC (8 bits) descende do BCD (6bits). ASCII possui 7 bits e a versão estendida possui 8 bits. Portanto, a alternativa D está correta e é o gabarito da questão.

Gabarito: Letra D

2. (MPE-RS/MPE-RS - 2012) Um valor inteiro (32 bits) está armazenado na memória com sua parte mais significativa no endereço inferior de uma área de memória. Essa representação é denominada

A) Little endian

B) Big endian

C) LSB (Least Significant Byte)

D) MSB (Most Significant Byte)

E) Opcode

Comentários:

Na representação big endian o valor fica armazenado na memória do jeito que estamos acostumados a escrever no papel. Ex.: AB 58 -> AB fica à esquerda (endereço menor de memória) e 58 fica à direita (endereço maior na memória). Little endian é o contrário, então AB 58 ficaria armazenado assim: 58 AB.

Portanto, a alternativa B está correta e é o gabarito da questão.

Gabarito: Letra B

3. (FGV/TJ-GO - 2014) Na codificação binária das mensagens de um certo protocolo, o padrão BIG ENDIAN é utilizado para representar inteiros de 32 bits. Isso significa que o inteiro 257 será representado pela seguinte sequência de bytes:

A) 00 00 01 01

B) 00 01 01 00

C) 01 01 00 00

D) 01 00 01 00

E) 01 00 00 01

Comentários:

Big endian é exatamente como nós estamos acostumados a escrever, da esquerda para a direita. Little endian inverte tudo! Então temos que ver como representar 257 em hexadecimal, pois as alternativas mostram 4 bytes, cada um representado por 2 caracteres em hexadecimal. Vamos ver em binário como fica o valor 257 (256 + 1):

...	256	128	64	32	16	8	4	2	1
...0000 000	1	0	0	0	0	0	0	0	1

Agrupando de 4 em 4 bits, temos: 00 00 01 01.

Portanto, a alternativa A está correta e é o gabarito da questão.

Gabarito: Letra A

4. (UFES/UFES - 2016) Esquemas de codificação, ou codificação de caracteres, possibilitam uma maneira comum para representar um caractere de dados na computação. A alternativa que NÃO apresenta um esquema de codificação é:

A) ASCII.

B) EBCDIC.

C) ISO 8859.

D) ABC.

E) Unicode.

Comentários:

ASCII e Unicode nós vimos nesta aula. EBCDIC (Extended Binary Coded Decimal Interchange Code) é uma codificação de caracteres de 8 bits que descende diretamente do código BCD (6 bits) e foi criado pela IBM como um padrão no início dos anos 1960. ISO 8859 é uma série de padrões de codificação de caracteres de 8 bits. ABC? Foi um nome inventado! Aliás, poderiam ter colocado alguma sigla que exista para dificultar um pouco 😊. Portanto, a alternativa D está correta e é o gabarito da questão.

Gabarito: Letra D

5. (FGV/IBGE - 2017) Uma aplicação deseja fazer um determinado processamento numérico, mas os dados fornecidos virão de computadores com diferentes estruturas de codificação numérica (little-endian ou big-endian), e portanto precisam sofrer uma prévia conversão de formato antes de serem processados. No modelo OSI, a camada adequada para realizar essa conversão é a camada de:

A) enlace;

B) rede;

C) transporte;

D) sessão;

E) apresentação.

Comentários:

Sei que esta aula não está relacionada com redes de computadores! Mas achei interessante mostrar uma das finalidades da camada de apresentação do modelo OSI. Muitos alunos têm dificuldade em entender essa camada, então aí está um belo exemplo! Se a codificação estiver em little endian e for interpretada como big endian (ou vice-versa), o valor será tratado equivocadamente, gerando erros grosseiros. Portanto, a alternativa E está correta e é o gabarito da questão.

Gabarito: Letra E

LISTA DE QUESTÕES - REPRESENTAÇÃO DOS DADOS- MULTIBANCAS

1. (IF-RJ/IF-RJ - 2010) Desde o advento da computação, vários códigos de caracteres foram desenvolvidos para representação interna de informação das máquinas. Assinale a alternativa que representa o código de 16 bits, por símbolo, que pretende codificar em um único código símbolos de qualquer linguagem conhecida no mundo, solucionando o problema dos demais códigos atuais, onde ocorre a necessidade de diversos conjuntos de códigos ou versões.

A) BCD

B) EBCDIC

C) ASCII

D) UNICODE

E) ASCIII

2. (MPE-RS/MPE-RS - 2012) Um valor inteiro (32 bits) está armazenado na memória com sua parte mais significativa no endereço inferior de uma área de memória. Essa representação é denominada

A) Little endian

B) Big endian

C) LSB (Least Significant Byte)

D) MSB (Most Significant Byte)

E) Opcode

3. (FGV/TJ-GO - 2014) Na codificação binária das mensagens de um certo protocolo, o padrão BIG ENDIAN é utilizado para representar inteiros de 32 bits. Isso significa que o inteiro 257 será representado pela seguinte sequência de bytes:

A) 00 00 01 01

B) 00 01 01 00

C) 01 01 00 00

D) 01 00 01 00

E) 01 00 00 01

4. (UFES/UFES - 2016) Esquemas de codificação, ou codificação de caracteres, possibilitam uma maneira comum para representar um caractere de dados na computação. A alternativa que NÃO apresenta um esquema de codificação é:

A) ASCII.

B) EBCDIC.

C) ISO 8859.

D) ABC.

E) Unicode.

5. (FGV/IBGE - 2017) Uma aplicação deseja fazer um determinado processamento numérico, mas os dados fornecidos virão de computadores com diferentes estruturas de codificação numérica (little-endian ou big-endian), e portanto precisam sofrer uma prévia conversão de formato antes de serem processados. No modelo OSI, a camada adequada para realizar essa conversão é a camada de:

A) enlace;

B) rede;

C) transporte;

D) sessão;

E) apresentação.

GABARITO

GABARITO

1- D
2- B
3- A
4- D
5- E

PROPOSIÇÕES X TABELA VERDADE

Bom, antes de iniciarmos com a ver as tabelas verdades é importante ver o conceito de uma proposição! Trata-se de uma oração declarativa que admite um valor lógico (V ou F). Por exemplo:

- "Eu estudo para concursos."
- "Eu estudo para concursos e gosto de viajar."
- "Se eu estudo para concursos, então gosto de viajar."

Agora um exemplo que não é uma proposição:

"Boa noite!"

Por quê? Ora, não podemos definir um valor lógico V ou F para "Boa noite!". Simples assim!

Em provas de concurso podemos encontrar em forma de orações (como acabamos de ver) ou na forma de expressões lógicas, tais como:

p

p ^ q

p → q

Mas por enquanto não se preocupe com tais expressões, pois vamos ver cada um dos conectivos (negação, conjunção etc.) utilizados nas expressões lógicas a seguir.

Negação

A negação de uma proposição pode ser representada por "~", "¬", " ¯ " (um risco acima da letra) ou "NOT".

A negação simplesmente inverte o valor definido pela proposição. Ex.:

p: "Gosto de jogar bola."

~p: "Não é verdade que gosto de jogar bola."

Uma maneira de representar as possibilidades é através da seguinte Tabela Verdade:

p	~p
V	F
F	V

Conjunção (E)

A conjunção ocorre com o conectivo lógico "E". Para que a proposição composta seja verdadeira, ambas proposições também devem ser! Também pode ser representado por "^", "." ou "AND". Por exemplo:

p: "Gosto de jogar bola"

q: "Gosto de estudar";

p ^ q: "Gosto de jogar bola e de estudar";

Pode ser verdade que "gosto de jogar bola", mas se for falso que "gosto de estudar", não é possível afirmar que é verdade que "Gosto de jogar bola e de estudar"!

Vejamos a Tabela Verdade:

p	q	p ^ q
V	V	V
V	F	F
F	V	F
F	F	F

Disjunção (OU)

A disjunção ocorre com o conectivo lógico "OU". Para que a proposição composta seja verdadeira, basta que uma das proposições seja V! Também pode ser representado por "v", "+" ou "OR". Por exemplo:

p: "Gosto de jogar bola";

q: "Gosto de estudar";

p v q: "Gosto de jogar bola ou de estudar";

Vejamos a Tabela Verdade:

p	q	p v q
V	V	V
V	F	V
F	V	V
F	F	F

Disjunção Exclusiva (XOR)

A disjunção exclusiva ocorre com o conectivo lógico "XOR" (eXclusive OR). Para que a proposição composta seja verdadeira, apenas uma deve ser verdadeira e a outra deve ser falsa! Note que agora queremos EXCLUSIVIDADE, então apenas UMA proposição deve ser V! Também pode ser representado "$\underline{v}$", "XOR" ou "$\oplus$". Por exemplo:

p: "Gosto de jogar bola";

q: "Gosto de estudar";

p $\underline{v}$ q: "Ou gosto de jogar bola, ou gosto de estudar";

Vejamos a Tabela Verdade:

p	q	p $\underline{v}$ q
V	V	F
V	F	V
F	V	V
F	F	F

Condicional (Se, Então)

Uma condicional (implicação lógica) é uma combinação do tipo "se p, então q" (representada por p → q). Usando o nosso exemplo clássico, podemos montar o seguinte:

p: "Gosto de jogar bola"

q: "Gosto de estudar"

p → q: "Se gosto de jogar bola, então gosto de estudar"

Por que chamamos de condicional? Porque envolve uma condição ("se gosto de jogar bola") que, caso ocorra, faz com que automaticamente a sua consequência ("gosto de estudar") ocorra. Ou seja, se "p" for Verdadeira, há a obrigação de "q" também ser Verdadeira!

