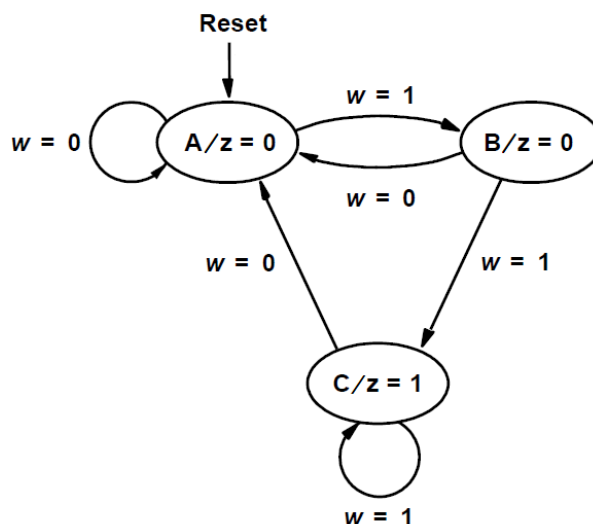


RA:	Nome:	Entrega
Lista de Exercícios Nº 8		(ver site)

Assunto coberto: capítulo 8 (Máquinas de estados), Memórias.
Questões que devem ser feitas no Quartus: Nenhuma.

Q1.

Considere a máquina de estados representada no diagrama abaixo. Projete o circuito desta máquina usando lógica combinacional e flip-flops JK.



Q2. (BV 8.27)

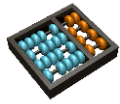
Projete e simule um circuito tipo contador, controlado por uma entrada w . Se $w = 1$, o contador soma 2 ao seu conteúdo, até atingir 6 ou 7. Atingindo 6 o próximo valor deve ser 0, caso atinja 7 o próximo é 1. Se $w = 0$, o contador subtrai 1 de seu conteúdo, agindo com contador decrescente normal. Use somente flip-flops JK neste exercício. (observação: este exercício foi feito na LE7 usando FFs tipo D).

Q3.

Projetar um conversor de 7 segmentos (entrada de 1 dígito BCD e 7 saídas para os LEDs do display de 7 segmentos) conforme dado em aula (ver slides), utilizando a abordagem com ROM. Escolher as dimensões da ROM a ser utilizada e definir o conteúdo das linhas para implementar as funções especificadas. Comentar o que pode ser feito com as condições de *dont care*.

Q4.

Suponha que você está isolado numa ilha deserta e seu único passatempo é um computador IBM PC XT com um processador Intel 80286, encontrado abandonado dentro de uma cabana construída com folhas de bananeira. Infelizmente, este autêntico modelo de 1983 sofreu um problema de fabricação de modo que seu



processador consegue trabalhar apenas com 20 bits de endereçamento da memória (sendo que o modelo Intel 80286 é capaz de trabalhar com 24 bits de endereçamento).

Você conseguiu abrir o gabinete do computador, depois muito bater com um coco, e averiguou que havia apenas um pente de memória SIMM de 256 KB com oito bancos (chips) de memória de 32KB cada (como visto na Figura 4-1).



Figura 4-1: Memória de 256 KB com oito bancos (32KB cada) .

Entediado, você decidiu escrever um programa em C. Após gerar o binário você decidiu executar o comando `dump.exe` do MS-DOS para analisar o código de seu executável. Você percebeu que um vetor de dados de seu programa, de tamanho 8KB foi alocado a partir do endereço $0x0F000_{16}$ na memória. Por curiosidade, você se fez as seguintes perguntas:

- Como ficou o espaço de endereçamento, ou seja, quais regiões do espaço completo de endereçamento (em função do número de linhas de endereço) estão apontando para memória existente e quais estão apontando para regiões vazias?
- Qual é a faixa de endereços que aponta para cada chip de memória?
- Como seria o projeto do decodificador de endereços para implementar o mapeamento de endereços verificado?
- Considerando que o vetor de seu programa ocupe endereços seqüenciais crescentes, onde ele ficaria alocado (quais chips)? Em quais linhas dos chips de memória?