

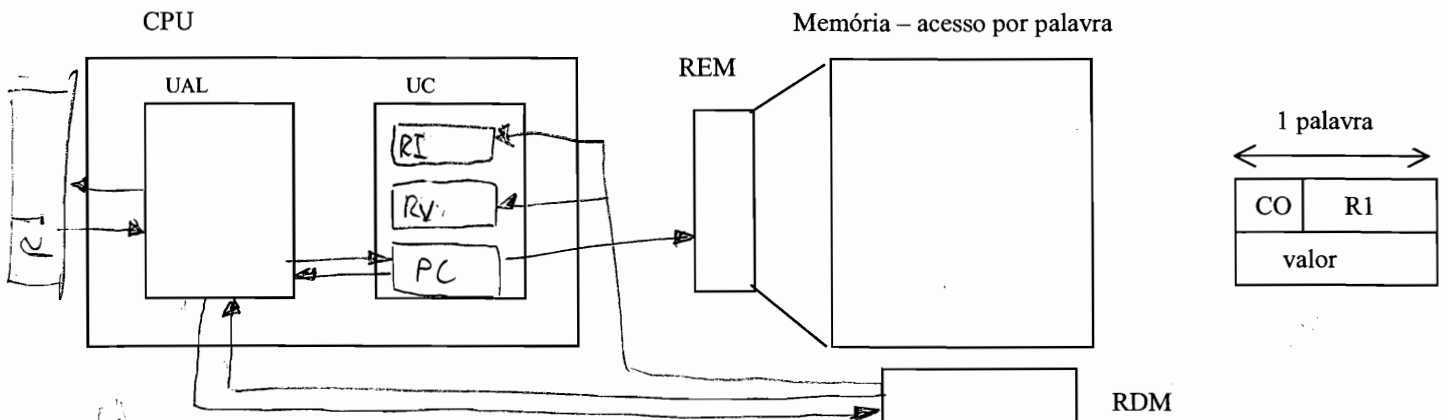
11:45 Q1. 0,8 Q2. 1,2 (1,2) Q3. 0,4 + 2,5 Q4. 1,6 2,5 Q5. 0,9 } 8,25
 Q6. 2,0 } 8,4
 8,4
 37
 EA 869 – Turma A – 1. Semestre 2008
 Prova 3 – 29/04/2008 – Prof. Léo Pini Magalhães
 (consulta a 1 folha A4 que não pode ser fotocópia)
 Nome: Érico Vieira Porto RA: 060428

Q1. (1,0) Na resolução de um problema usando um computador você deve considerar inúmeras questões, computabilidade e complexidade certamente. Levando em consideração o tamanho da entrada, explique com clareza e de forma sucinta computabilidade e complexidade.

Na resolução de um problema [levando em consideração o tamanho da entrada] a computabilidade diz respeito a existência ou não de um algoritmo que resolve o problema; e a complexidade define o desempenho, em relação ao tempo de máquina de um dado algoritmo, em função do tamanho da entrada.

computabilidade não se relaciona com tamanho da entrada.

Q2. (2,0) Considere a representação abaixo de um computador: (represente os regs e caminhos de dados necessários)



Represente no esquema acima e mostre na tabela abaixo (continue no verso da folha se necessário) todos os pulsos e micro-operações para o processamento da instrução:

