

Q1. $0,75 + 0,25$
 Q2. $0,5 + 0 + 0,5$
 Q3. \emptyset

Q4. $2,0$
 Q5. $0,8$

Q6. $1,1$
 Q7. $0,5$

47

6,3

EA 869 – Turma U – 1. Semestre 2006

Prova 1 – 05/4/2006 – Prof. Léo Pini Magalhães

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

Nome: Thiago Cesar Caloni Esteves

Número: 036210

Q1. (1,0) Na resolução de um problema você pode optar por 3 algoritmos que têm as seguintes funções de complexidade temporal:

- Algoritmo 1: $A + B \cdot n$
- Algoritmo 2: $C + D \cdot \log_2 n$
- Algoritmo 3: $E + F \cdot n^2$ (sendo n o tamanho da entrada e A, B, C, D, E, F ctes pertencentes a \mathbb{N}^*)

- (a) (0,75) torne o algoritmo 3 a solução mais apropriada (mais rápida) para $n=8$. Justifique a sua resposta;
 (b) (0,25) mostre graficamente o comportamento da complexidade dos algoritmos em função de n (de $n=0$ até ∞) para as constantes escolhidas em (a). Indique qual(is) algoritmo(s) você escolheria em função de n .

0,75

a) Algoritmo 1: $A + 8B$
 Algoritmo 2: $C + 3D$
 Algoritmo 3: $E + 64F$

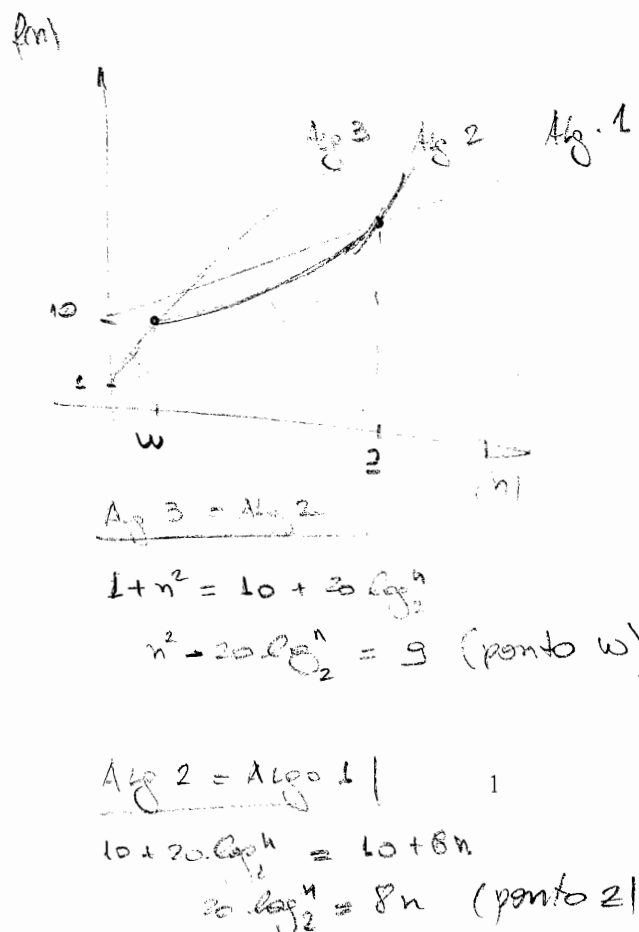
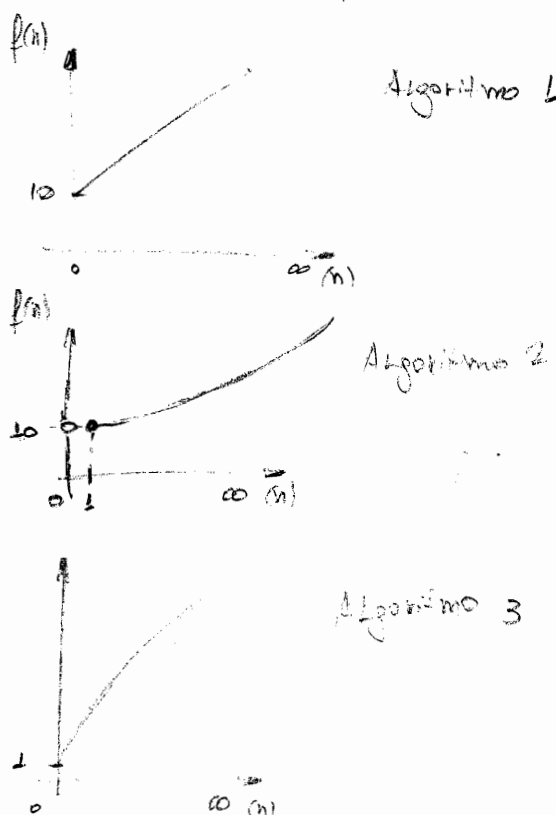
para $n=8$

