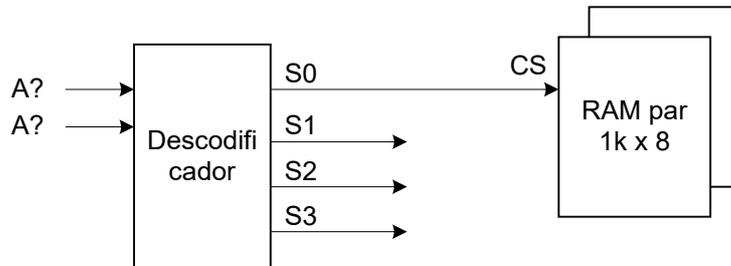


- 1 - Considere o circuito seguinte, que representa apenas uma parte do circuito de descodificação de endereços de um computador com base no PEPE (processador de 16 bits com endereçamento de byte). Cada uma das RAMs (par e ímpar) tem 1k células de 8 bits. Os bits do barramento de endereços que ligam ao descodificador devem ser os de menor peso possível que permitem que o sinal S0 esteja activo durante a gama de endereços necessária para aceder a todas as células das RAMs. Os detalhes da selecção de RAM par ou ímpar estão omitidos por não serem relevantes para este problema.



- a) - Indique quais os bits do barramento de endereços que devem ligar ao descodificador nestas condições. Justifique.

A12 e A11.

Cada RAM precisa de 10 bits para seleccionar uma das suas 1K células. O A0 é usado para distinguir entre RAM par e ímpar (no acesso de byte, com MOVB). Logo, o conjunto das duas RAMs, que tem 2K bytes no total (par + ímpar), precisa dos A10 a A0 para seleccionar todas as células.

Ligando o A12 e A11 no descodificador, que são os seguintes ao A10, garante-se que cada saída fique activa em 2 K endereços.

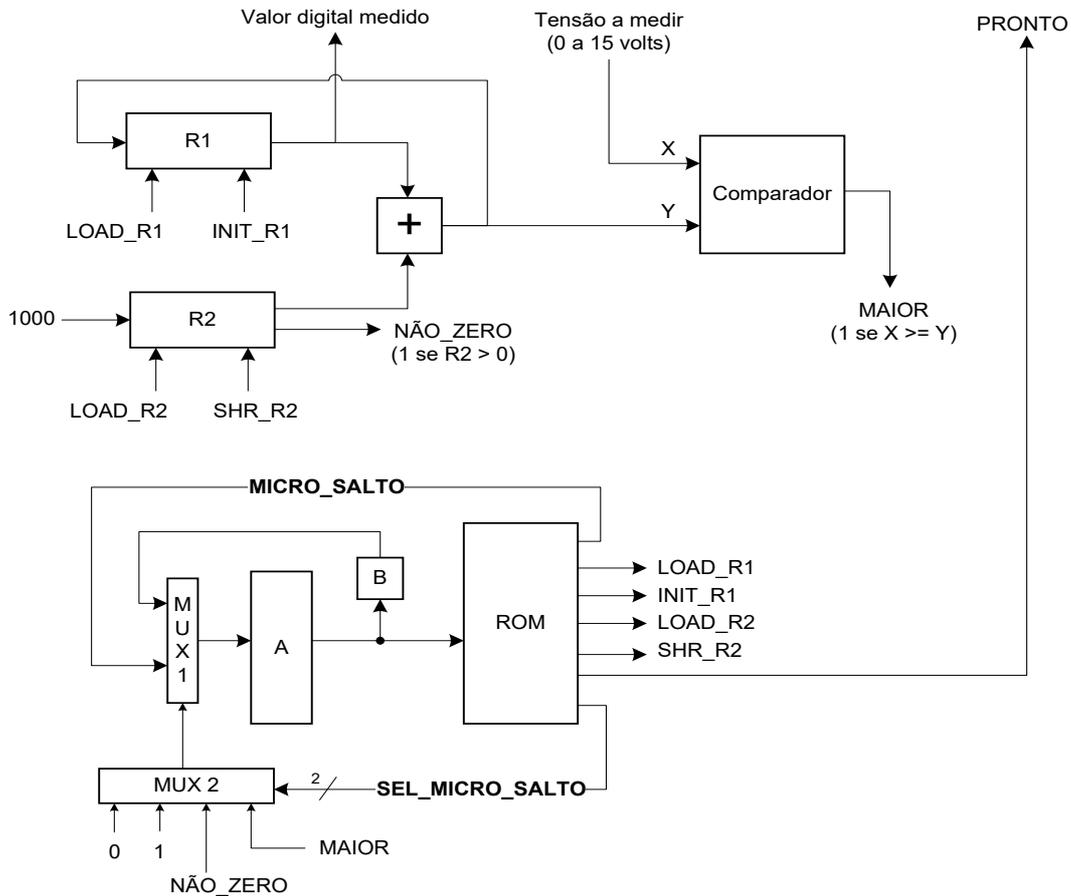
- b) - Se o processador aceder ao endereço 1234H, qual o sinal de saída do descodificador que fica activa? Justifique.