MOVEIND #3, (R1) ; (R1) ← 3

FZER BUSCA

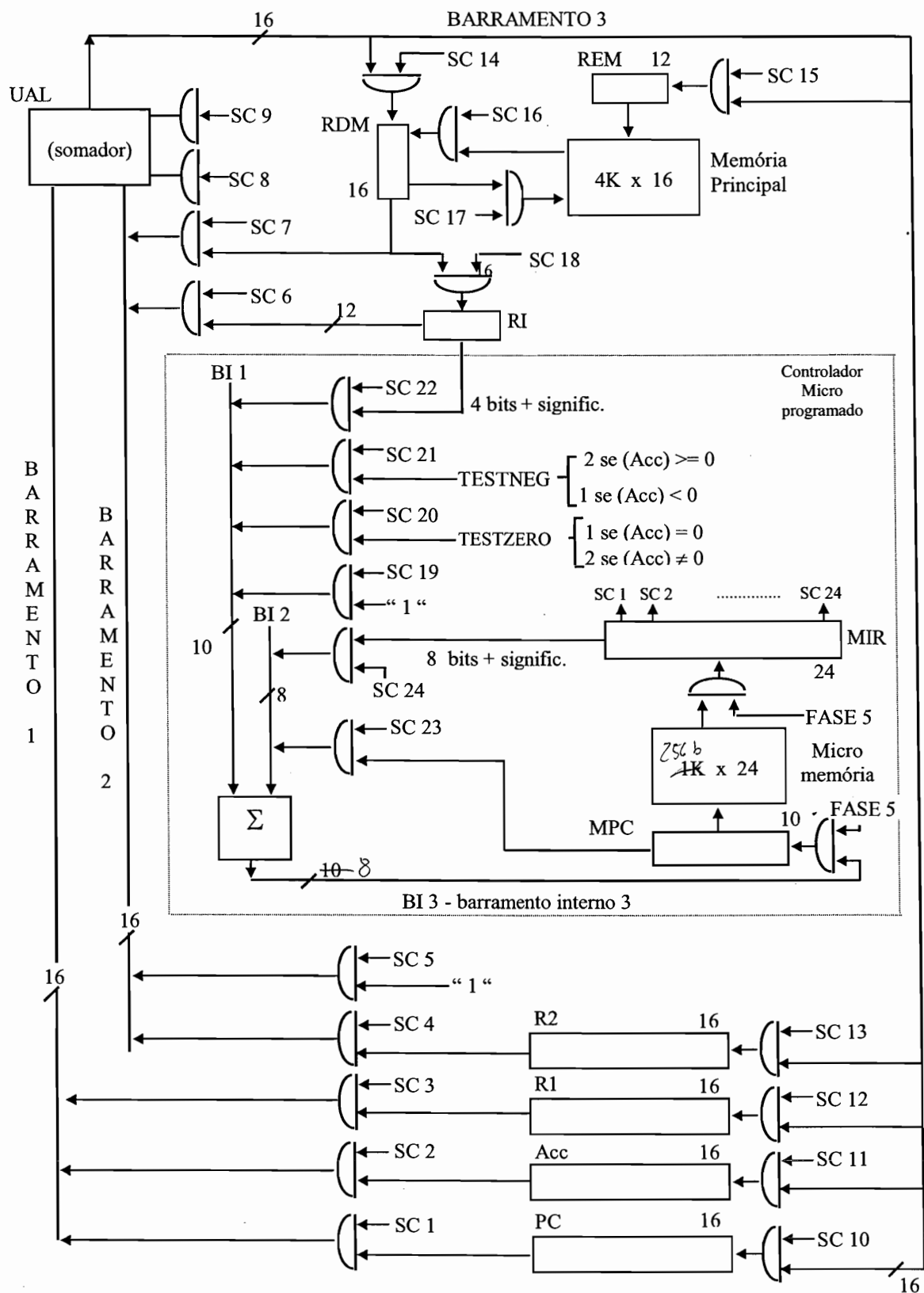
(na tabela use 1 linha para cada pulso de relógio)

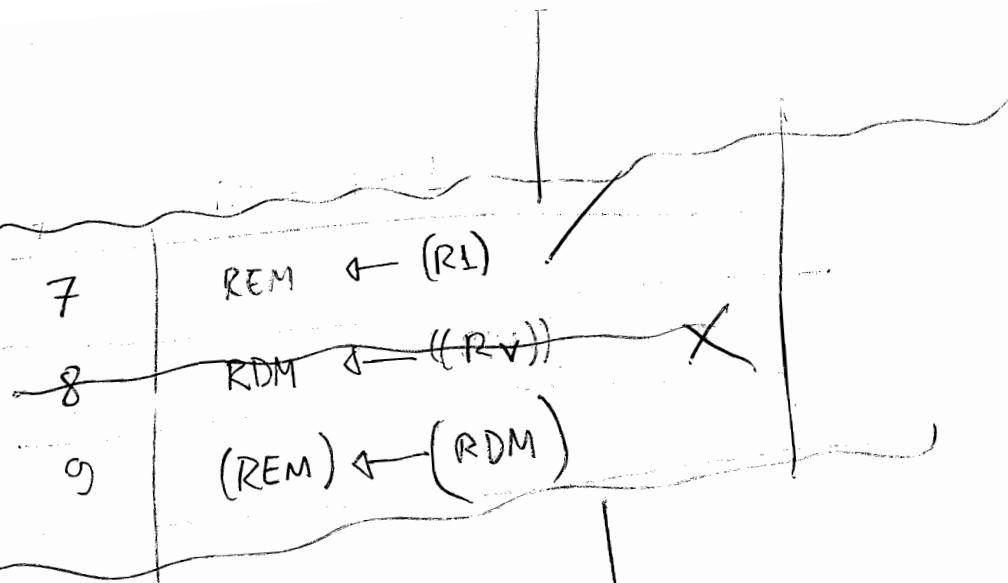
A melhor solução é aquela que demanda a menor qtd de pulsos.