Assim, para que o Algoritmo 3 seja a solução mais apropriada devemos ter $A=10$; $B=8$;
 $C=10$; $D=20$; $E=F=1$,

Alg 1 teria resultado $\Rightarrow 74$
 Alg 2: teria resultado $\Rightarrow 70$
 Alg 3: teria resultado $\Rightarrow 65$

Portanto, para $n=8$ a menor valor de n , a solução mais apropriada seria o Algoritmo 3, pois $A, \dots, F > 0$

0,25



Pod. Ben

Frederic Smith

}

East of ...

Q2. (1,5)

(a) problemas computáveis são aqueles que possuem um algoritmo que o resolve, ou seja, possui um procedimento que o resolve em um número finito de passos.

(b) problemas não-tratáveis são aqueles que, basicamente, possuem um algoritmo que o resolve com complexidade polinomial.

(c) problemas NP e P são aqueles (comentar computabilidade e complexidade)

Os problemas do tipo NP (não-polinomial) e P (polinomial) são aqueles que não computáveis, ou seja, possuem um algoritmo que os resolve, porém os do tipo NP são aqueles que demandam algoritmos com complexidade exponencial e os do tipo P são compostos por problemas

Q3 (1,5) Considere a estrutura de dados de tipo PILHA. Descreva um algoritmo que insira (escreva) 1 dado nesta estrutura. Mostre a sua solução através de um diagrama de blocos. Aborde os casos de pilha cheia e esta melhor vazia.

X é a quantidade de elementos que a pilha suporta de 0 a 99, ele guardava o valor do

A ponteiro, inicialmente em 0.

Solução é do ordem (de) polinomial.

Escreve PILHA:

Inicio

SIM NÃO

Escreve o dado no endereço 0

Incrementa $X++$

retorna

$Y = X$

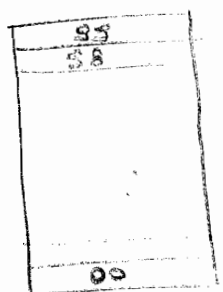
escreve dado de $(Y+1) \rightarrow Y$

Decrementa $Y--$

$Y = 0$?

$X++$

Escreve dado no endereço



Esta função quer dizer que ela pega o conteúdo que está no endereço $Y+1$ e o SALVA no endereço Y .

funcionando a pilha?

Então,

para o intervalo entre "n" variando de

$0 - w$, temos o algoritmo 3 que melhor
descreve o melhor resolver.

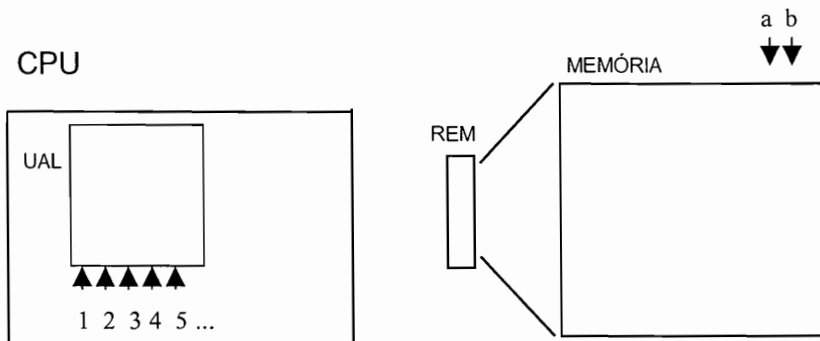
$w - z$ temos o algoritmo 2 | não! Reveja!