Se a condição p não ocorrer (valor Falso), q pode ocorrer (V) ou não (F), e ainda assim a frase é verdadeira. Muita atenção aqui!!!

Mas se a condição ocorrer (p é V) e o resultado não ocorrer (q é F), estamos diante de uma proposição composta que é falsa como um todo. Tudo o que vimos leva à seguinte Tabela Verdade:

p	q	p → q
V	V	V
V	F	F
F	V	V
F	F	V

Bicondicional (Se e Somente Se)

Uma bicondicional é uma combinação do tipo "p se e somente se q" (representada por p ↔ q). Indo direto ao nosso exemplo:

p: "Gosto de jogar bola"

q: "Gosto de estudar"

p ↔ q: "Gosto de jogar bola se e somente se gosto de estudar"

Analisando a frase acima, vemos que as duas coisas acontecem juntas ou então nenhuma delas acontece! Ou seja, sabendo que "gosto de jogar bola", já sabemos que "gosto de estudar". Da mesma forma, sabendo que "gosto de estudar", então sabemos que "gosto de jogar bola". Também podemos verificar que se não gosto de jogar bola, automaticamente não gosto de estudar!

Para resumir, a expressão p ↔ q só é verdadeira quando tanto p quanto q acontecem (ambas são Verdadeiras), ou então quando ambas não acontecem (Falsas). Se ocorrer qualquer outra combinação, a expressão p ↔ q será falsa. Portanto, temos a seguinte Tabela Verdade:

p	q	p ↔ q
V	V	V
V	F	F
F	V	F
F	F	V

CURIOSIDADE

Às vezes é cobrado em uma prova de concurso a quantidade de linhas de uma tabela verdade. Então eu pergunto a você: Quantas linhas existem em uma tabela verdade?

Já vimos exemplos de 2 e de 4 linhas, sendo 2 quando há uma proposição apenas (no caso da negação) e 4 quando há duas proposições (vimos nos demais casos, até agora). Claro que poderíamos ter três preposições (ou mais), como por exemplo:

p ^ q ^ r

No caso acima teríamos 8 linhas! Mas como saber disso? A fórmula é a seguinte:

2^n, onde n é o número de proposições. Ex.: $2^1 = 2$, $2^2 = 4$, $2^3 = 8$, ...

Ah, mas e se eu for péssimo em memorizar fórmulas? Bom, aí você pode montar a tabela na hora, com todas combinações de V e F, o que pode demorar um pouco!

Tautologias, Contradições e Contingências

Uma tautologia ocorre quando a expressão lógica é sempre verdadeira, independente dos valores lógicos das proposições simples que a compõem. Para verificarmos um exemplo veremos a seguir uma proposição composta e sua Tabela Verdade:

Tautologia

- ocorre quando a expressão lógica é sempre verdadeira

"Gosto de estudar ou não gosto de estudar"

Vamos decompor as proposições, para um melhor entendimento:

p: "Gosto de estudar"

~p: "não gosto de estudar"

A Tabela Verdade é mostrada a seguir, com um destaque na coluna que representa "Gosto de estudar ou não gosto de estudar". Portanto, podemos ver que se trata de uma tautologia!

p	~p	p v ~p
V	F	V
F	V	V

Uma contradição ocorre quando a expressão sempre é falsa, independente dos valores lógicos das proposições que a compõem. Vamos ver um exemplo:

"Gosto de estudar e não gosto de estudar"

Vamos decompor as proposições, para um melhor entendimento:

p: "Gosto de estudar"

~p: "não gosto de estudar"

A Tabela Verdade é mostrada a seguir, com um destaque na coluna que representa "Gosto de estudar e não gosto de estudar". Portanto, podemos ver que se trata de uma contradição!

p	~p	p ^ ~p
V	F	F
F	V	F

Uma contingência ocorre quando a expressão pode ser V ou F, dependendo da combinação de valores lógicos das proposições que a compõem (não sendo tudo V ou tudo F, claro). Vamos ver um exemplo:

Ex.: "Gosto de estudar e de viajar"

Vamos decompor as proposições, para um melhor entendimento:

p: "Gosto de estudar"

q: "gosto de viajar"

A Tabela Verdade é mostrada a seguir, com um destaque na coluna que representa "Gosto de estudar e de viajar". Podemos ver que se trata de uma contingência!

p	q	p ^ q
V	V	V
V	F	F
F	V	F
F	F	F

Equivalência Lógica

Duas proposições lógicas são equivalentes quando elas possuem a mesma Tabela Verdade. Só isso? Sim! Então vamos praticar...as duas proposições abaixo são equivalentes?

p → q: "Se gosto de jogar bola, então gosto de estudar"

~p v q: "Não gosto de jogar bola ou gosto de estudar"

Vamos criar as duas tabelas (passo a passo) e comparar se a última coluna de cada uma possui os mesmos valores lógicos (V ou F):

p	q	p → q
V	V	V
V	F	F
F	V	V
F	F	V

p	~p	q	~p v q
V	F	V	V
V	F	F	F
F	V	V	V
F	V	F	V

Como podemos ver, os valores lógicos em todas as linhas para as proposições "p → q" e "~p v q" são idênticos, então estamos diante de uma equivalência lógica!

1. (FCC/TRE-SP - 2017) Considere que uma expressão lógica envolva candidato (C), cargo político (P), votos (V) e ganhador (G). Para avaliar se uma dada expressão é verdadeira ou não, um Técnico deve usar uma Tabela da Verdade, que contém uma lista exaustiva de situações possíveis envolvendo as 4 variáveis. A Tabela da Verdade deve ter 4 colunas e

A) 8 linhas.

B) 16 linhas.

C) 4 linhas.

D) 32 linhas.

E) 64 linhas.

Comentários:

Lembrando a fórmula... $2^n = 2^4 = 16$ linhas!

Portanto, a alternativa B está correta e é o gabarito da questão.

Gabarito: Letra B

2. (NUCEPE/SEDUC-PI - 2015) Operadores lógicos servem para combinar resultados de expressões, cujo resultado será verdadeiro ou falso. Assinale a alternativa CORRETA conforme as expressões abaixo:

A) Verdadeiro AND Verdadeiro = Falso.

B) Verdadeiro AND Falso = Verdadeiro.

C) Verdadeiro OR Verdadeiro = Falso.

D) Falso OR Verdadeiro = Verdadeiro.

E) Verdadeiro NOT = Verdadeiro.

Comentários:

A alternativa E nem existe! O "NOT" deveria vir antes. As demais são mostradas abaixo.

A) Verdadeiro AND Verdadeiro = V.

B) Verdadeiro AND Falso = F.

C) Verdadeiro OR Verdadeiro = V.

D) Falso OR Verdadeiro = V.

Portanto, a alternativa D está correta e é o gabarito da questão.

Gabarito: Letra D

QUESTÕES COMENTADAS - PROPOSIÇÕES X TABELA VERDADE - MULTIBANCAS

1. (FUNCAB/DER-RO - 2010) Considere as seguintes sentenças de lógica proposicional.

I. $(((P \vee Q) \rightarrow \sim R) \rightarrow (Q \& P))$
II. $(X \& (W \rightarrow ((P \& W) \vee (S \leftrightarrow T))$
III. $(P \& (R \rightarrow ((P \& P) \vee (P \rightarrow P))))$

A opção que contém os respectivos números de linhas da tabela verdade de cada uma das sentenças é:

A) I tem 8 linhas na tabela verdade

II tem 32 linhas na tabela verdade

III tem 4 linhas na tabela verdade.

B) I tem 16 linhas na tabela verdade;

II tem 64 linhas na tabela verdade,

III tem 4 linhas na tabela verdade.

C) I tem 32 linhas na tabela verdade

II tem 32 linhas na tabela verdade

III tem 8 linhas na tabela verdade.

D) I tem 16 linhas na tabela verdade

II tem 16 linhas na tabela verdade

III tem 8 linhas na tabela verdade.

E) I tem 8 linhas na tabela verdade

II tem16 linhas na tabela verdade

III tem 16 linhas na tabela verdade.

