

Circuitos Digitais - ELE300 - Turma 001

Página Inicial

Avisos

Cronograma

Atividades

Fóruns

Collaborate

Calendário Lives

Notas

Menu das Semanas

Semana 1

Semana 2

Semana 3

Semana 4

Semana 5

Semana 6

Semana 7

Semana 8

Orientações para realização da prova

Orientações para realização do exame

Documentos e informações gerais

Gabaritos

Referências da disciplina

Facilitadores da disciplina

Repositório de REA's

Revisar envio do teste: Semana 4 - Atividade Avaliativa

Usuário

LIZIS BIANCA DA SILVA SANTOS

Curso

Circuitos Digitais - ELE300 - Turma 001

Teste

Semana 4 - Atividade Avaliativa

Iniciado

10/05/24 14:18

Enviado

10/05/24 14:35

Data de vencimento

10/05/24 23:59

Status

Completada

Resultado da tentativa

10 em 10 pontos

Tempo decorrido

16 minutos

Instruções

Olá, estudante!

1. Para responder a esta atividade, selecione a(s) alternativa(s) que você considerar correta(s);

2. Após selecionar a resposta correta em todas as questões, vá até o fim da página e pressione "Enviar teste".

3. A cada tentativa, as perguntas e alternativas são embaralhadas

Pronto! Sua atividade já está registrada no AVA.

Resultados exibidos

Todas as respostas, Respostas enviadas, Respostas corretas, Comentários, Perguntas respondidas incorretamente

Pergunta 1

1,43 em 1,43 pontos

Na prática, o sistema conhecido como "complemento de 2", para fins de representação de números com sinal, mostra-se de suma importância para a indústria da computação em geral. Afinal, essa modalidade possibilita que subtrações possam ser efetuadas, mais precisamente, a partir de meras adições. Isso é valioso em função de determinada circunstância.

Assinale a alternativa que corresponde à descrição correta da circunstância em questão.

Resposta Selecionada:

e. Uso de um mesmo circuito tanto na adição quanto na subtração.

Respostas:

a. Realização da prova dos nove (ou "noves fora") para corrigir cálculos.

b. Sinalização de números de magnitude acima da capacidade da CPU (Central de Processamento Unitário).

c. Blindagem do circuito contra interferência externa dos periféricos.

d. Apuração mais precisa de contas que envolvem valores fracionários.

e. Uso de um mesmo circuito tanto na adição quanto na subtração.

Comentário da resposta:

JUSTIFICATIVA

O expediente permitido pelo sistema de "complemento de 2" acaba se mostrando indispensável para a viabilização de vários sistemas computacionais, porque o fato de se poder usar um mesmo circuito seja para adição, seja para subtração, resulta, efetivamente, em poupar *hardware*, tanto em termos de produção quanto de consumo. Por sua vez, as alternativas "realização da prova dos nove (ou "noves fora") para corrigir cálculos", "apuração mais precisa de contas que envolvam valores fracionários", "sinalização de números de magnitude acima da capacidade da CPU (Central de Processamento Unitário)" e "blindagem do circuito contra interferência externa dos periféricos" levam à formulação de conceitos absolutamente fantasiosos e sem sentido, alheios ao objeto da questão de unificação de circuito para somar e para subtrair, razão pela qual são incorretas e devem ser descartadas.

Pergunta 2

1,43 em 1,43 pontos

Na soma dos números de 8 bits: **10110101+11010010** , quais sinais, carry ou overflow, são gerados?

Resposta Selecionada:

b. Apenas o sinal de carry.

Respostas:

a. Nenhum dos dois sinais.

b. Apenas o sinal de carry.

c. Apenas o sinal de overflow.

d. Não é possível responder apenas com esta informação.

e. Ambos os sinais de carry e overflow.

Comentário da resposta:

JUSTIFICATIVA

A soma destacada resulta em **10110101+11010010=110000111** . Observamos então a presença do carry, pois a resposta final tem 9 bits. Retirando esse bit, observamos que **10000111** representa um número negativo, assim como os dois números de entrada. Desta forma, não houve a presença do sinal de overflow.

