8,17 53

EA 869 – Turma U – 1. Semestre 2008
Prova 4 – 29/05/2008 – Prof. Léo Pini Magalhães
(sem consulta)

Q1. 2,0
Q2. 2,95
Q3. 2,4

Nome: Ricardo Diogo Righetto Número: 064144

Q1 (2,75) Considere um programa com o trecho abaixo. Mostre o estado da pilha, de seu registrador associado SP e dos registradores e endereços indicados, após a execução dos comandos (a) até (i). Em cada linha **mostre com um círculo** os valores alterados pelo comando. Se necessário coloque valores coerentes em PC. As instruções têm comprimento 1.

		move dado1, R1
9009	**(a)**	.....
900A	**(b)**	push R1
900B	**(c)**	call SUBROTINA
	(i)	pop R2
		move R2, dado2

		SUBROTINA
800E	**(d)**	pop R2
	(e)	pop R4
		.....(R2 não é alterado)
		.....(R4 recebe valor 1C)
	(f)	push R4
	(g)	push R2
	(h)	return

a) (2,0) preencha a tabela abaixo – **UTILIZE SEMPRE HEXADECIMAL**

\ após	R1	R2	R4	SP	1508	1509	150A	150B	150C	PC
(a)	**9**	**4**	**3**	**150A**	**5**	**2**	**6**	**B**	**1**	**900A**
(b)	9	4	3	150B	5	2	9	B	1	900B
(c)	9	4	3	150C	5	2	9	900C	1	800E
(d)	9	900C	3	150B	5	2	9	900C	1	800F
(e)	9	900C	9	150A	5	2	9	900C	1	8010
(f)	9	900C	1C	150B	5	2	1C	900C	1	8010+X
(g)	9	900C	1C	150C	5	2	1C	900C	1	8010+X+1
(h)	9	900C	1C	150B	5	2	1C	900C	1	900C
(i)	9	1C	1C	150A	5	2	1C	900C	1	900D

2

b)(0,75) considere dado1 e dado2 como as variáveis do programa principal. Justifique se a passagem de parâmetros é por valor ou endereço.

A passagem de parâmetros é por endereço, pois o retorno é feito para o endereço apontado por dado2.

16 - POS=08 SIZE= 2 CODE= 0000

PLA+POS= 03CE+08=03D6

17 - POS=0A SIZE= 2 CODE=0000

PLA+POS= 03D8

18- POS= 02 SIMB='SEC' ENDR= POS+PLA= 02+03CE=03D0

VALOR= 03CE (GEST)

(ENDR)= (03D0)=((03D0))+ VALOR= 0000+03CE=03CE

19- PLA= PLA+SCOMP=03CE+0C= 03DA

FIM

Q2. (3,75) Você utiliza um processador com uma arquitetura para controle de interrupção que controla 4 linhas de interrupção (L1 a L4) sendo o controle de atendimento por máscara. As linhas L1 a L4 tem a si associados os níveis 1 a 4 respectivamente. Os bits 0, 1, 2 e 3 da PSW armazenam a máscara das linhas L1, L2, L3 e L4 respectivamente. Programe o processador de forma a atuar como a seguir:

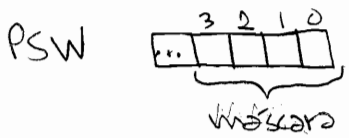
- a rotina de serviço Rot4 não poderá ser interrompida;
- a rotina de serviço Rot3 poderá ser interrompida por qualquer linha;
- a rotina de serviço Rot2 poderá ser interrompida pelas linhas L1 e L2;
- a rotina de serviço Rot1 só poderá ser interrompida por L3;
- O programa principal só poderá ser interrompido por L3.

O programa principal inicia em 1000h, as rotinas em: Rot1 2000h, Rot2 3000h, Rot3 4000h, Rot4 4050h. O esquema de interrupção é vetorizado (endereço do vetor inicia em 5000h) associando duas posições a cada linha de interrupção, a primeira contendo **Nova-PSW** e a outra **End-rotina**.

a: (1,0) Forneça as instruções geradas pela CPU ao aceitar a interrupção inicial. //→ comentário

```
PUSH PC // armazena valor atual de PC na pilha
PUSH PSW // armazena valor atual de PSW na pilha
MOVE END1, PSW // ativa a Nova-PSW adequada, apontada por END1
JUMP @(END1+1) // salta para o endereço da rotina de serviço adequada (End-rotina), armazena em END1+1 no vetor de interrupção
```

b. (2,75) Escreva um código (use ORG, DW, MOVE) para definir todos os elementos acima descritos com valores que determinem o comportamento descrito (vetor de interrupção, Rot1, Rot2, Rot3, Rot4, programa principal).




```
L4→ máscara 0000   //→ comentário
L3→ máscara 1111
L2→ máscara 0011
L1→ máscara 0100

        ORG 5000H
END1:   DW  0100  // Nova-PSW1
        DW ADR 2000H // End-rotina1
END2:   DW  0011  // Nova-PSW2
        DW ADR 3000H // End-rotina2
END3:   DW  1111  // Nova-PSW3
        DW ADR 4000H // End-rotina3
END4:   DW  0000  // Nova-PSW4
        DW ADR 4050H // End-rotina4
        ORG 2000H
Rot1:   ---       // Rotina de serviço de L1
        RTI
```

Continua no verso

Q3. (2,5) Defina uma macro que trate a soma de 1 elemento com até 2 outros elementos e armazene em um quarto elemento.
A chamada da macro tem o seguinte formato: SomaM B1,B2,B3,B4

(a) (1,0) Defina a macro SomaM:

- Considerando que B1 seja o parâmetro mandatório (sempre válido)
- Se não for especificado elemento para armazenar o resultado, este ficará em R1

```
MACRO     SomaM  B1, B2, B3, B4
          MOVE #0, R1
          ADD  B1, R1
          AIF ('B2'='') .SOMA2
          ADD B2, R1
.SOMA2:   AIF ('B3'='') .SOMA1
          ADD B3, R1
.SOMA1:   AIF ('B4'='') .FIM
SOMA11:   MOVE R1, B4
.FIM:     ENDMACRO
```

(b) (1,0) Realize as duas expansões indicadas abaixo:

(b1) SomaM X, ,Y,Z

```
MOVE #0, R1
ADD X, R1
ADD Y, R1
MOVE R1, Z
```

(b2) SomaM X, , ,Z

```
MOVE #0, R1
ADD X, R1
MOVE R1, Z
```

(c) (0,5) Em qual momento se dá o tratamento de macros? Qual a entrada e qual a saída do programa tratador de macros ?

O tratamento de macros se dá em tempo de montagem, ou seja, quando se faz a expansão do texto do código-fonte e este é "convertido" em código de máquina.

E entrada do programa tratador de macros é o texto do código-fonte, e sua saída é a versão expandida do mesmo.

```
        ORG 3000H
Rot2:   ---
        RTI          // Rotina de serviço de L2

        ORG 4000H
Rot3:   ---
        RTI          // Rotina de serviço de L3

        ORG 4050H
Rot4:   ---
        RTI          // Rotina de serviço de L4

        ORG 1000H
MAIN:   MOVE #0100,PSW  // Máscara do programa principal
        ---             // Programa principal
```

OBS: Com o esquema de máscaras definido no código acima, o programa deve funcionar como especificado. Foi considerado que o "nível" de cada linha de interrupção representava a máscara associada à sua rotina de serviço.

ref. não em
mas nível. ok!