

(com consulta a 1 folha-A4 que não pode ser fotocópia – assine a sua folha)

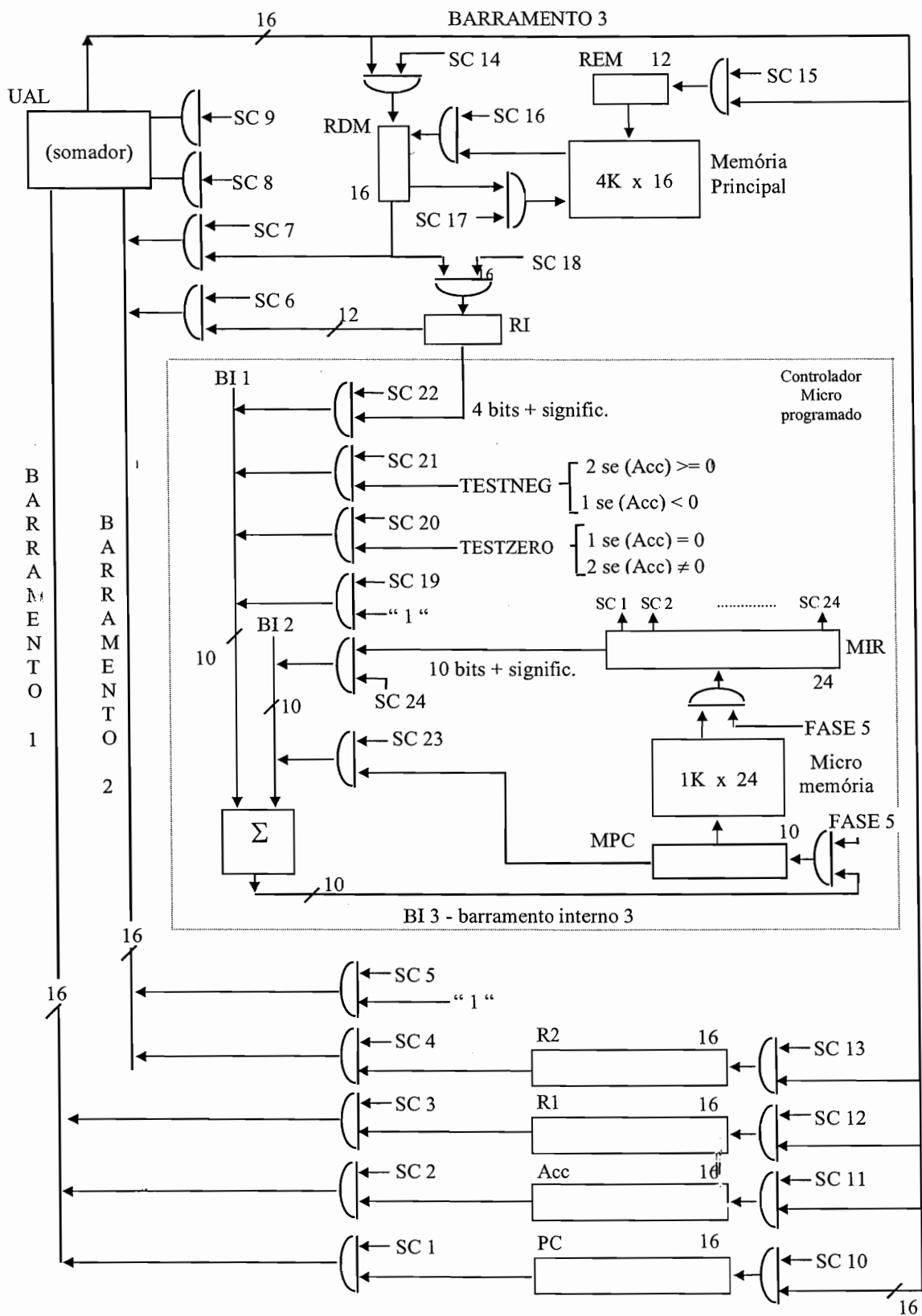
Número:

- O primeiro endereço livre da **micromemória** para uso é:  $A1_{16}$  + seus 2 últimos =  $\phi A$  (hex dígitos do RA (transforme para hexadecimal e opere).
- A área para o programas inicia no endereço  $100_{16}$  da **memória principal** (endereços 00 a FF são para o sistema e a pilha).   
  $60$  (hex)