Pergunta 3

1,44 em 1,44 pontos

Quando se considera a necessidade de trabalhar com somador e subtrator em linguagem VHDL, é preciso admitir que utilizar constantes não é uma tarefa tão trivial. Afinal, constantes precisam necessariamente ser incluídas em um *package* (ou seja, em um pacote).

Avalie as afirmativas a seguir e a relação proposta entre elas.

I. *Packages* contêm definições de componentes e outras informações que precisam ser mantidas ocultas de todas as demais entidades do arquivo de projeto,

PORQUE

II. no interior do pacote, a palavra-chave CONSTANT é seguida de seu nome simbólico, seu tipo e valor a lhe atribuir, mediante o uso do operador "[]".

Avaliando-se as afirmativas, conclui-se que:

Resposta Selecionada:

a. as duas afirmativas são falsas.

Respostas:

a. as duas afirmativas são falsas.

b. a primeira afirmativa é falsa, e a segunda é verdadeira.

c. as duas afirmativas são verdadeiras, e a segunda não justifica a primeira.

d. as duas afirmativas são verdadeiras, e a segunda justifica a primeira.

e. a primeira afirmativa é verdadeira, e a segunda é falsa.

Comentário da resposta:

JUSTIFICATIVA

A afirmativa I é falsa, porque, pelo contrário, os *packages* carregam definições de componentes e outras informações que precisam estar disponíveis (e não mantidas ocultas) junto a todas as demais entidades do arquivo de projeto.

A afirmativa II é falsa, uma vez que não é o operador "[]", mas sim o operador "[:=" que vincula a palavra-chave CONSTANT ao seu nome simbólico, ao seu tipo e ao valor que lhe deve ser atribuído.

Pergunta 4

1,44 em 1,44 pontos

Um sinal de 4 *bits* chamado s pode ser expresso em linguagem VHDL como "SIGNAL s :BIT_VECTOR (3 DOWNT0 0)". Nesse caso, cada um dos *bits* desses tipos de dados costuma ser designado por um número de elemento. No exemplo aqui aludido, de um vetor de *bits* nomeado s, esses *bits* poderiam muito bem ser rotulados como s3, s2, s1 e s0 — além de poderem ser arranjados em instâncias denominadas "conjuntos".

Avalie as afirmativas a seguir e a relação proposta entre elas.

I. A referência aos três *bits* menos significativos de s como "conjunto" é realizada em VHDL pela expressão s(3 DOWNT0 1),

PORQUE

II. dois conjuntos serão combináveis em uma expressão lógica apenas se possuírem diferentes números de *bits*.

Avaliando-se as afirmativas, conclui-se que:

Resposta Selecionada:

e. as duas afirmativas são falsas.

Respostas:

a. as duas afirmativas são verdadeiras, e a segunda não justifica a primeira.

b. as duas afirmativas são verdadeiras, e a segunda justifica a primeira.

c. a primeira afirmativa é falsa, e a segunda é verdadeira.

d. a primeira afirmativa é verdadeira, e a segunda é falsa.

e. as duas afirmativas são falsas.

Comentário da resposta:

JUSTIFICATIVA

A afirmativa I é falsa, porque a expressão s(3 DOWNT0 1) em VHDL resulta na referência aos três bits mais significativos (e não menos significativos) de s como conjunto.

A afirmativa II é falsa, porque a condição para que dois conjuntos sejam combináveis em uma expressão lógica é a garantia de que ambos tenham estritamente o mesmo tamanho, no sentido de comungarem do mesmo número de *bits* — portanto, não é possível fazê-lo com conjuntos com diferentes números de *bits*.