$z - \infty$ temos o algoritmo 1

ideia consiste em
reveja o crescimento de
 $n + \log_2 n$ para $n \rightarrow \infty$

2

Q4. (2,0) Considere a arquitetura abaixo similar à apresentada em aula:



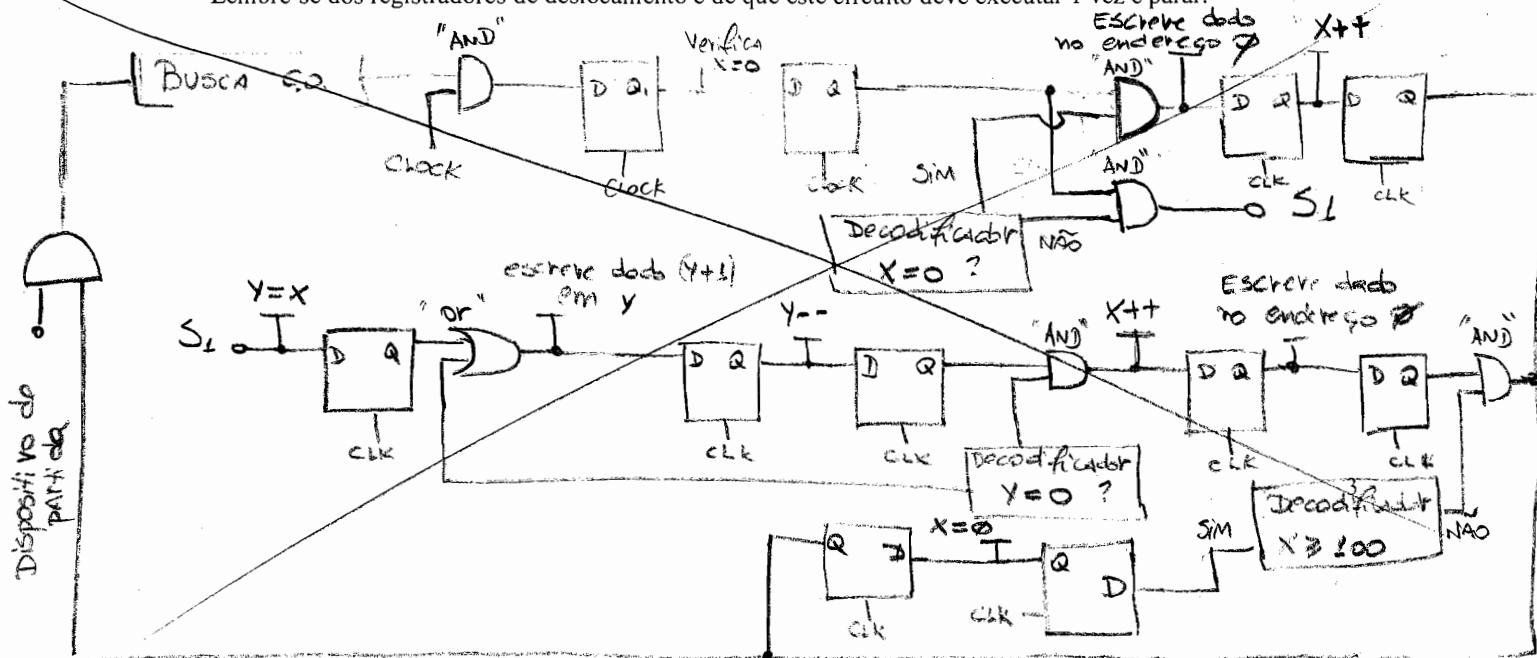
Os seguintes sinais de controle estão definidos: (não temos nesta arquitetura o reg. RDM e a UC não é mostrada)

