

Introdução aos Sistemas e Redes de Computadores

Licenciatura em Engenharia Informática

3º Teste – 12/01/2008 – Duração: 1 hora

**Por favor não mexa neste enunciado até o professor dar início ao teste.
Entretanto, leia esta página com muita atenção!**

Este enunciado é composto por:

- Uma folha de rosto (esta que está a ler);
- Uma folha de respostas (no final);
- Duas folhas (com o verso em branco) com o enunciado do teste.

Descrição do teste e preenchimento da folha de respostas:

- O teste contém três (3) perguntas de desenvolvimento.
Deve responder dentro da respectiva caixa na folha de respostas;
- O teste contém nove (9) perguntas de escolha múltipla.
Na zona inferior da folha de respostas está uma tabela com oito colunas, numeradas de 4 a 12. Deve colocar o número que corresponde à sua escolha na caixa central (branca) da coluna com o número da questão. Caso a sua resposta esteja rasurada, deverá repeti-la na caixa (sombreada) imediatamente por baixo.
Atenção: *Se a caixa de baixo (sombreada) estiver preenchida, a caixa central (branca) será ignorada. Muito importante: Respostas erradas descontam 20% da cotação da pergunta! Ou seja, se não responder a uma pergunta, perde 1.5 valores. Se responder errado, perde 1.8 valores.! O risco não compensa! Se não tem a certeza da resposta, a melhor opção é não responder.* A cotação mínima do grupo de perguntas de escolha múltipla é zero (0) valores.
- Como zona de rascunho pode e deve usar o verso desta folha de rosto e das restante folhas de enunciado.
Não use o verso da folha de respostas como rascunho.
- Se a sua resposta exceder o espaço reservado para alguma das questões 1), 2) ou 3), use o verso da folha de respostas, indicando muito claramente na caixa da pergunta “*ver verso*” e no verso identifique claramente a que pergunta a resposta se refere;
- Pode desagrafar as folhas do teste. No final apenas deverá entregar ao docente a folha de respostas.

Outras indicações:

- O teste é sem consulta;
- Não é permitido o uso de equipamento electrónico (telemóveis, calculadoras e afins);
- Todo o material, à excepção de lápis, borracha e caneta, deverá ser colocado junto ao quadro;
- Terá que ficar na sala até ao fim do teste, mesmo que queira desistir, ou já tenha completado o teste;
- Só pode entregar a folha de respostas quando o teste terminar;
- Em caso de desistência deverá indicá-lo no canto superior direito da folha de respostas, assinar, e entregar a folha de respostas ao docente (no final do teste);
- Não serão permitidas quaisquer saídas (por exemplo, WC) durante o teste;
- Não haverá nenhum esclarecimento de dúvidas. Os alunos deverão apenas avisar o docente em caso de suspeita de erro ou gralha no enunciado;
- O teste tem a duração de uma hora, sem tolerância. Quando o tempo terminar, deverá parar imediatamente mas não se pode levantar nem falar. O docente irá recolher as folhas de resolução (e apenas as folhas de resolução). Depois de recolhidas todas as folhas, o docente dará indicação que se poderão levantar e sair.
- Fraude ou tentativa de fraude, activa ou passiva, implica a reprovação imediata à cadeira!
- A melhor ordem para responder às questões não é, necessariamente, a ordem pela qual estas são apresentadas. Analise bem por onde irá começar a resolver o seu teste e quanto tempo deve dedicar a cada pergunta.

- [3.0 val.] Considerando a programação em *assembly* para o Marie estudada nas aulas, faça um programa que calcule o maior de uma sequência de números positivos introduzidos pelo utilizador. Obs: a sequência de números termina quando o número lido for zero, que não será considerado como fazendo parte da sequência.
- [3.0 val.] Implemente um programa completo em *assembly* para o Marie que:
 - Lê três valores introduzidos pelo utilizador e os guarda nas células de memória com etiquetas N1, N2, e N3;
 - Invoca uma subrotina, por si programada, que calcula $N1+N2-N3$ e devolve o resultado na etiqueta VAL;
 - Apresenta no ecrã o valor calculado na subrotina.
- [2.75 val.] Assuma que o Marie tem uma instrução `LOADI addr` que permite realizar um *load* indirecto. Apresente o RTL (*Register Transfer Language*) do ciclo *fetch-decode-execute* desta instrução.

Opcodes Marie	
Cód.	Instrução
0	JnS
1	Load
2	Store
3	Add
4	Subt
5	Input
6	Output
7	Halt
8	Skipcond
9	Jump
A	Clear
B	AddI
C	JumpI

- [1.25 val.] Considere o seguinte código em *assembly* do Marie:

input	contin, store z	O código apresentado calcula... (1) O resto da divisão inteira de x por y (2) O resto da divisão inteira de y por x (3) A divisão inteira de y por x (4) A divisão inteira de x por y (5) Nenhuma das anteriores
store x	jump ciclo	
input	fim, load z	
store y	output	
store z	halt	
ciclo, subt x	x, dec 0	
skipcond 000	y, dec 0	
jump contin	z, dec 0	
jump fim		

- [1.25 val.] Dado o seguinte estado da memória e dos registos:

Memória (Hex.)	Registos (Hex.)	Qual é o estado dos mesmos registos após uma iteração do ciclo Fetch-Decode-Execute?
100 1103	PC=102	(1) PC=103, AC=0006, MAR=103 (2) PC=102, AC=001A, MAR=103
101 3103	AC=0010	(3) PC=103, AC=0006, MAR=102 (4) PC=103, AC=000A, MAR=103
102 4103	MAR=103	(5) PC=103, AC=001A, MAR=103
103 000A		

- [1.25 val.] Dado o seguinte estado da memória e dos registos:

Memória (Hex.)	Registos (Hex.)	Qual é o estado dos mesmos registos após uma iteração do ciclo Fetch-Decode-Execute?
100 C105	PC=100	(1) PC=101, IR=C105, MAR=100, MBR=0000 (2) PC=102, IR=3103, MAR=102, MBR=0105 (3) PC=101, IR=3103, MAR=104, MBR=0103 (4) PC=102, IR=C105, MAR=105, MBR=0102 (5) Nenhuma das anteriores
101 5000	IR=0000	
102 3103	MAR=000	
103 0103	MBR=000	
104 0104		
105 0102		

- [1.25 val.] Dado o programa em código máquina na coluna da esquerda, indique qual é o programa equivalente no *assembly* do MARIE!

Memória (Hex.)	(1)	(2)	(3)	(4)	(5) Nenhuma das anteriores
5000	input	input	input	input	
8800	skipcond 000	skipcond 800	skipcond 800	skipcond 000	
2211	store 11	store 11	store 211	store 211	
2222	store 22	store 22	store 222	store 222	
7000		halt	halt		

8. [1.25 val.] Dado o programa em código máquina na coluna da esquerda, indique qual é o estado da tabela de símbolos do *assembler* no final do 1º passo do processo de assemblagem.

org 50	(1)	(2)	(3)	(4)	(5) Nenhuma das anteriores																								
load x																													
j, sub y	<table border="1"><tr><td>x</td><td>50</td></tr><tr><td>y</td><td>51</td></tr><tr><td>j</td><td>52</td></tr></table>	x	50	y	51	j	52	<table border="1"><tr><td>x</td><td>55</td></tr><tr><td>y</td><td>56</td></tr><tr><td>j</td><td>4</td></tr></table>	x	55	y	56	j	4	<table border="1"><tr><td>x</td><td>55</td></tr><tr><td>y</td><td>56</td></tr><tr><td>j</td><td>51</td></tr></table>	x	55	y	56	j	51	<table border="1"><tr><td>x</td><td>5</td></tr><tr><td>y</td><td>1</td></tr><tr><td>j</td><td>400</td></tr></table>	x	5	y	1	j	400	
x	50																												
y	51																												
j	52																												
x	55																												
y	56																												
j	4																												
x	55																												
y	56																												
j	51																												
x	5																												
y	1																												
j	400																												
skipcond 400																													
halt																													
jump j																													
x, dec 5																													
y, dec 1																													

9. [1.25 val.] Qual das seguintes afirmações é verdadeira?

- (1) O *data bus* transporta os bits que definem o endereço de memória a aceder
- (2) O *address bus* estabelece a ligação entre o *data bus* e a memória
- (3) O *data bus* estabelece a ligação entre os registos e a ULA
- (4) O *data bus* estabelece a ligação entre o *address bus* e a memória
- (5) Nenhuma das anteriores

10. [1.25 val.] Qual das seguintes afirmações é verdadeira?

- (1) Quando o processador está a tratar uma interrupção A , se chegar uma outra interrupção $B \neq A$, então o processador suspende o tratamento de A e vai tratar B , retomando posteriormente o tratamento de A
- (2) Quando o processador recebe uma interrupção suspende imediatamente o ciclo *fetch-decode-execute (fde)*, vai processar a interrupção, retomando depois o ciclo *fde* no ponto onde estava
- (3) A rotina de tratamento de interrupções do tipo A é invocada automaticamente quando o processador recebe uma interrupção desse tipo
- (4) Uma interrupção é sempre desencadeada por um evento externo ao microprocessador
- (5) Nenhuma das anteriores

11. [1.25 val.] Qual das seguintes afirmações é verdadeira?

- (1) Uma célula de memória do Marie contém 16 bits
- (2) Uma célula de memória do Marie contém 8 bits
- (3) Uma célula de memória do Marie contém duas regiões diferenciadas, uma com 4 e outra com 12 bits
- (4) Uma célula de memória do Marie contém 32 bits
- (5) Nenhuma das anteriores

12. [1.25 val.] Qual das seguintes afirmações é verdadeira?

- (1) As instruções num processador CISC são todas mais complexas que as de um processador RISC
- (2) Um programa para um processador RISC tem, normalmente, mais instruções (é maior) que um programa idêntico para um processador CISC
- (3) Uma das características dos processadores RISC é terem um conjunto de instruções máquina elevado
- (4) O Pentium (de 64 bits) é um processador CISC enquanto o AMD64 (também de 64 bits) é um processador RISC
- (5) Nenhuma das anteriores

Nome: _____ Número: _____

Questão 1:

%

Questão 2:

%

Questão 3:

%

Questão ____: (use como substituição ou continuação)

Versão: acabd

Questão:	4	5	6	7	8	9	10	11	12
Resposta:									