Comentários:

Esse tipo de questão ficaria inviável caso você não se lembre da fórmula 2^n, porque montar cada uma das tabelas levaria muito tempo! Vamos ver cada uma delas:

I. Existem três proposições simples (P, Q, R), logo $2^3 = 8$ linhas;

II. Existem cinco proposições simples (X, W, P, S, T), logo $2^5 = 32$ linhas;

III. Existem duas proposições simples (P, R), logo $2^2 = 4$ linhas.

Portanto, a alternativa A está correta e é o gabarito da questão.

Gabarito: Letra A

2. (FAURGS/UFRGS - 2013) A tabela verdade de 2 variáveis A e B é mostrada a seguir.

A	B	Operador
F	F	F
F	V	V
V	F	V
V	V	F

O operador representado na tabela verdade só resulta em verdadeiro quando as duas variáveis tiverem valores lógicos diferentes entre si. Esse operador é conhecido como

A) OR.

B) AND.

C) NOT.

D) XOR.

E) XAND.

Comentários:

Nem precisa tentar um por um dos operadores lógicos para ver qual deles se "encaixa" com as respostas mostradas na tabela! Sabemos que o operador que exige "exclusividade", ou seja, APENAS um deve ser V e o outro deve ser F, para que o "resultado" seja V é o OU EXCLUSIVO (XOR). Portanto, a alternativa D está correta e é o gabarito da questão.

Gabarito: Letra D

3. (CESPE/SERPRO - 2013) Considerando as variáveis A = 3, B = 5 e C = 2, julgue os itens que se seguem.

O resultado da expressão NOT A <= B é verdadeiro.

Comentários:

Vamos resolver passo a passo, com a substituição das variáveis com os valores (imagine que o NOT seja aplicado ao resto, como se existissem parênteses):

NOT (3 <= 5)

NOT (V)

F

O resultado deveria ser FALSO, mas a banca deu como verdadeiro! Portanto, a questão está errada.

Gabarito: Errada

4. (CESPE/SERPRO - 2013) Considerando as variáveis A = 3, B = 5 e C = 2, julgue os itens que se seguem.

O resultado da expressão (A + B) < (B + C) AND B = (C + A) é verdadeiro.

Comentários:

Vamos resolver passo a passo, com a substituição das variáveis com os valores (imagine que o NOT seja aplicado ao resto, como se existissem parênteses):

(3 + 5) < (5 + 2) AND 5 = (2 + 3)

8 < 7 AND 5 = 5

F AND V

F

O resultado deveria ser FALSO! Portanto, a questão está errada.

Gabarito: Errada

5. (FGV/AL-MA - 2013) Dado que and = E lógico, or = ou lógico, not = negação lógica, ∧ = ou exclusivo, True = verdadeiro e False = falso, assinale a alternativa que apresenta a expressão lógica que resulta em True.

A) (((True and False) or True)∧((True and False) or (not False)))

B) (((not (False and True)) or False)∧((False or True) and True))

C) (((True and False) or True)∧((True and False) and (not False)))

D) not True

E) (((False and True) or False)∧((False and True) and (not True)))

Comentários:

Cuidado com a pegadinha! Estamos acostumados a ver o ^ como o operador lógico E e não como o operador XOR! Então vamos reescrever e analisar cada uma das alternativas:

A) (((True and False) or True) XOR ((True and False) or (not False)))

(((FALSE) or TRUE) XOR ((FALSE) or (TRUE)))

(TRUE XOR TRUE)

FALSE

B) (((not (False and True)) or False) XOR ((False or True) and True))

(((not FALSE) or False) XOR (TRUE and True))

((TRUE or False) XOR (TRUE and True))

(TRUE XOR TRUE)

FALSE

C) (((True and False) or True) XOR ((True and False) and (not False)))

((FALSE or True) XOR (FALSE and TRUE))

(TRUE XOR FALSE)

TRUE

D) not True

FALSE

E) (((False and True) or False) XOR ((False and True) and (not True)))

((FALSE or False) XOR (FALSE and FALSE))

(FALSE XOR FALSE)

FALSE

Portanto, a alternativa C está correta e é o gabarito da questão.

Gabarito: Letra C

6. (IBFC/TRE-AM - 2014) Ao usarmos o operador lógico OR, com variáveis binárias A e B, teremos a tabela abaixo. Identifique a alternativa que apresenta os resultados da terceira coluna (de cima para baixo):

A	B	A OR B
0	0	?
0	1	?
1	0	?
1	1	?

A) 0 - 0 - 0 - 1

B) 0 - 1 - 1 - 1

C) 1 - 0 - 0 - 0

D) 0 - 1 - 1 - 0

Comentários:

Quando usamos o operador lógico "OU", basta que uma proposição seja V. No caso em tela, como são duas (A, B), se houver uma ou duas com o valor lógico V, o "A OU B" será V.

Então teremos 0 – 1 – 1 – 1. Portanto, a alternativa B está correta e é o gabarito da questão.

Gabarito: Letra B

7. (NUCEPE/SEDUC-PI - 2015) Operadores lógicos servem para combinar resultados de expressões, cujo resultado será verdadeiro ou falso. Assinale a alternativa CORRETA conforme as expressões abaixo:

A) Verdadeiro AND Verdadeiro = Falso.

B) Verdadeiro AND Falso = Verdadeiro.

C) Verdadeiro OR Verdadeiro = Falso.

D) Falso OR Verdadeiro = Verdadeiro.

E) Verdadeiro NOT = Verdadeiro.

Comentários:

A alternativa E nem existe! O "NOT" deveria vir antes. As demais são mostradas abaixo.

A) Verdadeiro AND Verdadeiro = V.

B) Verdadeiro AND Falso = F.

C) Verdadeiro OR Verdadeiro = V.

D) Falso OR Verdadeiro = V.

Portanto, a alternativa D está correta e é o gabarito da questão.

Gabarito: Letra D

8. (INSTITUTO AOCP/CASAN - 2016) A tabela verdade apresenta os estados lógicos das entradas e das saídas de um dado no computador. Ela é a base para a lógica binária que, igualmente, é a base de todo o cálculo computacional. Sabendo disso, assinale a alternativa que apresenta a fórmula que corresponde ao resultado da tabela verdade dada.

p	q	resultado
V	V	V
V	F	F
F	V	F
F	F	F

A) (p ^ q)

B) (p v q)

C) (p → q)

D) (¬ p)

E) (¬ q)

Comentários:

Minha dica é a seguinte: veja que na coluna do resultado o único V aparece quando ambas proposições são V. Sabemos que isso ocorre com o conectivo "E". No dia da prova fica mais fácil assim, mas sugiro que você monte as tabelas para praticar. Portanto, a alternativa A está correta e é o gabarito da questão.

Gabarito: Letra A

9. (FGV/COMPESA - 2016) Considere a expressão lógica A → B, lida como "se A é verdadeiro então B é verdadeiro".

Dado que A e B são expressões lógicas, assinale a opção que indica uma expressão lógica equivalente à referida expressão.

A) ~A or B

B) ~A or ~B

C) ~A and B

D) ~A and ~B

E) ~(A or B)

Comentários:

Para encontrar uma expressão lógica equivalente temos que encontrar a mesma tabela verdade! Primeiro vamos verificar a tabela verdade de "A → B":

A	B	A → B
V	V	V
V	F	F
F	V	V
F	F	V

Agora vamos ver a tabela de "~A or B" (alternativa A):

A	~A	B	~A or B
V	F	V	V
V	F	F	F
F	V	V	V
F	V	F	V

De primeira já encontramos a resposta! Mas sugiro que você crie as demais tabelas para praticar!

Portanto, a alternativa A está correta e é o gabarito da questão.

Gabarito: Letra A

10. (UFV/UFV-MG - 2017) Observe a expressão lógica abaixo:

((((true AND true) OR false) AND true) AND (true OR (true AND false)))

Considerando os operadores lógicos AND (e) e OR (ou), e os operandos lógicos true (verdadeiro) e false (falso), é CORRETO afirmar que o valor lógico dessa expressão é:

A) verdadeiro.

B) falso.

C) indefinido.

D) nulo.

Comentários:

Vamos analisar passo a passo:

((((true AND true) OR false) AND true) AND (true OR (true AND false)))

(((TRUE OR false) AND true) AND (true OR FALSE))

((TRUE AND true) AND (true OR FALSE))

(TRUE AND TRUE)

TRUE

Portanto, a alternativa A está correta e é o gabarito da questão.

Gabarito: Letra A

11. (FCC/TRE-SP - 2017) Considere que uma expressão lógica envolva candidato (C), cargo político (P), votos (V) e ganhador (G). Para avaliar se uma dada expressão é verdadeira ou não, um Técnico deve usar uma Tabela da Verdade, que contém uma lista exaustiva de situações possíveis envolvendo as 4 variáveis. A Tabela da Verdade deve ter 4 colunas e

A) 8 linhas.

B) 16 linhas.

C) 4 linhas.