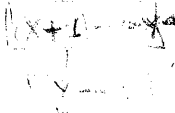
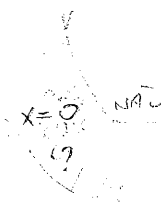
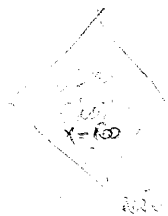
- 1: $TMP \leftarrow (\text{barr. dados})$ 3: $Acc \leftarrow (Acc) + 1$ 5: $Acc \leftarrow (\text{barr. dados})$ a: memória $\leftarrow (\text{barr. dados})$
 2: barr. dados $\leftarrow (TMP)$ 4: complementa Acc 6: barr. dados $\leftarrow (Acc)$ b: barr. dados $\leftarrow (\text{memória})$

Utilize a tabela abaixo para definir uma sequência de microcomandos que obtenha o complemento de dois da posição de memória de interesse (já contida em REM) e armazene-o na mesma posição de memória.

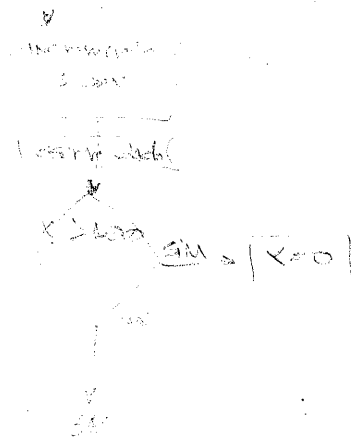
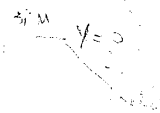
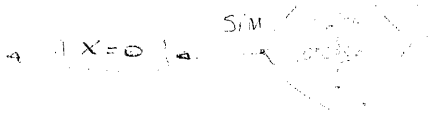
Relógio	Microcomandos	Comentários (OBRIGATÓRIO)
1	barr. dados $\leftarrow (\text{memória})$ $Acc \leftarrow \text{barr. dados}$	transfere o conteúdo da memória PARA Acc
2	Complementa Acc	faz a primeira etapa para fazer o complemento de 2.
3	$Acc \leftarrow (Acc) + 1$	faz a segunda etapa para fazer o complemento de 2.
4	barr. dados $\leftarrow (Acc)$ a: memória $\leftarrow \text{barr. dados}$	transfere o conteúdo de Acc, ou seja, o complemento de dois da posição de memória desejada, para a mesma posição de memória
5		
6		

Q5. (1,0) Mostre uma possível implementação de uma UC específica para sequenciar a sua solução de Q3. Lembre-se dos registradores de deslocamento e de que este circuito deve executar 1 vez e parar.