**PUSH Acc** ; sendo R1 o apontador de pilha e o CO=0101

end. hexadec	microoperações	Sinais de controle	seus comentários
$AL + OA = AB$	1, 15 16 19, 23	$REM \leftarrow (PC) + 60$ $RDM \leftarrow ((REM))$ $MPC \leftarrow (MPC) + 1$	Se (PC) $\rightarrow$ leva p/ o início do programa e busca a instrução
AC	1, 5, 10 18 22, 23	$PC \leftarrow (PC) + 1$ $RI \leftarrow (RDM)$ $MPC \leftarrow (MPC) + RI_{16}$	Incrementa PC e coloca a instrução no Registrador de Instruções
$AC + CO = BD$	<del>7D</del> BE ou! <u>2, 3, 4, 5, 6, 8, 24</u>	$MPC \leftarrow (hexa) BD + 1$	Desvia para a instrução
?	:	:	
BE	3, 15 19, 23	$REM \leftarrow (R_1)$ $MPC \leftarrow (MPC)$	Coloca SP no Registrador de Endereços
BF	2, 14 13, 23	$RDM \leftarrow (Acc)$ $MPC \leftarrow (MPC) + 1$	transfere Acc para o registrador de dados
CO	5, 3, 12 19, 23	$RL \leftarrow (RL) + L$ $MPC \leftarrow (MPC) + 1$	Soma o SP, no caso $SP = 70 + L$
CL	<u>17</u> ou! 19, 23	$m \leftarrow (RDM)$ $MPC \leftarrow (MPC) + 1$	guarda o valor atual estava em Acc no local onde SP Ap
CZ	1, 2, 4, 6, 8, 24	$MPC \leftarrow AB$	retorna para a busca do próximo

$$\begin{array}{ccccccc} & B & & & E & & \\ \hline & | & | & | & | & | & | \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ & & & 7 & D & & \end{array}$$



**Q2. (2,6)** Considere um programa com o trecho abaixo onde a cada linha está indicado o tipo de endereçamento do operando fonte (o destino é sempre modo reg. direto). Para cada instrução defina o valor do EE (endereço efetivo) do operando fonte (é sempre o primeiro) e o resultado final da execução da instrução.

Seus valores iniciais:

R1= 10	
	reg.P= 2000
	SP= 1500
end de aux é = 20	Rx= 2500
(aux) é 30	Rb= 3000
PC= 1000	

Instrução	EE-fonte	conteúdo de R2 ao final
MOVE R1, R2 ; reg. direto	EE = <sup>destino</sup> end R1	$R2 \leftarrow (R1) / R2 = 10$
MOVE (R1), R2 ; reg. indir.	EE = 10	$R2 \leftarrow ((10)) / R2 = (10)$
MOVE aux, R2 ; abs. Direto	EE = 20	$R2 \leftarrow (aux) / R2 = 30$
MOVE (aux), R2; abs. Indireto	EE = 30	$R2 \leftarrow ((aux)) / R2 = (20)$
MOVE #10, R2 ; imediato	EE = 10 X	$R2 \leftarrow 10 / R2 = 10$
MOVE -3(PC), R2 ; relativo	EE = 997	$R2 \leftarrow (PC) - 3 / R2 = (997)$
MOVE (R1)+, R2 ; auto-incremento	EE = 10	$R2 \leftarrow (R1) / R1 \leftarrow (R1) + 1 / R2 = 10$
MOVE -(R1), R2 ; auto-decremento	EE = 9	$R1 \leftarrow (R1) - 1 / R2 \leftarrow (R1) / R2 = 9$
PUSH end ; pilha	EE = 1500	R2 não altera
MOVE P(5), R2 ; paginado por reg.página	EE = 2000 + 5 EE = 2005	$R2 \leftarrow (2005)$
MOVE (Rx(30)), R2 ; indexado indireto	EE = 2500 + 30 EE = (2530) X	$R2 \leftarrow ((2530))$
MOVE 45(Rb), R2 ; registr. Base	EE = 3000 + 45 EE = 3045	$R2 \leftarrow (3045)$
MOVE Rx(Rb), R2 ; reg. base e indexado	EE = 3000 + 2500 EE = 5500	$R2 \leftarrow (5500)$

9,1  
140  
1

24

Q3. (2,4) Sejam o trecho de programa abaixo e a subrotina utilizada pelo mesmo (considere que todas as variáveis utilizadas já foram definidas anteriormente):

X1 = 0	
B = 2	SUB (valor: X1; ref: X2, X3; valor: X4)
C = 7	(1) IF (X1=0) then X2 = 5;
D = 5	else X2 = 3;
CALL SUB(B, C, A,C+D)	(2) X1 = X2 + X1;
A = A-2 A = 8	(3) X3 = 2 * X2;
C = C-1 C = 4	(4) X4 = X3 + X1;
CALL SUB(B, C, A,C+D)	RETURN
.....	