D) 32 linhas.

E) 64 linhas.

Comentários:

Lembrando a fórmula... $2^n = 2^4 = 16$ linhas!

Portanto, a alternativa B está correta e é o gabarito da questão.

Gabarito: Letra B

12. (CEPS-UFPA/UNIFESSPA - 2018) Considere a seguinte tabela verdade:

A	B	Resultado
V	V	V
V	F	F
F	V	F
F	F	F

O operador que representa corretamente essa tabela verdade é

A) E.

B) OU.

C) identidade.

D) inverso.

E) negação.

Comentários:

Já vimos o mesmo cenário há pouco, heim! Então vou repetir o comentário:

Minha dica é a seguinte: veja que na coluna do resultado o único V aparece quando ambas proposições são V. Sabemos que isso ocorre com o conectivo "E". No dia da prova fica mais fácil assim, mas sugiro que você monte as tabelas para praticar.

Portanto, a alternativa A está correta e é o gabarito da questão.

Gabarito: Letra A

13. (IDECAN/CRF-SP - 2018) Utilizando o operador lógico "e", a tabela-verdade a seguir terá sua equivalência completada na ordem:

A	B	C	A e B e C
V	V	F	
F	V	F	
V	F	V	
V	V	V	
F	V	V	

A) V, V, V, V, V.

B) F, F, F, F, F.

C) V, V, V, F, V.

D) F, F, F, V, F.

Comentários:

Podemos ver que gostam mesmo do conectivo "E"! Vamos lá...para o resultado ser V, todas as proposições devem ser V! Nessa questão são três proposições (A, B, C), então apenas uma linha terá como resultado V (a 4ª linha). A tabela completa teria 8 linhas (23), mas apenas 5 foram mostradas. Assim, teremos na coluna de resultado a sequência F – F – F – V – F. Portanto, a alternativa D está correta e é o gabarito da questão.

▪

Gabarito: Letra D

14. (IBADE/CAERN - 2018) Quanto ao preenchimento da tabela verdade, marque a alternativa que apresente os valores lógicos para as letras a, b, c e d.

P	T	P ^ T	(P ^ T) v T
V	V	*a*	V
F	F	F	*b*
V	F	*c*	F
F	V	F	*d*

A) V, V, V e V

B) V, F, F e V

C) F, F, F e F

D) V, F, V e F

E) F, V, V e F

Comentários:

Vamos ver cada uma das letras mostradas na tabela:

a = P ^ T = V ^ V = V

b = (P ^ T) v T = F v F = F

c = P ^ T = V ^ F = F

d = (P ^ T) v T = F v V = V

Portanto, a alternativa B está correta e é o gabarito da questão.

Gabarito: Letra B

15. (COSEAC/UFF - 2019) Considere a tabela da verdade abaixo. A expressão que executa a tabela é:

A	B	C	S
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

A) $S = A\,\overline{B} + \overline{C}$
B) $S = \overline{A\,B + C}$
C) $S = A\,B + \overline{C}$
D) $S = \overline{A\,B} + C$
E) $S = A\,B + C$

Comentários:

Vamos reescrever a expressão lógica, começando pela alternativa (A) e lembrando que 0 é F e 1 é V:

(A ^ ~B) v ~C

Linha 1: (0 ^ ~0) v ~0

(0 ^ 1) v 1

0 v 1 = 1

Linha 2: (0 ^ ~0) v ~1

(0 ^ 1) v 0

0 v 0 = 0

Linha 3: (0 ^ ~1) v ~0

(0 ^ 0) v 1

0 v 1 = 1

Linha 4: (0 ^ ~1) v ~1

(0 ^ 0) v 0

0 v 0 = 0

Linha 5: (1 ^ ~0) v ~0

(1 ^ 1) v 1

1 v 1 = 1

Linha 6: (1 ^ ~0) v ~1

(1 ^ 1) v 0

1 v 0 = 1

Linha 7: (1 ^ ~1) v ~0

(1 ^ 0) v 1

0 v 1 = 1

Linha 8: (1 ^ ~1) v ~1

(1 ^ 0) v 0

0 v 0 = 0

Na primeira já encontramos o resultado! Questão que exige atenção! Sugiro que você faça as outras quatro para treinar, mas no dia da prova sugiro que deixe esse tipo de questão por último e só faça as demais alternativas se tiver sobrado tempo, para conferir.

Portanto, a alternativa A está correta e é o gabarito da questão.

Gabarito: Letra A

16. (IADES/CAU-AC - 2019) Para construir a tabela verdade da proposição ~(p v ~q), um estudante montou o quadro apresentado.

p	q	~ q	p V ~q	~(p V ~q)
V	V			
V	F			
F	V			
F	F			

Ao se preencher completamente e corretamente a tabela, o número de F encontrado na última coluna é igual a

A) 1.

B) 3.

C) 4.

D) 0.

E) 2.

Comentários:

Não tem outra forma de fazer a questão...vamos preencher a tabela verdade:

p	q	~q	p v ~q	~(p v ~q)
V	V	F	V	F
V	F	V	V	F
F	V	F	F	V
F	F	V	V	F

Portanto, a alternativa B está correta e é o gabarito da questão.

Gabarito: Letra B

LISTA DE QUESTÕES - PROPOSIÇÕES X TABELA VERDADE - MULTIBANCAS

1. (FUNCAB/DER-RO - 2010) Considere as seguintes sentenças de lógica proposicional.

I. $(((P \vee Q) \rightarrow \sim R) \rightarrow (Q \& P))$
II. $(X \& (W \rightarrow ((P \& W) \vee (S \leftrightarrow T))$
III. $(P \& (R \rightarrow ((P \& P) \vee (P \rightarrow P))))$

A opção que contém os respectivos números de linhas da tabela verdade de cada uma das sentenças é:

A) I tem 8 linhas na tabela verdade

II tem 32 linhas na tabela verdade

III tem 4 linhas na tabela verdade.

B) I tem 16 linhas na tabela verdade;

II tem 64 linhas na tabela verdade,

III tem 4 linhas na tabela verdade.

C) I tem 32 linhas na tabela verdade

II tem 32 linhas na tabela verdade

III tem 8 linhas na tabela verdade.

D) I tem 16 linhas na tabela verdade

II tem 16 linhas na tabela verdade

III tem 8 linhas na tabela verdade.

E) I tem 8 linhas na tabela verdade

II tem16 linhas na tabela verdade

III tem 16 linhas na tabela verdade.

2. (FAURGS/UFRGS - 2013) A tabela verdade de 2 variáveis A e B é mostrada a seguir.

A	B	Operador
F	F	F
F	V	V
V	F	V
V	V	F

O operador representado na tabela verdade só resulta em verdadeiro quando as duas variáveis tiverem valores lógicos diferentes entre si. Esse operador é conhecido como

A) OR.

B) AND.

C) NOT.

D) XOR.

E) XAND.

3. (CESPE/SERPRO - 2013) Considerando as variáveis A = 3, B = 5 e C = 2, julgue os itens que se seguem.

   O resultado da expressão NOT A <= B é verdadeiro.

4. (CESPE/SERPRO - 2013) Considerando as variáveis A = 3, B = 5 e C = 2, julgue os itens que se seguem.

   O resultado da expressão (A + B) < (B + C) AND B = (C + A) é verdadeiro.

5. (FGV/AL-MA - 2013) Dado que and = E lógico, or = ou lógico, not = negação lógica, ∧ = ou exclusivo, True = verdadeiro e False = falso, assinale a alternativa que apresenta a expressão lógica que resulta em True.

A) (((True and False) or True)∧((True and False) or (not False)))

B) (((not (False and True)) or False)∧((False or True) and True))

C) (((True and False) or True)∧((True and False) and (not False)))

D) not True

E) (((False and True) or False)∧((False and True) and (not True)))

6. (IBFC/TRE-AM - 2014) Ao usarmos o operador lógico OR, com variáveis binárias A e B, teremos a tabela abaixo. Identifique a alternativa que apresenta os resultados da terceira coluna (de cima para baixo):

A	B	A OR B
0	0	?
0	1	?
1	0	?
1	1	?

A) 0 - 0 - 0 - 1

B) 0 - 1 - 1 - 1

C) 1 - 0 - 0 - 0

D) 0 - 1 - 1 - 0

7. (NUCEPE/SEDUC-PI - 2015) Operadores lógicos servem para combinar resultados de expressões, cujo resultado será verdadeiro ou falso. Assinale a alternativa CORRETA conforme as expressões abaixo:

A) Verdadeiro AND Verdadeiro = Falso.

B) Verdadeiro AND Falso = Verdadeiro.

C) Verdadeiro OR Verdadeiro = Falso.

D) Falso OR Verdadeiro = Verdadeiro.

E) Verdadeiro NOT = Verdadeiro.