PULSO	MICROOPERAÇÃO
1	REM ← (PC)
2	RDM ← ((REM)) , PC ← (PC) + 1
3	RI ← (RDM)
4	REM ← (PC)
5	RDM ← ((REM)) , PC ← (PC) + 1
6	RV ← (RDM)

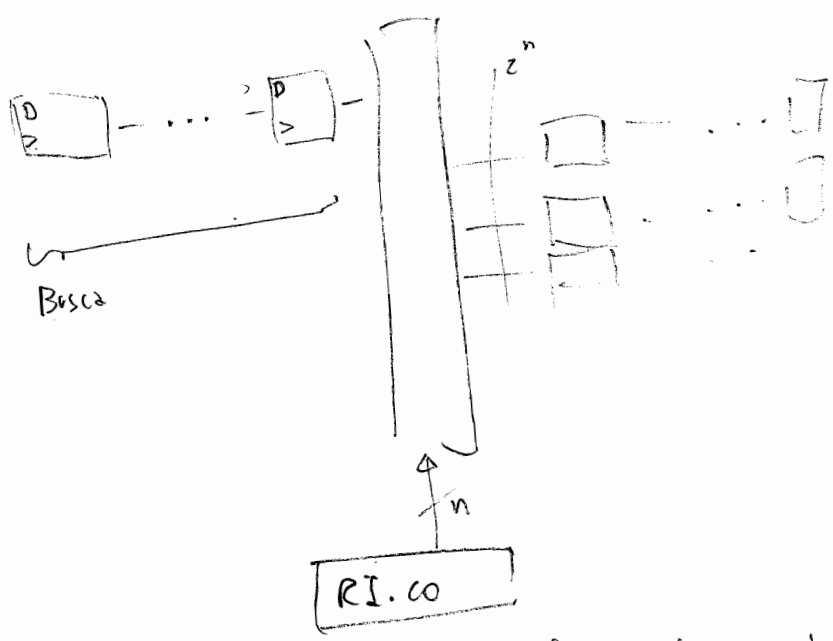
continua no verso

Q3. (2,5) Considere a máquina microprogramada abaixo.





Foi assumido que o controlador é do tipo:



ou!
implementar
p/ a
resposta

Onde as ligações por fio tornam desnecessárias o mapeamento das unidades de controle microprogramadas

TEST NEG
RZ ← (end)

2 | E
000 1110

ACC
REM ← (R1) ; 6, 15
RDM ← ((REM)) ; 16
M 3
RZ ← (RDM) 7, 13
MPC ← (0)

Mostre o microprograma para executar a instrução:

“ler a posição de memória end para R2 se ACC < 0”.

Esta instrução tem tamanho de 16 bits, sendo 4 bits para o CO e 12 bits para end.

- a) Mostre o mapeamento (defina CO como o dígito de unidade de seu RA, se for zero some 4) na posição da micromemória indicada por “AQUÍ”; RA = 06 04 2 8
- b) Inicie o microprograma na posição indicada “PRO”. Esta é a posição de número, em hexadecimal, dE (d é o dígito decimal de seu RA).

Atenção: cada linha da tabela corresponde a 1 endereço da micromemória; a fase de busca já está preenchida.

Endereço micromemória (hexadecimal)	Sinais de controle	Microoperações
0	1,15,16,19,23	REM ← (PC) ; RDM ← ((REM)) ; MPC ← (MPC) + 1
1	1,5,10,18,22,23	PC ← (PC) + 1 ; R1 ← (RDM) ; MPC ← (MPC) + (R1 - CO)
2	3,5,6,7, 24	MPC ← ZE
3	6,15, 16, 19, 23	REM ← (R1, end) ; RDM ← ((REM)) ; MPC ← (MPC) + 1
4	21, 23	MPC ← (MPC) + TEST NEG
5	7, 13	RZ ← (RDM) ; MPC ← 0
6	15, 16, 19, 23	MPC ← 0
7	?	?
8	?	?

