



AOC1 SLE3 - Segunda Avaliação

Esta avaliação explora os conteúdos dos capítulos 6, 7 e 8 do programa da disciplina Arquitetura e Organização de Computadores 1. Estes capítulos abordam os conceitos relacionados com os seguintes conteúdos: *Sistema de Interconexão de Computadores*, *Mecanismos de Entrada/Saída*, *Memória Externa* e *Programação MIPS Assembly*.

A avaliação terá duração de 2(duas) horas/aula. Os enunciados das questões estarão disponíveis nesse formulário. Ao encerrar a avaliação, o discente deverá inserir respostas formatadas na seção de formulário correspondente. Quando tiver concluído, envie o formulário com as respostas.

1. Caro(a) aluno(a), gentileza escrever o seu nome completo. *

Davi de Pontes Pasquini

2. Em relação aos módulos de Sistema de Interconexão, Entrada/Saída e seus mecanismos, analise os itens a seguir e marque se a assertiva é verdadeira ou falsa.

As funções mais importantes de um módulo de E/S podem ser divididas nas seguintes categorias: controle e temporização, comunicação com o processador, comunicação com dispositivos, área de armazenamento temporário de dados e detecção de erros. *

(1 Ponto)

☒ Verdadeiro

☐ Falso

3. Um barramento que utiliza **temporização assíncrona** permite a melhor utilização de dispositivos com diferentes taxas de transferência. *

(1 Ponto)

☒ Verdadeiro

☐ Falso

4. O método **daisy-chaining** é uma forma de atribuir prioridades a dispositivos de E/S. Ele permite que os próprios dispositivos decidam quem pode interromper a CPU, ao invés de delegar a CPU a decisão. *

(1 Ponto)

☒ Verdadeiro

☐ Falso

5. O mecanismo de Entrada/Saída independente (**isolated I/O**) utiliza um conjunto de instruções para acessar a memória e um outro conjunto de instruções para acessar os dispositivos de I/O. *

(1 Ponto)

☒ Verdadeiro

☐ Falso

6. O mecanismo de Entrada/Saída orientado a interrupção (**interrupt driven I/O**) a CPU gasta muito tempo verificando o status da controladora de E/S envolvida na operação, a esse tempo é acrescido uma latência da execução da rotina de tratamento da interrupção (**isr**). *

(1 Ponto)

☐ Verdadeiro

☒ Falso

7. Considere um barramento de E/S (*I/O bus*) operando em uma frequência de **30 MHz**, com capacidade de transferir **4 bytes** de dados em um ciclo de barramento. Suponha que um projetista esteja considerando anexar dois módulos controladores de dispositivo a este barramento, listados a seguir:

- Disco rígido, com uma taxa de transferência de **40 MB/s**.
- Placa de vídeo, com uma taxa de transferência de **128 MB/s**.

O barramento com a especificação apresentada tem condições de suportar esses dispositivos? *

(1 Ponto)

☐ Sim

☒ Não

8. O intervalo de tempo medido entre o recebimento de um sinal de interrupção e a realização do serviço (*interrupt handler*) é chamado de *

(1 Ponto)

☐ Interrupt delay

☒ Interrupt latency

☐ Interrupt cycle time

☐ Interrupt context switch

9. Qual das sentenças abaixo **NÃO** é verdadeira para o protocolo **UART** (*Universal Asynchronous Receiver Transmitter*): *

(1 Ponto)

☐ Requer apenas dois fios para transmissão de dados *full duplex*

☐ Existe um bit de paridade para verificação de erros.

☐ É adequado para comunicação ponto-a-ponto entre dispositivos.

☒ Realiza uma comunicação *half duplex*

10. Considere uma operação de E/S que usa a técnica de **DMA** no **modo roubo de ciclo** (*cycle stealing*), pode-se afirmar que

*

(1 Ponto)

- ☐ O controlador DMA (DMAC) toma posse do barramento (dados, endereço e controle) enquanto transfere um bloco de dados entre a memória e o dispositivo de E/S.
- ☒ A transferência de dados ocorre entre o dispositivo de E/S e a memória durante cada ciclo de clock alternativo.
- ☐ Enquanto o microprocessador está executando um programa, um dispositivo assumirá o controle do barramento (dados, endereço e controle) quando o barramento não estiver em uso pelo microprocessador.
- ☐ O controlador DMA aguarda o microprocessador terminar a execução do programa e então assume o controle do barramento.

11. Analise as alternativas a seguir e identifique a **INCORRETA**. *

(1 Ponto)

- ☐ A CPU usa a técnica de DMA para lidar com as entradas de disco.
- ☒ A CPU sempre usa a técnica de polling para lidar com as entradas do teclado.
- ☐ DMA é impróprio para teclados porque não há necessidade de colocar o código das teclas pressionadas diretamente na RAM.
- ☐ Um teclado pode usar interrupções sem prejudicar significativamente o desempenho da CPU porque as teclas são digitadas de forma relativamente lenta.