Preencha a tabela abaixo:

	1. chamada				2. chamada			
	A	B	C	D	A	B	C	D
✓ Antes da chamada	X	2	7	5	4	2	2	5
✓ Após (1)	X	2	3	5	4	2	3	5
✓ Após (2)	X	2	3	5	4	2	3	5
✓ Após (3)	6	2	3	5	6	2	3	5
✓ Após (4)	6	2	3	5	6	2	3	5

OBRIGATÓRIO - Explique aqui, sucintamente o seu raciocínio para cada linha da 1. chamada e da 2. chamada (por exemplo escreva os seus cálculos).

Chamada 1

Antes 1  $X_1 = 2$   $X_2 = C = 7$   $X_3 = A = ?$   $X_4 = C + D = 12$

Linha 1  $X_1 \neq 0 \therefore X_2 = 3$ , como é por referência  $C = 3$  (houve alteração)

Linha 2  $X_1 = 3 + 2 = 5$  não há alteração, pois  $X_1$  foi passado como valor

Linha 3  $X_3 = 2 \cdot 3$   $X_3 = A = 6$  houve alteração pois  $X_3$  foi passado por referência.

Linha 4  $X_1 = 3 + 5 = 8$  não há alteração pois  $X_1$  é passado por valor

$$A = A - 2 = 6 - 2 = 4$$

altera-se valor de A

$$C = C - 1 = 3 - 1 = 2$$

Alterar valor de C

Chamada 2

Antes  $X_1 = 2$   $X_2 = C = 2$   $X_3 = A = 4$   $X_4 = 7$

Linha 1  $X_1 \neq 0 \therefore X_2 = 3$ , houve alteração em C,  $\therefore C = 2$

Linha 2  $X_1 = 2 + 2 = 4$  não há alteração pois  $X_1$  é passado por valor

Linha 3  $X_3 = 2 \cdot 3 = 6 \therefore A = 6$ , pois  $X_3$  é passado por ref.

Linha 4  $X_4 = 6 + 4 = 10$  não há alteração pois  $X_4$  é passado por valor

Q4. (2,5) Defina uma macro que trate a soma de até 4 elementos e armazene em um quinto elemento. Use o registrador R3 no interior da macro.

A chamada da macro tem o seguinte formato: SomaM B1,B2,B3,B4,B5

0,7/ (a) (1,0) Defina a macro SomaM:

→ VIRAR FOLHA

0/ (b) (1,0) Realize duas expansões, uma para soma de 2 elementos, outra para 4.

0/ (b1) SomaM X,Y,,Z 2

não use parâmetros  
de expansão  
do seu macro!

```
MOV X, R3
ADD Y, R3
MOV R3, Z
END MACRO
```

Adotando Z  
como a variável  
do resultado.

0

(b2) SomaM ,X,X,,Z

```
MOV X, R3
ADD X, R3
MOV R3, Z
END MACRO
```

0

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

MACROS são sequências de instruções definidas uma única vez no programa e cada vez que é chamada é substituída pela sequência de instruções da definição. O tratamento de macros é feito em tempo de montagem, em que o montador expande, no próprio programa, a macro chamada, ou seja, há uma substituição textual de código fonte para código fonte.

Exemplo: Entrada = chamada da macro  
Saída = substituição textual do macro pelo código definido

MACRO SOMA PAR1, PAR2, RESU

MOV PAR1, R3

ADD PAR2, R3

MOV R3, RESU

END MACRO

0,9

MACRO ~~SOMA~~ ~~M2~~

B1, B2, B3, B4, B5

MOV R0, R3  $\ominus$  -0,1

AIF (B1=" "). teste 2

MOV B1, R3

MOV R3, B5

(.) teste 2 AIF (B2=" "). teste 3

SOMA B5, B2, B5

~~ADD teste 3~~

(.) teste 3 AIF (B3=" "). teste 4

SOMA B5, B3, B5

~~ADD teste 4~~

(.) teste 4 AIF (B4=" "). FIM

SOMA B5, B4, B5

~~ADD FIM~~

(.) FIM

END MACRO

não funciona  
corretamente  
no caso

$\ominus$  -0,2

Z não foi inicializado.

expressão // ~~SOMA~~ X, Y, Z

```
MOV X, R3
MOV R3, Z
MOV Z, R3
ADD Y, R3
MOV R3, Z
```

~~SOMA~~ M, X, Y, Z

```
MOV Z, R3
ADD X, R3
MOV R3, Z
```

reduz