0,25 AQUI

0,2 PRO

90

30

entendi!

94
213

Q4 (2.0) A tabela a seguir fornece um trecho de um programa explicitando as formas de endereçamento utilizadas:

1 →
2 →
3 →
4 →
5 →
6 →

PC	C.O.	Operando fonte		Operando destino		Mnemônico
		modo	endereço	modo	endereço	
500	MOVE	registrador direto	R1	registrador indireto	R2	MOVE R1, (R2)
501	MOVE	absoluto direto	60	indexado	10	MOVE 60, RIX(10)
502	MOVE	imediato	3	relativo	7	MOVE #3, 7(PC) (**)
503	MOVE	baseado	50	absoluto indireto	510	MOVE RB(50), (510)
504	MOVE	autodecremento	R2	absoluto direto	3	MOVE -(R2), 3
505	PUSH	registrador direto	R2	pilha	SP	PUSH R2

() lembre-se de usar o PC já incrementado para calcular o endereço**

Seguindo o programa acima, mostre a seguir os valores dos elementos indicados no instante (1 a 6) assinalado.

Depois de Elemento	1	2	3	4	5	6
PC=500	501	502	503	504	505	506
R1=50	50	50	50	50	50	50
R2=60	60	60	60	60	59	59
end. 50 = 10	10	50	50	50	50	50
end. 60 = 20	50	50	50	50	50	50
RIX = 40	40	40	40	40	40	40
end. 510 = 5	5	5	3	50 ²	50 ³	50 ³
reg. base = 0	0	0	0	0	0	0
end. 3 = 200	200	200	200	200 ⁵⁰	8	8
end. 59 = 8	8	8	8	8	8	59
SP = 59	59	59	59	59	59	60

1,625

Q5. (1,0) Unidades de controle podem ser implementadas exclusivamente "por hardware" ou microprogramadas. **Explique com clareza e de forma sucinta** como se dá a implementação de cada opção e cite, em sua opinião, a (uma) principal vantagem e a (uma) principal desvantagem de cada opção.

0,9
A unidade de controle pode ser implementada por hardware utilizando uma sequência de flip-flops que implemente a fase de busca e escolha, dando o código de operação, entre sequências de flip-flops que executem a instrução e ao final retorne a busca. Essa implementação liga as saídas dos flip-flops da unidade de controle diretamente a VAL e outros elementos da máquina. [Uma vantagem dessa máquina é que pode ser facilmente implementada em sistemas bastante simples], porém alterar a arquitetura caso necessário é extremamente trabalhoso e custoso.

confusões: velocidade
Ao ser microprogramada, a Unidade de controle utiliza de uma memória para armazenar os microcomandos antes executados por flip-flops. Dessa maneira a máquina pode ser projetada sistematicamente, tendo como única desvantagem uma perda de velocidade devido aos ciclos de acesso à sua memória de microprogramas.

Q6. (2,0) O programa abaixo é composto por um programa principal e duas rotinas.

a = 9
b = 3
c = 2
(0) CALL SUB1(a,d,b*c,c)
:
:

SUBROUTINE SUB1(ref:x,y; valor:w,z)
(1) $y = w + z = 9$
(2) IF $y > x$ THEN $y = x$ ELSE $x = y$ $8 > 9$
(3) $w = x$ $w = 8$
(4) $x = w + y$ $x = 8 + 8 = 16$
CALL SUB2 (20, x+3)
RETURN

SUBROUTINE SUB2(ref:s; valor:t)
 $s = s * t$
IF $s > 10$ THEN $s = s / t$ ELSE $s = t$
RETURN

20
a. (1,5) Preencha a tabela a seguir com os valores correspondentes das variáveis indicadas:

	a	b	c	d
imed. antes de (0)	9	3	2	?
imed. após (1)	9	3	2	8
imed. após (2)	8	3	2	8
imed. após (3)	8	3	2	8
imed. após (4)	16	3	2	8

b. (0,5) Há algum erro nas passagens de parâmetros? Indique, se for o caso, qual é o erro e discuta-o.

Há um erro na passagem de parâmetro para a subrotina SUB2, onde a chamada ref insere o constante 20, onde deveria haver uma variável a ser referenciada.