12. Considere as seguintes técnicas de transferência de dados:

1. Programmed I/O
2. Interrupt-driven I/O
3. Direct Memory Access (DMA)

Ordene as técnicas de forma **decrecente** segundo o **número de interrupções geradas**. *

(1 Ponto)

- ☐ 1, 2, 3
- ☒ 2, 3, 1

☐ 3, 1, 2

☐ 3, 2, 1

13. Qual das seguintes afirmações a seguir é verdadeira sobre o **controlador DMA**? *

(1 Ponto)

- ☐ DMA é uma abordagem para realizar transferências de dados em massa entre a memória e o dispositivo externo sem a intervenção do processador.
- ☐ O controlador DMA atua como um processador para transferências DMA e supervisiona todo o processo.
- ☐ O controlador DMA possui 3 registradores.
- ☒ Todas as alternativas acima

14. Para coordenar a transmissão de dados entre transmissor e receptor, barramentos assíncronos usam protocolos de **handshaking** (aperto de mão). Qual das opções a seguir está **fora de ordem** em um esquema de transmissão **handshaking assíncrono**?

Dicas: etapas podem estar faltando. *

(1 Ponto)

- ☐ O remetente escreve nas linhas de dados.
- ☐ O remetente define o controle "data ready" para 1.
- ☒ O receptor escreve nas linhas de endereço e de controle.
- ☐ O receptor lê as linhas de dados.
- ☐ O remetente define o sinal de controle "data ready" para 0.

15. Quantos bytes de dados podem ser enviados em **20** segundos por um canal de comunicação serial assíncrona **P82** (paridade par, 8 bits de dados, dois stop bits e um start bit), com taxa de transmissão (*baud rate*) de **115200 bps**? *

(1 Ponto)

☐ 288000

- ☒ 192000
- ☐ 160000
- ☐ 120000

16. Identifique a sentença que **NÃO** esta relacionada com interrupções de hardware. *

(1 Ponto)

- ☒ Interrupção de hardware é síncrona.
- ☐ Interrupções de hardware são causadas por dispositivos externos, especialmente falha em hardware.
- ☐ Interrupções de hardware não incrementam o registrador *Program Counter (PC)*.
- ☐ Interrupção de hardware tem prioridade mais baixa do que interrupções de software.

17. Em relação a programação MIPS assembly, o conceito "**modos de endereçamento do processador**" é usado para ... *

(1 Ponto)

- ☐ Diferenciar entre memória do código e memória de dados.
- ☐ Determinar o número de posições de memória acessíveis.
- ☐ Especificar os tipos de dados que podem ser acessados.
- ☒ Determinar a localização dos operandos da instrução.

18. Qual das opções a seguir traduziria corretamente a pseudo-instrução **move \$t1, \$t0** em uma sequência de instruções nativas do MIPS? *

(1 Ponto)

- ☐ add \$t1, \$t0, \$t0
- ☐ add \$t0, \$t1, \$t1
- ☒ add \$t1, \$t0, \$0
- ☐ addi \$t0, \$t1, 0

19. O fragmento de código abaixo, escrito em MIPS assembly, expressa uma sequência de comandos usados para preservar o conteúdo dos registradores **\$s0** e **\$s1** no topo da pilha:

fun:

addi \$sp, \$sp, -8

sw \$s0, 0(\$sp)

sw \$s1, 4(\$sp)

aqui será inserido uma sequência de instruções

...

Assumindo que nem o ponteiro da pilha nem o conteúdo da pilha foram alterados pelo código do corpo da função, qual das seguintes opções a seguir permitiria recuperar o conteúdo de **\$s0** e **\$s1** e retorna **\$sp** ao valor original (pré-decrementado)? *

(1 Ponto)

- ☐ addi \$sp, \$sp, 8; lw \$s0, 4(\$sp); lw \$s1, 0(\$sp)
- ☐ addi \$sp, \$sp, 8; lw \$s0, 0(\$sp); lw \$s1, 4(\$sp)
- ☐ lw \$s0, 4(\$sp); lw \$s1, 0(\$sp); addi \$sp, \$sp, 8
- ☒ lw \$s0, 0(\$sp); lw \$s1, 4(\$sp); addi \$sp, \$sp, 8

20. Para o problema a seguir considere a declaração de um tipo estruturado *Point* e um fragmento de código escrito em linguagem C:

```
struct Point {  
    char x;  
    char y;  
} p[10];  
...  
int i;  
for(i=0; i<10; i++) {  
    p[i].x = 0;  
    p[i].y = 0;  
}
```

A seguir, é ilustrado um código que converte o fragmento de código C para a linguagem de montagem do MIPS (*assembly*). Preencha os espaços em branco que garanta a boa conversão C para MIPS:

la	\$8, p	# \$8 aponta para p[0]
addi	\$9, \$8, __1__	# \$9 aponta para p[10]
L1:	bge \$8, \$9, L2	# controle da repetição do laço
sw	\$0, 0(\$8)	# p[i].x = 0
sw	\$0, __2__(\$8)	# p[i].y = 0
addi	\$8, \$8, __3__	# i++
j	L1	
L2:		

Marque a alternativa que completa o código corretamente, preenchendo os espaços indicados por **1**, **2** e **3**, nessa ordem.