8. (INSTITUTO AOCP/CASAN - 2016) A tabela verdade apresenta os estados lógicos das entradas e das saídas de um dado no computador. Ela é a base para a lógica binária que, igualmente, é a base de todo o cálculo computacional. Sabendo disso, assinale a alternativa que apresenta a fórmula que corresponde ao resultado da tabela verdade dada.

p	q	resultado
V	V	V
V	F	F
F	V	F
F	F	F

A) (p ^ q)

B) (p v q)

C) (p → q)

D) (¬ p)

E) (¬ q)

9. (FGV/COMPESA - 2016) Considere a expressão lógica A → B, lida como "se A é verdadeiro então B é verdadeiro".

Dado que A e B são expressões lógicas, assinale a opção que indica uma expressão lógica equivalente à referida expressão.

A) ~A or B

B) ~A or ~B

C) ~A and B

D) ~A and ~B

E) ~(A or B)

10. (UFV/UFV-MG - 2017) Observe a expressão lógica abaixo:

((((true AND true) OR false) AND true) AND (true OR (true AND false)))

Considerando os operadores lógicos AND (e) e OR (ou), e os operandos lógicos true (verdadeiro) e false (falso), é CORRETO afirmar que o valor lógico dessa expressão é:

A) verdadeiro.

B) falso.

C) indefinido.

D) nulo.

11. (FCC/TRE-SP - 2017) Considere que uma expressão lógica envolva candidato (C), cargo político (P), votos (V) e ganhador (G). Para avaliar se uma dada expressão é verdadeira ou não, um Técnico deve usar uma Tabela da Verdade, que contém uma lista exaustiva de situações possíveis envolvendo as 4 variáveis. A Tabela da Verdade deve ter 4 colunas e

A) 8 linhas.

B) 16 linhas.

C) 4 linhas.

D) 32 linhas.

E) 64 linhas.

12. (CEPS-UFPA/UNIFESSPA - 2018) Considere a seguinte tabela verdade:

A	B	Resultado
V	V	V
V	F	F
F	V	F
F	F	F

O operador que representa corretamente essa tabela verdade é

A) E.

B) OU.

C) identidade.

D) inverso.

E) negação.

13. (IDECAN/CRF-SP - 2018) Utilizando o operador lógico "e", a tabela-verdade a seguir terá sua equivalência completada na ordem:

A	B	C	A e B e C
V	V	F	
F	V	F	
V	F	V	
V	V	V	
F	V	V	

A) V, V, V, V, V.

B) F, F, F, F, F.

C) V, V, V, F, V.

D) F, F, F, V, F.

14. (IBADE/CAERN - 2018) Quanto ao preenchimento da tabela verdade, marque a alternativa que apresente os valores lógicos para as letras a, b, c e d.

P	T	P ^ T	(P ^ T) v T
V	V	*a*	V
F	F	F	*b*
V	F	*c*	F
F	V	F	*d*

A) V, V, V e V

B) V, F, F e V

C) F, F, F e F

D) V, F, V e F

E) F, V, V e F

15. (COSEAC/UFF - 2019) Considere a tabela da verdade abaixo. A expressão que executa a tabela é:

A	B	C	S
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

A) $S = A\,\overline{B} + \overline{C}$
B) $S = \overline{A\,B + C}$
C) $S = A\,B + \overline{C}$
D) $S = \overline{A\,B} + C$
E) $S = A\,B + C$

16. (IADES/CAU-AC - 2019) Para construir a tabela verdade da proposição ~(p v ~q), um estudante montou o quadro apresentado.

p	q	~ q	p V ~q	~(p V ~q)
V	V			
V	F			
F	V			
F	F			

Ao se preencher completamente e corretamente a tabela, o número de F encontrado na última coluna é igual a

A) 1.

B) 3.

C) 4.

D) 0.

E) 2.

GABARITO

GABARITO

1- A	2- D	
3- Errada	8- A	13- D
4- Errada	9- A	14- B
5- C	10- A	15- A
6- B	11- B	16- B
7- D	12- A	

ÁLGEBRA BOOLEANA E CIRCUITOS LÓGICOS

Nesta parte da aula veremos que os circuitos lógicos nada mais são do que colocar "na prática", no circuito integrado, aquilo que é estudado na álgebra booleana! E, depois de estudar as proposições e tabela verdade, fica muito mais tranquilo entendermos tudo isso! Vamos lá...

Álgebra Booleana

Uma álgebra booleana é uma 6-upla $(X, \vee, \wedge, \neg, 0, 1)$ consistindo em um conjunto X munido de duas operações binárias e uma operação unária. Abaixo vemos as duas operações binárias.

- v (também denotado por +), geralmente chamado de "ou";
- ^ (também denotado por * ou .), geralmente chamado de "e".

A operação unária $\neg$, também denotada por ~ ou por uma barra superior, é geralmente chamado de "não".

Por fim, as duas constantes:

- 0, também denotada por F, geralmente chamado de "zero" ou de "falso" (*false*);
- 1, também denotada por T (*true*) ou V, geralmente chamado de "um" ou de "verdadeiro".

Tudo isso deve satisfazer os seguintes axiomas[1], para quaisquer $a, b, c \in X$:

Propriedades Associativas	$(a \vee b) \vee c = a \vee (b \vee c)$	$(a \wedge b) \wedge c = a \wedge (b \wedge c)$
Propriedades Comutativas	$a \vee b = b \vee a$	$a \wedge b = b \wedge a$
Propriedades Absortivas	$a \wedge (a \vee b) = a$	$a \vee (a \wedge b) = a$
Propriedades Distributivas	$a \vee (b \wedge c) = (a \vee b) \wedge (a \vee c)$	$a \wedge (b \vee c) = (a \wedge b) \vee (a \wedge c)$
Elementos Neutros	$a \vee 0 = a$	$a \wedge 1 = a$
Elementos Complementares	$a \vee \neg a = 1$	$a \wedge \neg a = 0$

Não se assuste com esses símbolos todos! É só utilizar o + o lugar do "ou" e a * no lugar do "e", e lembrar de como funciona na matemática com a adição e a multiplicação!

Nos elementos neutros, "qualquer coisa" ou 0 tem como resultado "qualquer coisa". Ou seja, tem como resultado o valor de a. Nos elementos complementares, "qualquer coisa" ou a negação dela

[1] Axioma: sentença ou proposição que não é provada ou demonstrada e é considerada como óbvia ou como um consenso inicial necessário para a construção ou aceitação de uma teoria. Por essa razão, é aceito como verdade e serve como ponto inicial para dedução de outras verdades (dependentes de teoria).

tem como resultado verdadeiro, sempre! E "qualquer coisa" e a negação dela tem como resultado falso, sempre!

Alguns teoremas importantes são mostrados na sequência. Dado uma álgebra booleana sobre X, são válidos para quaisquer $a, b \in X$:

- Propriedades Idempotentes
  $a \vee a = a$
  $a \wedge a = a$
  Exemplos:
  a = 0: 0 v 0 = 0
  a = 1: 1 ^ 1 = 1

- Dupla Negação
  $\neg(\neg a) = a$
  Exemplo:
  a = 1: ~(~1) = ~(0) = 1

- Leis de De Morgan
  $\neg(a \vee b) = \neg a \wedge \neg b$
  $\neg(a \wedge b) = \neg a \vee \neg b$
  Exemplos:
  a = 1, b = 0: ~(1 v 0) = ~1 ^ ~0 = 0 ^ 1 = 0
  a = 1, b = 0: ~(1 ^ 0) = ~1 v ~0 = 0 v 1 = 1

- Leis de Absorção
  $a \vee (a \wedge b) = a$
  $a \wedge (a \vee b) = a$
  Exemplos:
  a = 1, b = 0: 1 v (1 ^ 0) = 1
  a = 1, b = 0: 1 ^ (1 v 0) = 1

- Elementos Absorventes
  $a \vee 1 = 1$
  $a \wedge 0 = 0$
  Exemplos:
  a = 1: 1 v 1 = 1
  a = 1: 1 ^ 0 = 0

- Negações do Zero e do Um

$$\neg 0 = 1$$
$$\neg 1 = 0$$

- Definições alternativas da operação binária XOR ("ou exclusivo")

$$a \veebar b = (a \vee b) \wedge (\neg a \vee \neg b)$$
$$a \veebar b = (a \wedge \neg b) \vee (\neg a \wedge b)$$

Exemplos:
a = 0, b = 0: 0 $\oplus$ 0 = (0 v 0) ^ (~0 v ~0) = (0 v 0) ^ (1 v 1) = 0 ^ 1 = 0
a = 0, b = 0: 0 $\oplus$ 0 = (0 ^ ~0) v (~0 ^ 0) = (0 ^ 1) v (1 ^ 0) = 0 v 0 = 0

Circuitos Lógicos

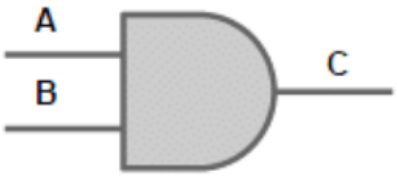
Uma porta lógica é um circuito eletrônico (peça de hardware). A combinação de milhões desses elementos serve para produzir processadores (CPUs), memória principal (RAM), memória cache, interfaces de E/S, entre outros componentes. Para facilitar o entendimento, vamos ver proposições e tabelas verdade, junto com as portas lógicas equivalentes. Entendendo bem a tabela verdade é só associar o "V" ao valor 1 e o "F" ao valor 0. Retomaremos as tabelas verdade para ficar mais fácil o entendimento. Vamos lá...

