Arquitetura de Computadores
2005/2006 2º Semestre
Repescagem 1º Teste - 10/07/2006

Número: _____ Nome: _____

INSTRUÇÕES:

- A duração da prova é de 2 horas.
- Preencha imediatamente o seu número e nome de forma bem legível; identifique todas as folhas.
- Responda apenas no presente enunciado; não serão aceites folhas adicionais. Seja sucinto.
- Cada grupo deve ficar em folhas separadas. Pode usar o verso das folhas para rascunho.
- A cotação das perguntas está indicada entre parênteses à direita do texto.

I

1. Considere os seguintes valores para os registos do processador P3:

R1	R2	R3	R4	R5	R6	R7	PC	SP	RE
0001h	FFFFh	0370h	41F0h	805Ah	0101h	0000h	01A9h	FADEh	0005h

Para as alíneas a), b) e c), indique qual o novo valor, em hexadecimal, para todos (e apenas) os registos que são alterados pelo ciclo completo de execução de cada instrução. Use ? caso não tenha informação suficiente para determinar o novo valor de um registo.

As alíneas são independentes, isto é, assuma como valores iniciais para cada alínea os indicados na tabela acima.

a) ADDC R1, 0000 (1)

R1	R2	R3	R4	R5	R6	R7	PC	SP	RE

b) MVBH R3, R4 (1)

R1	R2	R3	R4	R5	R6	R7	PC	SP	RE

c) RORC R7, 12 (1)

R1	R2	R3	R4	R5	R6	R7	PC	SP	RE

d) Para o ciclo completo de execução da instrução OR M[SP+2], R2 indique na tabela seguinte a sequência de acessos à memória, especificando o endereço, dados e tipo de acesso (Leitura/Escrita). (2)

Nota 1: A tabela tem 5 posições, utilize apenas as que achar necessárias.

Nota 2: Utilize os valores iniciais dos registos indicados na tabela no cimo desta página.

Nota 3: Use ? para indicar que não tem informação suficiente para determinar um dado valor.

	Endereço	Dados	L / E
1			
2			
3			
4			
5			

II

2.1 Escreva um programa em Assembly do P3 equivalente ao seguinte programa em C. Identifique quais os registros usados e procure otimizar o programa (memória usada e rapidez). (2)

```
#define MAX 256

register short int odd;
register short int even;
register short int index;
short int checksum;
short int DATA[MAX];

odd = 0;
even = 0;
checksum = 0;

index = MAX-1;
do{
    checksum += DATA[index];
    if((DATA[index]%2) == 0) /* % -> resto da divisão inteira */
        even++;
    else
        odd++;
    index--;
} while(index >= 0);
```

2.2 Pretende-se desenvolver a rotina ProcNome em Assembly do P3 que processa uma cadeia de caracteres com o nome de uma pessoa. A rotina deve assegurar que cada nome fica em minúsculas com o primeiro caracter em maiúscula (ver exemplo mais abaixo).

Considere-se que:

1. Cada cadeia de caracteres é terminada pelo caracter '@', representada pelo valor 40h;
2. O caracter referente ao “espaço” é representado pelo valor 20h;
3. A codificação de uma determinada letra minúscula é obtida somando 32 (20h) ao valor da representação da letra maiúscula correspondente.

Exemplos: char(A) = 41h => char(a) = 61h

char(B) = 42h => char(b) = 62h

...

char(Z) = 5Ah => char(z) = 7Ah

4. A cadeia de caracteres só possui letras maiúsculas e minúsculas (NÃO acentuadas), caracteres “espaço” e o caracter terminador (@).

A rotina recebe o endereço da primeira posição da cadeia de caracteres através da pilha e retorna, também através da pilha, o número total de caracteres da cadeia (sem contar o caracter terminador). Exemplo de chamada de ProcNome:

```
PUSH Nome
CALL ProcNome
POP M[NumChars]
```

Exemplo do funcionamento da rotina:

Antes:	M[Nome]: ANTONIO vICEntE foZ@	M[NumChars]: ?
Depois:	M[Nome]: Antonio Vicente Foz@	M[NumChars]: 19

a) Desenhe um fluxograma que realize a função pretendida. (1)

Número: _____ Nome: _____

4/6

b) Escreva a rotina em linguagem Assembly do P3.

(3)

III

3.1 Considere a seguinte rotina de tratamento de uma interrupção que é chamada periodicamente (a variável `var1` foi colocada a 0 no início do programa).

```
TrataInt: BR.NZ L2  
          INC M[var1]  
L2:      RTI
```

a) Indique, justificando, qual a funcionalidade desta rotina e qual o erro de concepção cometido que impede o seu correcto funcionamento. **(1,5)**

b) Re-escreva a rotina `TrataInt` por forma a que ela passe a cumprir correctamente a sua função. **(2)**

3.2 Indique todos os passos de inicialização necessários (escrever código assembly) para que a rotina `TrataInt` seja evocada quando ocorre uma interrupção gerada pelo temporizador do P3. **(1,5)**

IV

4.1 Considere o seguinte troço de código em Assembly do P3:

```
ORIG 10h  
  
PUSH M[R2+80h]  
BR.Z -15
```

Traduza estas instruções Assembly para o código objecto correspondente, indicando o endereço de memória e o respectivo conteúdo em hexadecimal. **(2)**

Nota 1: Tenha em atenção a base de representação das constantes.

Nota 2: Preencher apenas as posições necessárias da tabela seguinte.

Endereço	Conteúdo

4.2 As instruções de deslocamento (*shift*) do P3 têm apenas 1 operando. Indique, justificando, uma vantagem e uma desvantagem caso essas instruções tivessem 2 operandos e dê um exemplo concreto de cada situação. **(2)**