*

(4 Pontos)

- ☐ 80, 4, 1
- ☐ 80, 4, 8
- ☐ 10, 1, 1
- ☒ 20, 1, 2

21. Faça uma análise do fragmento de código MIPS assembly abaixo. Descreva de maneira objetiva o que o código realiza.

```
lui $t0, 0xCAFE
addi $t1, $zero, 4
xor $t2, $t2, $t2
L1: srl $at, $t0, 28
sll $t2, $t0, 4
or $t0, $t2, $at
addi $t1, $t1, -1
bne $t1, $0, L1 *
```

(4 Pontos)

```
lui $t0, 0xCAFE # coloca 0xCAFE no registrador $t0
addi $t1, $zero, 4 # $t1 = 4
xor $t2, $t2, $t2 # $t2 = 0
L1: srl $at, $t0, 28 # $at = $t0 >> 28 // desloca 28 bits pra direita
sll $t2, $t0, 4 # $t2 = $t0 << 4 // desloca 4 bits pra esquerda
or $t0, $t2, $at # $t0 = 1 se $t2 ou $at ou os dois forem 1
addi $t1, $t1, -1 # $t1 = $t1 - 1
bne $t1, $0, L1 # se $t1 != 0, vá para L1
```

O programa desloca os bits de \$t0 4 (\$t1) vezes, ou seja, joga o CAFE para bits mais

22. Considere a geometria de uma unidade de disco rígido (HD) de **15000 rpm**, com **8** cabeçotes de leitura, **1024** cilindros, **512** setores por trilha e setores de **512** bytes. A unidade HD possui um tempo de busca médio de **2ms** (*seek time*).

- **a)** Qual é a capacidade da unidade de disco rígido?
- **b)** Qual é o tempo de acesso médio de um setor da unidade de disco?
- **c)** Calcule o tempo estimado para transferir um arquivo de **6 MB**. Considere que o arquivo foi armazenado na unidade de disco em setores, cilindros e trilhas contíguas.

*

(4 Pontos)

a) cabeçotes * bytes * cilindros * setores por trilha
 $8 * 512 * 1024 * 512 = 2147483648$ Bytes ou aprox 2 Gigabytes

b) $Tam = \text{tempo de busca} + (1 / 2 * (\text{velocidade de rotação})) + \text{capacidade do setor} / (\text{capacidade da trilha} * \text{velocidade de rotação})$
 $Tam = 0.002 + (1 / 2 * 250) + 512 / (262144 * 250)$
 $Tam = 0.0040078125$ seg ou 4.0078 ms

c) Tamanho de cada trilha = 0,15 MB (262144 bytes)
6 MB ocupam 40 trilhas, logo:

Primeira trilha:

$TamC = \text{tempo de busca} + (1 / 2 * (\text{velocidade de rotação})) + \text{capacidade do setor} / (\text{capacidade da trilha} * \text{velocidade de rotação})$
 $TamC = 0,002 + (1 / 2 * 250) + 512 / (262144 * 250)$
 $TamC = 0.0040078125$ seg ou 4.0078 ms

Para as outras 39 trilhas tempo de busca = null, portanto:

$TamCC = (1 / 2 * (\text{velocidade de rotação})) + \text{capacidade do setor} / (\text{capacidade da trilha} * \text{velocidade de rotação})$
 $TamCC = (1 / 2 * 250) + 512 / (262144 * 250)$
 $TamCC = 0,0020078125$ seg ou 2,0078 ms

Tempo de transferência dos 6 MB:

$\text{Tempo} = TamC + (39 * Tam)$
 $\text{Tempo} = 4,0078 + (39 * 2,0078)$
 $\text{Tempo} = 4,0078 + 78,3042$

Este conteúdo foi criado pelo proprietário do formulário. Os dados que você enviar serão enviados ao proprietário do formulário. A Microsoft não é responsável pela privacidade ou práticas de segurança de seus clientes, incluindo aqueles do proprietário deste formulário. Nunca forneça sua senha.

Da plataforma Microsoft Forms | [Política de privacidade](#) | [Condições de uso](#)