Começamos pela tabela verdade do "E":

p	q	p ^ q
V	V	V
V	F	F
F	V	F
F	F	F

A porta AND (E) é mostrada abaixo. Podemos ver que temos duas entradas (A, B) e a saída C. Também pode ser representada da seguinte forma:

C = A . B



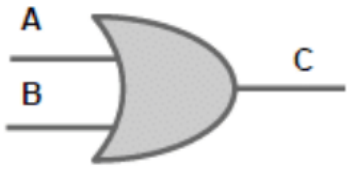
Por exemplo, se A = 1 e B = 1, então C será 1. Se A = 0 e B = 1, então C será 0, e assim por diante.

Agora veremos a tabela verdade do "OU":

p	q	p v q
V	V	V
V	F	V
F	V	V
F	F	F

A porta OU (OR) é mostrada abaixo. Podemos ver que temos duas entradas (A, B) e a saída C. Também pode ser representada da seguinte forma:

$$C = A + B$$



Por exemplo, se A = 1 e B = 0, então C será 1. Se A = 0 e B = 0, então C será 0, e assim por diante.

A tabela verdade do "NOT" é:

p	~p
V	F
F	V

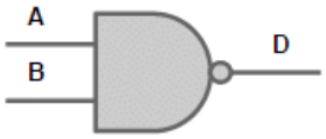
A porta NOT (Negação) é mostrada abaixo. Podemos ver que temos uma entrada (A) e a saída $\bar{A}$. Também pode ser representada da seguinte forma:

$$\bar{A} = \text{NOT A}$$

A saída de NOT (0) é 1 e de NOT (1) é 0.

A porta NAND é a negação do AND, ou seja, aplica-se o AND e depois inverte a saída (se 0, vira 1 e, se 1, vira 0). Pode ser representado assim:

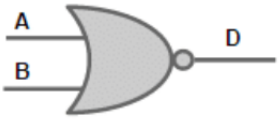
$$D = \overline{A \cdot B}$$



Por exemplo, se A = 1 e B = 1, a saída é 0. Se A = 1 e B = 0, a saída é 1, e assim por diante.

A porta NOR é a negação do OR, ou seja, aplica-se o OR e depois inverte a saída (se 0, vira 1 e, se 1, vira 0). Pode ser representado assim:

$$D = \overline{A + B}$$



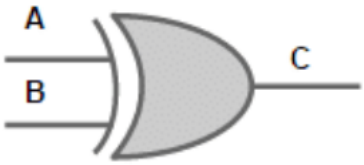
Por exemplo, se A = 1 e B = 0, a saída é 0. Se A = 0 e B = 0, a saída é 1, e assim por diante.

Agora vejamos a tabela verdade do "XOR":

p	q	p v̲ q
V	V	F
V	F	V
F	V	V
F	F	F

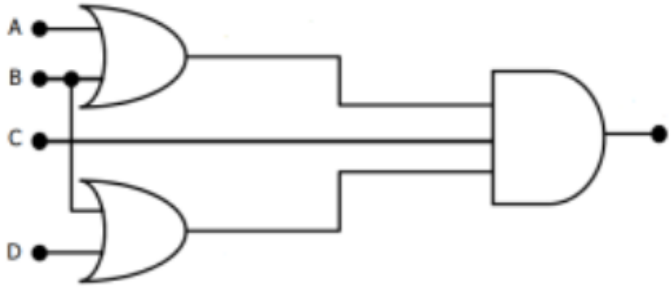
A porta XOR pode ser representada assim:

$$C = A \oplus B$$



Por exemplo, se A = 1 e B = 0, a saída é 1. Se A = 0 e B = 0, a saída é 0, e assim por diante.

Agora vamos juntar algumas portas lógicas para montar um circuito:



[2]

Qual seria a saída para esse circuito? Vamos ver o passo a passo para criar a expressão booleana correspondente.

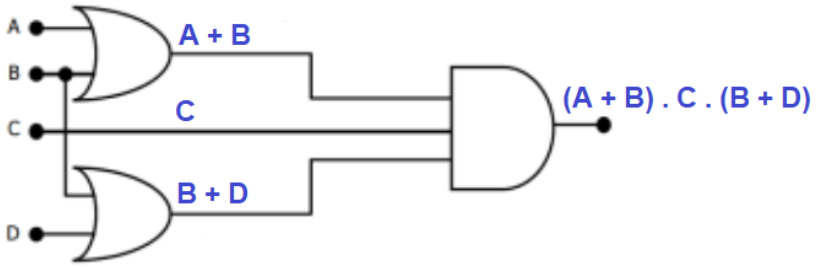
1°) Há uma porta lógica OR, recebendo A, B;

2°) C "passa reto", direto para a porta lógica mais à direita;

3°) Há uma porta lógica OR, recebendo B, D;

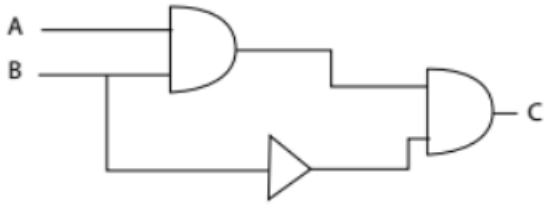
4°) Há uma porta lógica AND recebendo o resultado de 1°, C, resultado de 3°.

Dessa forma, temos como saída (A + B) . C . (B + D), conforme podemos ver abaixo.



1. (UFMG/UFMG - 2019) Dado o circuito lógico representado abaixo e as entradas A e B, defina a expressão representada pela saída C:

[2] Fonte: Concurso para o Colégio Pedro II (2018) – Cargo: Analista de Tecnologia da Informação.



A) A ^ B ^ B

B) A + B + B

C) A + B + ¬B

D) A ^ B ^ ¬B

Comentários:

1°) Porta AND, tendo como entrada A, B. Temos a saída A ^ B.

2°) Porta NOT (mesmo sem a "bolinha", o triângulo "puro" também é a representação para NOT). Temos a saída ¬B.

3°) Porta AND das saídas de 1° e 2°. Temos (A ^ B) ^ ¬B. Como é tudo AND, podemos tirar os parênteses, ficando assim: A ^ B ^ ¬B.

Portanto, a alternativa D está correta e é o gabarito da questão.

Gabarito: Letra D

2. (FGV/AL-RO - 2018) A figura a seguir apresenta a tabela verdade de duas portas lógicas.

PORTA LÓGICA 1		
A	B	S
0	0	0
0	1	1
1	0	1
1	1	1

PORTA LÓGICA 2		
A	B	S
0	0	0
0	1	1
1	0	1
1	1	0

As portas lógicas 1 e 2 são, respectivamente,

A) OR e XOR.

B) OR e NAND.

C) AND e NOR.

D) AND e XOR.

E) NOR e OR.

Comentários:

Na tabela da porta lógica 1 vemos que basta A ou B ter o valor 1 para que S seja 1. Quando A e B são 1, a saída S também é 1. Esse é o comportamento da porta OR.

Na tabela da porta lógica 2 vemos que quando A e B são iguais, a saída é 0, e quando são diferentes a saída é 1. Esse é o comportamento da porta XOR.

Portanto, a alternativa A está correta e é o gabarito da questão.

Gabarito: Letra A

QUESTÕES COMENTADAS - ÁLGEBRA BOOLEANA E CIRCUITOS LÓGICOS - MULTIBANCAS

1. (ESAF/ATRFB - 2006) Analise as seguintes operações relacionadas à Aritmética Computacional, considerando que os valores utilizados estão na representação hexadecimal.

   I. ((2222 AND AAAA) XOR FFFF) = DDDD.

   II. ((2222 OR BBBB) XOR FFFF) = DDDD.

   III. ((2222 NOT CCCC) XOR FFFF) = 3333.

   IV. ((2222 XOR DDDD) XOR FFFF) = 3333.

   Indique a opção que contenha todas as afirmações verdadeiras.

A) I e II

B) II e III

C) III e IV

D) II e IV

E) I e III

Comentários:

Vamos fazer o passo a passo, desde a conversão de hexadecimal para binário (agrupando de 4 em 4 bits, que fica mais fácil). Na sequência aplicamos as operações lógicas e, por fim, convertemos o valor em binário para hexadecimal.