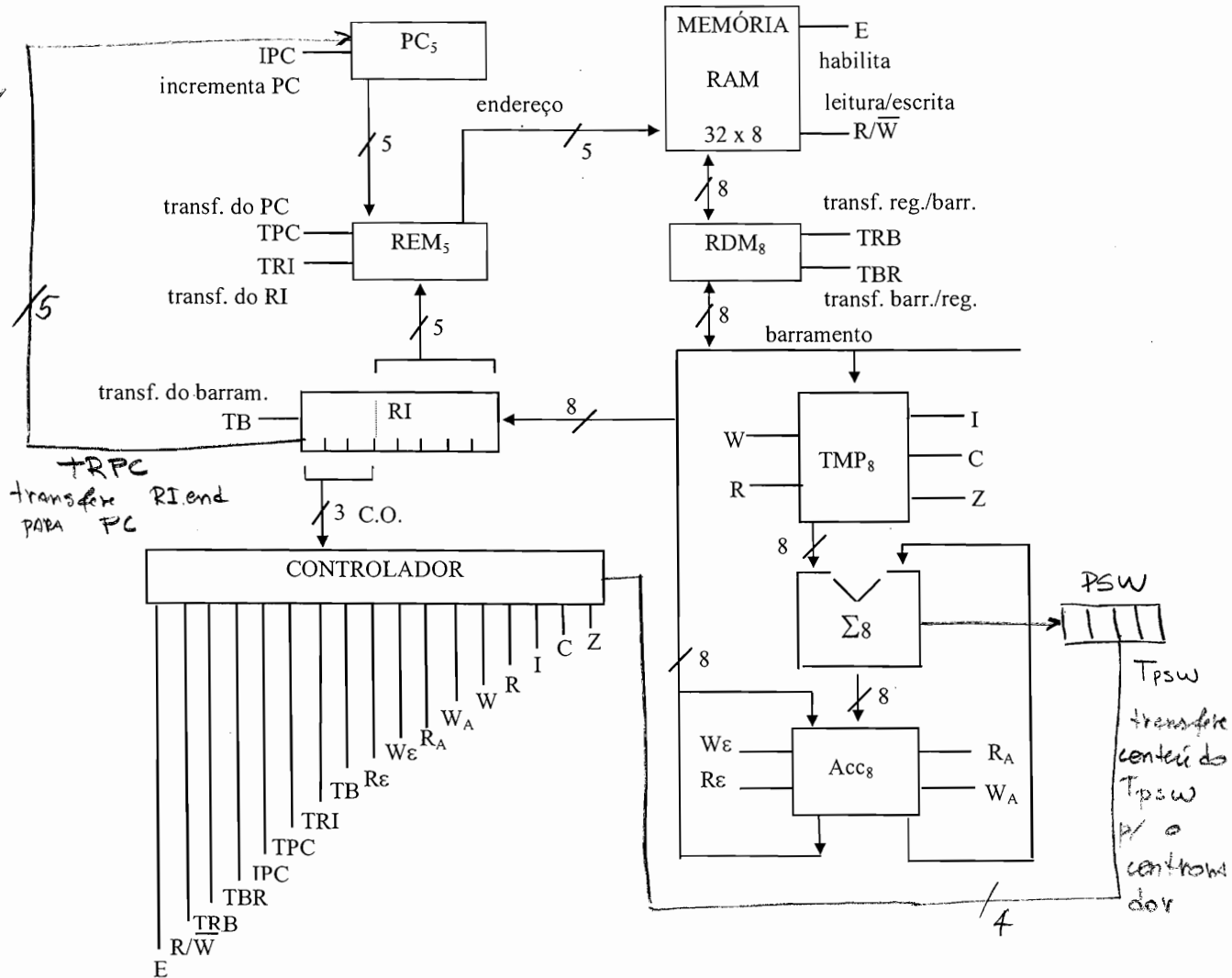




... ! ...



Q6. (1,5) Para o esquema a seguir: (instrução com 8 bits, 3 dígitos mais significativos são o Cód. Operação)



Acrescente à arquitetura um registrador de estado – PSW – que possui um campo com 4 flags indicando o ocorrido na última operação realizada. Com este elemento a arquitetura poderá realizar operações de desvio condicional (desvio se positivo, se negativo, etc.).

Para a instrução **JNZ end** (desvio de execução para a instrução no endereço “end” se operação anterior resultou em número não zero):

(a) Defina o PSW adequadamente na figura acima e possíveis novos microcomandos para JNZ e ligações;

(b) Preencha o quadro abaixo, busca e execução, para a execução da instrução JNZ.

Justifique a sua solução (após a tabela e/ou no verso).