Pergunta 5

1,42 em 1,42 pontos

Os intervalos de valores de um número de 8 bits representando números inteiros sem sinal (NSS), números em complemento de um (C1) e número em complemento de dois (C2) são, respectivamente:

Resposta Selecionada:

b. NSS: de 0 a 255; C1: de -127 a 127; C2: de -128 a 127

Respostas:

a. NSS: de 0 a 255; C1: de -127 a 127; C2: de -127 a 128

b. NSS: de 0 a 255; C1: de -127 a 127; C2: de -128 a 127

c. NSS: de 0 a 255; C1: de 0 a 255; C2: de -128 a 127

d. NSS: de 0 a 7; C1: de -4 a 3; C2: de -3 a 4

e. NSS: de 0 a 512; C1: de -255 a 255; C2: de -256 a 255

Comentário da resposta:

JUSTIFICATIVA

Um número sem sinal de n bits tem valores de 0 a 2^n-1 , ou seja, no caso de n=8, isso resulta em números de 0 a 255. Um número em complemento de um tem valores no intervalo de $-(2^{n-1}-1)$ a $(2^{n-1}-1)$, ou seja, de -127 a 127 no presente caso. Finalmente, em complemento de dois, os números têm valores no intervalo -2^{n-1} a $(2^{n-1}-1)$, ou seja, -128 a 127 para números de 8 bits.

Pergunta 6

1,42 em 1,42 pontos

O procedimento mais direto que pode ser adotado para fins de representação de um número com sinal é empreender a transformação do MSB (*Most Significant Bit*, ou, em português, Bit Mais Significante) no *bit* de sinal (convencionando-se trabalhar com "0" para positivo e "1" para negativo), mantendo assim os demais *bits* sem qualquer tipo de alteração. Por isso, a forma de sinal-magnitude de 8 *bits* acaba por representar determinada faixa de valores numéricos em base decimal.

Assinale a alternativa que corresponde à descrição correta da faixa de valores em questão.

Resposta Selecionada:

b. -127 a +127.

Respostas:

a. -128 a +128.

b. -127 a +127.

c. -132 a 0.

d. 0 a 132.

e. -256 a +256.

Comentário da resposta:

JUSTIFICATIVA

Com 8 *bits* de forma de sinal-magnitude, compreende-se, consequentemente, a faixa que vai de -127 a +127 — afinal, é a conversão direta da sequência binária 11111111 a 01111111, com o MSB de cada um desses extremos indicando o sinal (no caso, negativo e positivo, respectivamente), e os demais *bits*, o valor numérico efetivo. Por sua vez, as alternativas "-132 a 0", "0 a 132", "-256 a +256 " e "-128 a +128" levam a sequências numéricas matematicamente impossíveis de serem obtidas a partir de uma forma de sinal-magnitude de 8 *bits*, razão pela qual são incorretas e devem ser descartadas.

Pergunta 7

1,42 em 1,42 pontos

Uma vez que comparadores são circuitos que visam equiparar a magnitude de dois números binários, estabelecendo, assim, as relações que se impõem entre eles, a igualdade de 2 *bits* pode ser muito facilmente testada por determinada função — já para mais de 2 *bits*, emprega-se ainda uma porta do tipo AND no arranjo necessário de portas.

Assinale a alternativa que corresponde à descrição correta da função em questão.

Resposta Selecionada:

e. XNOR.

Respostas:

a. NOR.

b. XOR.

c. NAND.

d. XAND.

e. XNOR.

Comentário da resposta:

JUSTIFICATIVA

De forma muito elemental, a função XNOR consegue testar a igualdade de 2 *bits*, uma vez que resulta em saída verdadeira ("1") apenas quando os *bits* de entrada forem, ambos, "0" ou "1". Já para fins de apuração da igualdade de números formado por *n bits*, torna-se necessário dispor de *n* portas XNOR (de 2 *bits* cada), que alimentem uma porta final AND de *n* bits de entrada. Por sua vez, as alternativas "XAND", "NAND", "XOR" e "NOR" levam à formulação de valores completamente divergentes do teste com a função XNOR, razão pela qual são incorretas e devem ser descartadas.

Sexta-feira, 15 de Novembro de 2024 14h46min29s BRT

← OK