

- 1) André Lisboa (1,9 + 3 + 3 + 1,7) = 9,6
  - 1 - 0,5+0,4+0,5+0,5 (Erro bobo na hora de gerar o código objeto)
  - 2 - 3
  - 3 - 3
  - 4 - 1,7 (Faltou citar que AND, ORI e XORI, funcionam, pois usam extensão de 0)
- 2) Bruno Xavier (1,5 + 2,8 + 2,3 + 1,2 ) = 7,8
  - 1 - 0,5+0,4+0,5+0,1 (Errou no código intermediário do J e errou quase tudo no srl)
  - 2 - 2,8 (Erro bobo no cálculo do tempo de execução)
  - 3 - 2,3 (Não respondeu à terceira pergunta da questão)
  - 4 - 1,2
- 3) César Munari (1,5 + 1,1 + 2,1 + 1,7) = 6,4
  - 1 - 0,7+0,6+0,2 (Não processou corretamente o endereço ao gerar o código objeto do J)
  - 2 - 1,1 (O tempo de execução de um programa depende do número de vezes que cada instrução é executada. Ignoraste loops e saltos completamente)
  - 3 - 2,1 (Isto é apenas uma parte da solução. Terias que combinar este produto Booleano em algumas somas Booleanas [portas OU], o que não fizeste)
  - 4 - 1,7 (Errinhos sobre o caminho [uso do PC] e sobre o sinal uins.bw)
- 4) Felipe Lacerda (2 + 2,5 + 3 + 2) = 9,5
  - 1 - 2
  - 2 - 2,5 (Errou na contagem de ciclos pois lb executa em 5 ciclos de relógio, mas usou 4)
  - 3 - 3
  - 4 - 2
- 5) Guilherme Silva (1,1 + 1,7 + 2,2 + 1,2) = 6,2
  - 1 - 0,2+0,3+0,5+0,1 (Entendeu errado a geração do código objeto do J. Inverteu a posição dos registradores no sub. Dois erros ao converter o beq: o salto é para trás [negativo] e o operando de 16 bits no código intermediário não é um endereço, mas um deslocamento.)
  - 2 - 1,7 (Errou a contagem de ciclos por 81 ciclos. Errou o que o programa faz.)
  - 3 - 2,2 (Diz que não se trata de erro, mas é um erro, sem efeito contudo.)
  - 4 - 1,2 (Vários erros no desenho e nas respostas)
- 6) Guilherme Antonini (1,8 + 2,7 + 2,4 + 2) = 8,9
  - 1 - 0,7+0,6+0,7-0,2 (disse que bgt ocupa 2 posições de memória, mas ele ocupa 12 posições)
  - 2 - 2,7 (O exemplo 'a' <= L <= 'j' é errado e sem sentido, l (minúsculo) vem depois do j e L é < a em ASCII!!)
  - 3 - 2,4 (Sem dizer o que são as entradas das portas AND, a questão está terrivelmente incompleta)
  - 4 - 2
- 7) Henrique Paim (1,2 + 1,6 + 3 + 0,7) = 6,5
  - 1 - 0,4+0,4+0,4 (A bgt é uma pseudo traduzida para 3 instruções, você só gerou código aqui para a primeira delas)
  - 2 - 1,6 (Os cálculos de ciclos não fazem nenhum sentido. Errou o valor do tempo em uma ordem de grandeza)
  - 3 - 3
  - 4 - 0,7 (Alguns erros de rotas no MOVZ. Operação da ULA incorreta. Operação na ULA é "Passa opl para a saída")
- 8) Henrique Mrás (2 + 1,6 + 0 + 1,5) = 5,1
  - 1 - 0,7+0,6+0,7
  - 2 - 1,6 (Não calculou tamanho do programa. Há vários erros no cálculo do número de ciclos, valor certo é 1263. \$t3 não é lugar para salvar valores, Posição de memória No é) (4 certas e 6 erradas.)
  - 3 - 0
  - 4 - 1,5 (Acertou as 22 que funcionam, mas acrescentou mais 6 que não funcionam como se funcionassem)