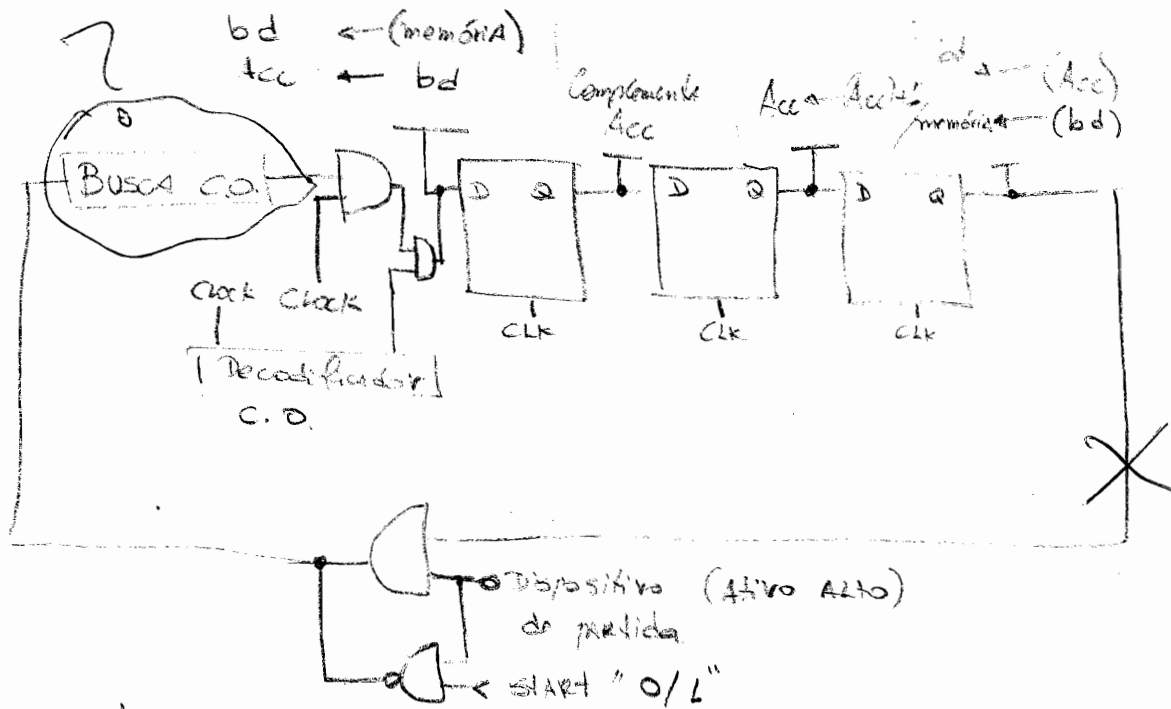
JNZ

Relógio	Microcomandos	Microoperações
1	TPC	$REM \leftarrow (PC)$
2	E / R/W / IPC	$PC \leftarrow (PC) + 1$
3	TRB	$RDM \leftarrow ((REM))$
4	TRB TBR TB	$RI \leftarrow (RDM)$
5	TRI	$REM \leftarrow (RI\ end)$
6	TRB	$RDM \leftarrow ((REM))$
7	+psw	controlador $\leftarrow (PSW)$
8	+RPC	$PC \leftarrow (RI\ end)$