I. ((2222 AND AAAA) XOR FFFF) = DDDD.

0010 0010 0010 0010

AND 1010 1010 1010 1010

0010 0010 0010 0010

XOR 1111 1111 1111 1111

1101 1101 1101 1101 = DDDD → OK!

II. ((2222 OR BBBB) XOR FFFF) = DDDD.

0010 0010 0010 0010

OR 1011 1011 1011 1011

1011 1011 1011 1011

XOR 1111 1111 1111 1111

0100 0100 0100 0100 = 4444 → Não bateu!

III. ((2222 NOT CCCC) XOR FFFF) = 3333.

NOT é uma operação unária! Não existe 2222 NOT CCCC!

IV. ((2222 XOR DDDD) XOR FFFF) = 3333.

0010 0010 0010 0010

XOR 1101 1101 1101 1101

1111 1111 1111 1111

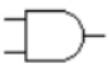
XOR 1111 1111 1111 1111

0000 0000 0000 0000 → Não bateu!

O gabarito da questão é a letra E, ou seja, o que é mostrado em I e III estariam corretas! O problema é que o que é apresentado em III não tem sentido! Talvez tenha ocorrido algum erro de digitação no momento de elaborar o item III! Portanto, a alternativa E é o gabarito da questão, mas deveria ser ANULADA!

Gabarito: Letra E

2. (NC-UFPR/COPEL - 2017) Considere as seguintes portas lógicas: 1. Porta NAND. 2. Porta NOT. 3. Porta XOR. 4. Porta NOR. 5. Porta OR. 6. Porta AND. Com relação às simbologias adotadas para portas lógicas em eletrônica digital, numere os parênteses relacionando as figuras com as respectivas portas.

( )  ( )  ( )  ( )  ( )  ( ) 

Assinale a alternativa que apresenta a numeração correta dos parênteses, da esquerda para a direita

A) 6 – 4 – 2 – 5 – 3 – 1.

B) 6 – 2 – 5 – 3 – 4 – 1.

C) 3 – 1 – 4 – 6 – 5 – 2.

D) 1 – 4 – 2 – 5 – 3 – 6.

E) 1 – 2 – 4 – 3 – 5 – 6.

Comentários:

Essas figuras devem estar bem claras em seu entendimento! Lembrando que a única que tem apenas uma entrada é o NOT, então já sabemos que a terceira figura é o NOT! Ou seja, o que entra 1, sai 0 e vice-versa! Para os outros, vale lembrar que quando tem uma "bolinha" na direita é a negação, ou seja, a porta AND com uma bolinha fica a porta NAND. A porta OR com a bolinha fica a porta NOR. E aquela porta que é igual à porta OR, mas que tem um "risquinho" paralelo é a porta XOR! Analisando as alternativas, sabendo apenas as duas primeiras figuras, já se sabe a resposta! Portanto, a alternativa A está correta e é o gabarito da questão.

Gabarito: Letra A

3. (FGV/AL-RO - 2018) A figura a seguir apresenta a tabela verdade de duas portas lógicas.

PORTA LÓGICA 1		
A	B	S
0	0	0
0	1	1
1	0	1
1	1	1

PORTA LÓGICA 2		
A	B	S
0	0	0
0	1	1
1	0	1
1	1	0

As portas lógicas 1 e 2 são, respectivamente,

A) OR e XOR.

B) OR e NAND.

C) AND e NOR.

D) AND e XOR.

E) NOR e OR.

Comentários:

Na tabela da porta lógica 1 vemos que basta A ou B ter o valor 1 para que S seja 1. Quando A e B são 1, a saída S também é 1. Esse é o comportamento da porta OR.

Na tabela da porta lógica 2 vemos que quando A e B são iguais, a saída é 0, e quando são diferentes a saída é 1. Esse é o comportamento da porta XOR.

Portanto, a alternativa A está correta e é o gabarito da questão.

Gabarito: Letra A

4. (UFGD/UFGD - 2019) Portas ou circuitos lógicos são dispositivos que operam um ou mais sinais lógicos de entrada para produzir uma e somente uma saída, dependente da função implementada no circuito. São geralmente usadas em circuitos eletrônicos, por causa das situações que os sinais deste tipo de circuito podem apresentar: presença de sinal, ou "1"; e ausência de sinal, ou "0".

   Disponível em: https://pt.wikipedia.org/wiki/Porta_l%C3%B3gica. Acesso em: 20 fev. 2019.

   Para a tabela verdade apresentada a seguir, é correto afirmar que, em ordem, as portas lógicas são:

A	B	X
0	0	1
0	1	1
1	0	1
1	1	0

A	B	X
0	0	1
0	1	0
1	0	0
1	1	0

A	B	X
0	0	0
0	1	0
1	0	0
1	1	1

A	B	X
0	0	0
0	1	1
1	0	1
1	1	1

Disponível em: http://www.dpi.inpe.br/~carlos/Academicos/Cursos/ArqComp/aula_5bn1.html. Acesso em: 22 fev. 2019

A) NAND; NOR; AND; OR.

B) NAND; NOR; OR; AND.

C) AND; NOR; NAND; OR.

D) NAND; OR; AND; NOR.

E) NOR; NAND; OR; AND.

Comentários:

As tabelas mais difíceis de verificar são a NAND e a NOR, pois é a negação daquilo que estamos acostumados a montar tabela (OR, AND etc.). Minha sugestão é começar pelas outras tabelas. Por

exemplo, vamos começar pela alternativa A, que diz que a terceira tabela é do AND e a quarta é do OR.

3ª tabela: só tem saída 1 quando A e B são 1. Comportamento do AND.

4ª tabela: basta uma das entradas ser 1 para ter a saída 1. Comportamento do OR.

Olhando as alternativas, já mataríamos a questão aqui! O examinador foi bonzinho! Mas vamos ver as duas primeiras (já sabendo que são NAND e NOR):

1ª tabela: aplico o AND e depois inverto a resposta. Realmente é a tabela NAND!

2ª tabela: aplico o OR e depois inverto a resposta. Realmente é a tabela NOR!

Portanto, a alternativa A está correta e é o gabarito da questão.

Gabarito: Letra A

5. (UFMA/UFMA - 2019) Considerando a expressão booleana S = A.(B + C) + D e a sequência da tabela da verdade a seguir, assinale a alternativa que apresente os resultados de S, na respectiva ordem das possiblidades de A, B, C e D apresentada nesta tabela.

A	B	C	D
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	0
1	0	0	1
1	0	1	0
1	0	1	1
1	1	0	0
1	1	0	1
1	1	1	0
1	1	1	1

A) 0, 1, 1, 0, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0

B) 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1

C) 0, 1, 0, 1, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1

D) 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 1, 1, 1, 1, 1

E) 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 0, 1, 0, 1

Comentários:

Vamos aplicar A . (B + C) + D para as primeiras linhas da tabela e, de acordo com os resultados, podemos matar a questão! Lembre-se que o AND tem prioridade sobre o OR (assim como acontece com a multiplicação sobre a adição). E os parênteses devem ser resolvidos primeiro.

1ª linha (0, 0, 0, 0): 0 . (0 + 0) + 0 = 0 . 0 + 0 = 0 + 0 = 0

2ª linha (0, 0, 0, 1): 0 . (0 + 0) + 1 = 0 . 0 + 1 = 0 + 1 = 1

3ª linha (0, 0, 1, 0): 0 . (0 + 1) + 0 = 0 . 1 + 0 = 0 + 0 = 0

Aqui já podemos ver que só pode ser a alternativa B ou C. Mas vamos fazer mais umas linhas...

4ª linha (0, 0, 1, 1): 0 . (0 + 1) + 1 = 0 . 1 + 1 = 0 + 1 = 1

5ª linha (0, 1, 0, 0): 0 . (1 + 0) + 0 = 0 . 1 + 0 = 0 + 0 = 0

6ª linha (0, 1, 0, 1): 0 . (1 + 0) + 1 = 0 . 1 + 1 = 0 + 1 = 1

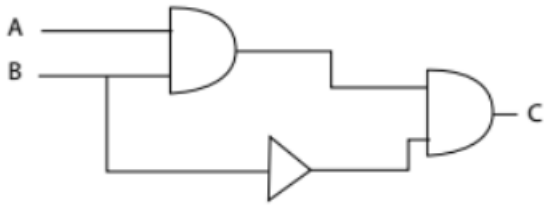
Já está bom...podemos ver que as respostas são iguais até a 8ª linha. Então vamos fazer a 9ª linha para sabermos a resposta!

9ª linha (1, 0, 0, 0): 1 . (0 + 0) + 0 = 1 . 0 + 0 = 0 + 0 = 0

Pronto! A alternativa B está correta e é o gabarito da questão.

Gabarito: Letra B

6. (UFMG/UFMG - 2019) Dado o circuito lógico representado abaixo e as entradas A e B, defina a expressão representada pela saída C:



A) A ^ B ^ B

B) A + B + B

C) A + B + ¬B

D) A ^ B ^ ¬B

Comentários:

1°) Porta AND, tendo como entrada A, B. Temos a saída A ^ B.