- 9) Jessé Rodrigues (2 + 2,2 + 3 + ) = 7,2  
 1 - 0,5+0,5+0,5+0,5  
 2 - 2,2 (Errou ao calcular o número de ciclos gasto para executar o programa. lb gasta 5 ciclos para executar, consulte a documentação!! Errou o tamanho da área de dados)  
 3 - 3  
 4 - 0 (Questão não respondida.)
- 10) Luiz Buzzacaro (1,3 + 1,6 + 3 + 2) = 7,9  
 1 - 0,5+0,5+0,2+0,1 (Código correto na linha 13 seria 0x0810000A. Rótulo correto na linha 16 seria t\_row)  
 2 - 1,6 (A área de dados cresce a medida que o programa executa. Ao final, em S há 10 elementos da Série de Fibonacci [mais 40 bytes]. As linhas 3-15 executam em 15 ciclos. O laço é repetido 8 vezes e não 7. Resposta ao item b é sem sentido: o que o programa faz é gerar os N primeiros elementos da Série de Fibonacci e escrevê-los a partir da posição de memória S)  
 3 - 3  
 4 - 2
- 11) Mateus da Rosa (1,4 + 2,8 + 1,8 + 1,4) = 7,4  
 1 - 0,5+0,3+0,5+0,1 (Não faz sentido ignorar a instrução já traduzida para código objeto na linha [14] ao traduzir a linhas 12, deveria ter contado, é instrução como as demais. Explicação sobre a linha 14 está incorreta, teria de ter gerado o endereço de salto que é 0x00400078, o endereço onde está o rótulo f)  
 2 - 2,8 (Tamanho da área de dados errado, aceita-se 21 ou 19 bytes e não 36 como dito)  
 3 - 1,8 (Colocou 6 retângulos com texto dentro [e.g. uins.i=ADDU], mas não disse o que são os retângulos. O certo seria que fossem portas lógicas AND)  
 4 - 1,4 (Faltaram alguns caminhos e um está errado, o "11111" do mux M4. Discutiu apenas 5 dos 11 sinais de controle e errou o ciclo de ativação de wreg e CY2)
- 12) Mateus Oliveira (2 + 3 + 3 + 1,4) = 9,4  
 1 - 0,7+0,6+0,7  
 2 - 3  
 3 - 3  
 4 - 1,4 (Disse que nas instruções de deslocamento por número constante de bits o registrador contém o dado imediato, o que é incorreto, pois o registrador RS contém o valor do registrador a deslocar, não o dado imediato. Mesmo erro para as instruções tipo I cujo imediato vem pelo RIM e não pelo RS)
- 13) Miguel Almeida (2 + 2,7 + 2,8 + 1,9) = 9,4  
 1 - 0,6+0,7+0,7  
 2 - 2,7 (Afirma que o laço do programa é executado 40 vezes, mas ele só executa 38 vezes)  
 3 - 2,8 (Ao representar vetores de bits de tamanho fixo é importante sempre mostrar os 0s à esquerda também)  
 4 - 1,9 (Dizer "leva opl a ALU" não faz sentido. Opl é a entrada superior da ALU, sempre)
- 14) Nathan Espíndola (2 + 1,3 + 2,5 + 1,8) = 7,6  
 1 - 0,7+0,6+0,7  
 2 - 1,3 (Erro: cada caracter ASCII ocupa 1 byte, não 4 bytes. Erro: Número de ciclos citado é mais do que o dobro do valor correto [461], sem desenvolvimentos. item d) não realizado)  
 3 - 2,5 (Colocou 4 retângulos com texto dentro [e.g. & uins.i=ADDIU], mas não disse o que são os retângulos. O certo seria que fossem portas lógicas AND. Porque não usou o símbolo certo? Ou não explicou no texto?)  
 4 - 1,8 (Alguns sinais nã foram marcados como atuando)
- 15) Pedro Filippi (2 + 2,2 + 3 + 2) = 9,2  
 1 - 0,5+0,5+0,5+0,5  
 2 - 2,2 (Não expressou o tempo de execução em ns como solicitado. Disse que cada elemento da lista encadeada ocupa 4 bytes, o que está errado, pois para cada campo de valor de 4 bytes é necessário um campo de endereço do próximo elemento)

da lista; logo cada elemento da lista ocupa 8 bytes)

3 - 3

4 - 2

16) Rafael da Rosa (1,3 + 1 + 2 + 1,1) = 5,4

1 - 0,5+0,2+0,3+0,3 (Na linha 12, demonstrou não saber como se calcula o offset. Na linha 11, a decodificação para obter a SLTU está incompleta. Na linha 14 a resposta não contém o código fonte, apenas o intermediário)

2 - 1 (Contou li como 8 bytes, mas li ocupa apenas 4 bytes e ainda somou errado. Mesmo aumentando o número de ciclos para algumas linhas, chegou a valor bem abaixo do correto (837 ciclos), o que indica contas bem incorretas. Ao dizer o que o programa faz usou apenas comentários sintáticos, sem semântica)

3 - 2

4 - 1,1 (Justificativas fracas ou inexistentes para que instruções ainda funcionam)

17) Raffael Daltoé (1,7 + 2,7 + 1,7 + 1,3) = 7,4

1 - 0,5+0,5+0,2+0,5 (Gerou código incorreto para o LB)

2 - 2,7 (Não respondeu que valores são escritos na memória)

3 - 1,7 (Acertou 8, errou 2)

4 - 1,3 (Justificativas incorretas para instruções que funcionam e texto sobre instruções que param de funcionar, o que não foi pedido)

18) Thiago Grehs (2 + 3 + 1 + 1) = 7,0

1 - 0,5+0,5+0,5+0,5

2 - 3

3 - 1 (item b não respondido)

4 - 1 (Resposta à questão é cópia descarada de gabarito antigo, a questão é o quanto do que copiou entendeu...)