busca
0,2/0,5

teste?
0,25/0,5
0,25/0,75

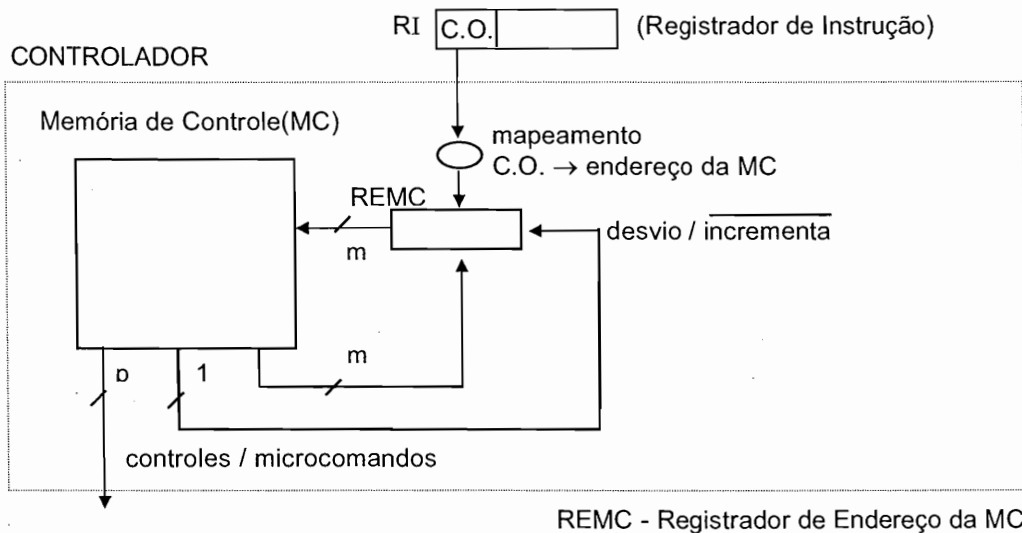
bd \rightarrow hormonal de
dados



Assim, a UC aguardará o dispositivo de partida sempre que o sistema terminar a execução. Quando ligamos o sistema, start está em "0" e assim ele dará o primeiro ciclo. Após isso, start vai para "1" e permanece neste valor sem se alterar, fazendo com que o dispositivo de partida comande os próximos ciclos.

0.8

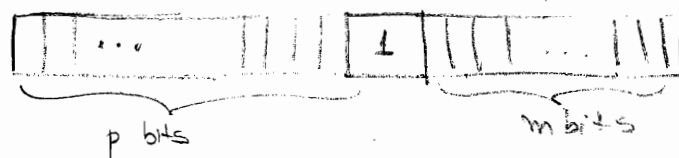
Q7. (1,5) Retome a questão Q5 e lembre-se do controlador microprogramado de Wilkes (abaixo).



Explique, respondendo aos 3 itens abaixo, como se daria o controle (sequenciamento dos microcomandos) em Q5 usando este controlador microprogramado.

- defina uma microinstrução típica (com $p+1+m$ bits). Consulte a Q4 como referência.
- escreva o microprograma que substitui o controlador da Q5.
- comente o funcionamento.

0,5/0,5 a)

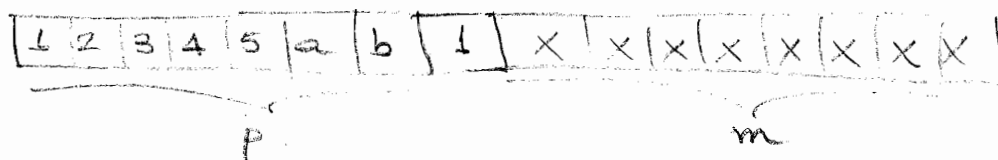


No caso teríamos $p=8$, pois este é o número de microcomandos possíveis na CPU



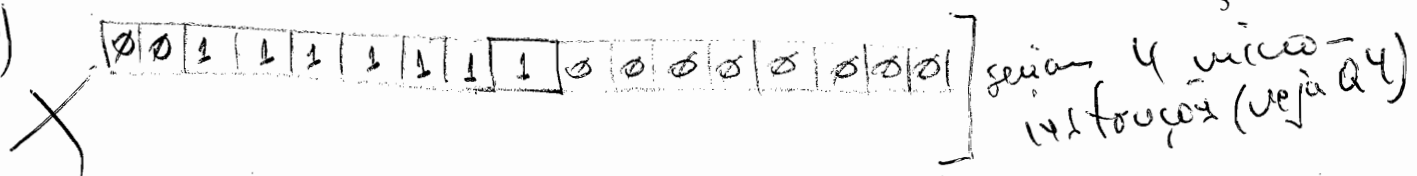
No caso, m é igual ao próximo endereço no caso de desvio, se não, o próximo endereço será o (anterior + 1). Para a questão m pode ser qualquer valor pois não temos restrição no tamanho do mesmo. Adoptamos 8 bits

Ani m :



esta é a nossa microinstrução típica para este controlador.

b)



a) As ligações estão feitas na figura e
os novos comandos podem ser.

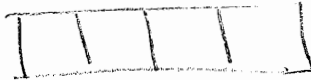
JNN end (desvia pt posição "end" se o
resultado da última operação
for negativo)

JPP end (desvia se resultado for
positivo)

JNO end (desvia se houve over-flow)

JNP end (desvia se resultado um número
não zero)

Assim: PSW



possuem 4 bits para indicar
cada uma das operações
possíveis acima.

b) Outras tabelas