2°) Porta NOT (mesmo sem a "bolinha", o triângulo "puro" também é a representação para NOT). Temos a saída ¬B.

3°) Porta AND das saídas de 1° e 2°. Temos (A ^ B) ^ ¬B. Como é tudo AND, podemos tirar os parênteses, ficando assim: A ^ B ^ ¬B.

Portanto, a alternativa D está correta e é o gabarito da questão.

Gabarito: Letra D

7. (UFMG/UFMG - 2019) Nos circuitos lógicos abaixo, considere que A tem valor Falso e B tem valor Verdadeiro.

Qual o valor booleano final de cada circuito?

A) 1. Falso; 2. Verdadeiro.

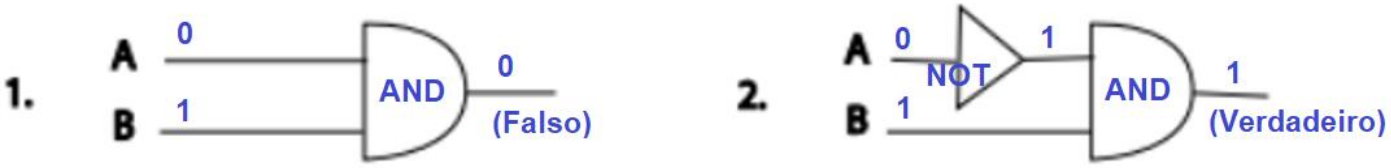
B) 1. Falso; 2. Falso.

C) 1. Verdadeiro; 2. Falso.

D) 1. Verdadeiro; 2. Verdadeiro.

Comentários:

Melhor "riscar" na figura:



Portanto, a alternativa A está correta e é o gabarito da questão.

Gabarito: Letra A

Lista de Questões - Álgebra Booleana e Circuitos Lógicos - Multibancas

1. (ESAF/ATRFB - 2006) Analise as seguintes operações relacionadas à Aritmética Computacional, considerando que os valores utilizados estão na representação hexadecimal.

   I. ((2222 AND AAAA) XOR FFFF) = DDDD.

   II. ((2222 OR BBBB) XOR FFFF) = DDDD.

   III. ((2222 NOT CCCC) XOR FFFF) = 3333.

   IV. ((2222 XOR DDDD) XOR FFFF) = 3333.

   Indique a opção que contenha todas as afirmações verdadeiras.

A) I e II

B) II e III

C) III e IV

D) II e IV

E) I e III

2. (NC-UFPR/COPEL - 2017) Considere as seguintes portas lógicas: 1. Porta NAND. 2. Porta NOT. 3. Porta XOR. 4. Porta NOR. 5. Porta OR. 6. Porta AND. Com relação às simbologias adotadas para portas lógicas em eletrônica digital, numere os parênteses relacionando as figuras com as respectivas portas.

( )  ( )  ( )  ( )  ( )  ( ) 

Assinale a alternativa que apresenta a numeração correta dos parênteses, da esquerda para a direita

A) 6 – 4 – 2 – 5 – 3 – 1.

B) 6 – 2 – 5 – 3 – 4 – 1.

C) 3 – 1 – 4 – 6 – 5 – 2.

D) 1 – 4 – 2 – 5 – 3 – 6.

E) 1 – 2 – 4 – 3 – 5 – 6.

3. (FGV/AL-RO - 2018) A figura a seguir apresenta a tabela verdade de duas portas lógicas.

PORTA LÓGICA 1		
A	B	S
0	0	0
0	1	1
1	0	1
1	1	1

PORTA LÓGICA 2		
A	B	S
0	0	0
0	1	1
1	0	1
1	1	0

As portas lógicas 1 e 2 são, respectivamente,

A) OR e XOR.

B) OR e NAND.

C) AND e NOR.

D) AND e XOR.

E) NOR e OR.

4. (UFGD/UFGD - 2019) Portas ou circuitos lógicos são dispositivos que operam um ou mais sinais lógicos de entrada para produzir uma e somente uma saída, dependente da função implementada no circuito. São geralmente usadas em circuitos eletrônicos, por causa das situações que os sinais deste tipo de circuito podem apresentar: presença de sinal, ou "1"; e ausência de sinal, ou "0".

Disponível em: https://pt.wikipedia.org/wiki/Porta_l%C3%B3gica. Acesso em: 20 fev. 2019.

Para a tabela verdade apresentada a seguir, é correto afirmar que, em ordem, as portas lógicas são:

A	B	X
0	0	1
0	1	1
1	0	1
1	1	0

A	B	X
0	0	1
0	1	0
1	0	0
1	1	0

A	B	X
0	0	0
0	1	0
1	0	0
1	1	1

A	B	X
0	0	0
0	1	1
1	0	1
1	1	1

Disponível em: http://www.dpi.inpe.br/~carlos/Academicos/Cursos/ArqComp/aula_5bn1.html. Acesso em: 22 fev. 2019

A) NAND; NOR; AND; OR.

B) NAND; NOR; OR; AND.

C) AND; NOR; NAND; OR.

D) NAND; OR; AND; NOR.

E) NOR; NAND; OR; AND.

5. (UFMA/UFMA - 2019) Considerando a expressão booleana S = A.(B + C) + D e a sequência da tabela da verdade a seguir, assinale a alternativa que apresente os resultados de S, na respectiva ordem das possiblidades de A, B, C e D apresentada nesta tabela.

A	B	C	D
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	0
1	0	0	1
1	0	1	0
1	0	1	1
1	1	0	0
1	1	0	1
1	1	1	0
1	1	1	1

A) 0, 1, 1, 0, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0

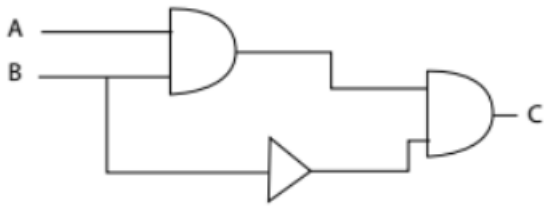
B) 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1

C) 0, 1, 0, 1, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1

D) 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 1, 1, 1, 1, 1

E) 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 0, 1, 0, 1

6. (UFMG/UFMG - 2019) Dado o circuito lógico representado abaixo e as entradas A e B, defina a expressão representada pela saída C:



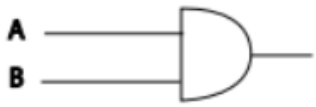
A) A ^ B ^ B

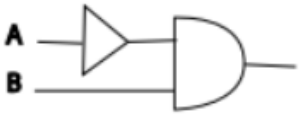
B) A + B + B

C) A + B + ¬B

D) A ^ B ^ ¬B

7. (UFMG/UFMG - 2019) Nos circuitos lógicos abaixo, considere que A tem valor Falso e B tem valor Verdadeiro.

1. 

2. 

Qual o valor booleano final de cada circuito?

A) 1. Falso; 2. Verdadeiro.

B) 1. Falso; 2. Falso.

C) 1. Verdadeiro; 2. Falso.

D) 1. Verdadeiro; 2. Verdadeiro.

GABARITO

GABARITO

1- E
2- A
3- A
4- A
5- B
6- D
7- A

ESSA LEI TODO MUNDO CONHECE:
PIRATARIA É CRIME.

Mas é sempre bom revisar o porquê e como você pode ser prejudicado com essa prática.

1. 

Professor investe seu tempo para elaborar os cursos e o site os coloca à venda.

2. 

Pirata divulga ilicitamente (grupos de rateio), utilizando-se do anonimato, nomes falsos ou laranjas (geralmente o pirata se anuncia como formador de "grupos solidários" de rateio que não visam lucro).

3. 

Pirata cria alunos fake praticando falsidade ideológica, comprando cursos do site em nome de pessoas aleatórias (usando nome, CPF, endereço e telefone de terceiros sem autorização).

4. 

Pirata compra, muitas vezes, clonando cartões de crédito (por vezes o sistema anti-fraude não consegue identificar o golpe a tempo).

5. 

Pirata fere os Termos de Uso, adultera as aulas e retira a identificação dos arquivos PDF (justamente porque a atividade é ilegal e ele não quer que seus fakes sejam identificados).

6. 

Pirata revende as aulas protegidas por direitos autorais, praticando concorrência desleal e em flagrante desrespeito à Lei de Direitos Autorais (Lei 9.610/98).

7. 

Concurseiro(a) desinformado participa de rateio, achando que nada disso está acontecendo e esperando se tornar servidor público para exigir o cumprimento das leis.

8. 

O professor que elaborou o curso não ganha nada, o site não recebe nada, e a pessoa que praticou todos os ilícitos anteriores (pirata) fica com o lucro.

Deixando de lado esse mar de sujeira, aproveitamos para agradecer a todos que adquirem os cursos honestamente e permitem que o site continue existindo.