400H são 1K e 1000H são 4 K.

1234H é algures entre 4 K e 5 K. A saída S0 fica activa entre 0 e 2K-1, S1 entre 2K e 4K-1, e S2 entre 4K e 6K-1.

Logo, a saída activa é S2.

- 2 - Considere o circuito seguinte, que implementa um voltímetro digital (medidor de tensão de 0 a 15 volts, em que o resultado é apresentado como um número binário de 4 bits). O voltímetro funciona por aproximações sucessivas, comparando a tensão de entrada com um resultado parcial cada vez mais perto do valor final. Começa por comparar com metade da escala (1000), depois com 1100 ou 0100 (consoante o resultado da comparação) e assim sucessivamente para afinar o resultado.



O circuito é controlado por uma unidade microprogramada, em que:

- LOAD_R1 e LOAD_R2 memorizam valores em R1 e R2 (registos de 4 bits);
- INIT_R1 inicializa R1 a 0 e SHR_R2 desloca R2 de um bit para a direita;
- NÃO_ZERO e MAIOR são entradas para a unidade de controlo e estão descritas no desenho;
- O sinal PRONTO indica quando a medição estiver pronta (quando R2=0);
- O multiplexer MUX1 selecciona a entrada de baixo se o valor à saída do multiplexer MUX2 for 1.

a) - Identifique cada um dos blocos A e B nesta figura.

Bloco	O que é, para que serve
A	MPC, registo que mantém o endereço da microinstrução corrente na ROM de microcódigo
B	Somador de 1 unidade, destinado a incrementar o MPC para endereçar a microinstrução seguinte

b) - Preencha a tabela seguinte com as microinstruções necessárias para implementar a funcionalidade da divisão inteira. Use “SIM” para indicar os sinais que devem estar activos em cada ciclo de relógio, e deixe em branco as células dos sinais não activos ou não relevantes. Em SEL_MICRO_SALTO, identifique a entrada do multiplexer a seleccionar (quando relevante).

Endereço na ROM	Operações	LOAD_R1	INIT_R1	LOAD_R2	SHR_R2	PRONTO	SEL_MICRO_SALTO	MICRO_SALTO
0	Inicializa R1 (com 0) e R2 (com 1000, em binário)		Sim	Sim				
1	Se MAIOR = 1, salta para o passo 3						MAIOR	3
2	Memoriza em R1 a sua entrada (R1 + R2)	Sim						
3	Desloca R2 para a direita				Sim			
4	Se R2 for diferente de 0, salta para o passo 1						NÃO_ZERO	1
5	Activa PRONTO (acabou)					Sim		

c) - Quantos bits deve ter cada palavra da ROM? Justifique.

10 bits. Todos os sinais gerados têm 1 bit, excepto SEL_MICRO_SALTO que tem 2 e MICRO_SALTO, que tem 3 (há 6 microinstruções).

3 - Suponha que o PEPE (processador de 16 bits com endereçamento de byte) inclui uma *cache* de dados de mapeamento directo com blocos de 4 palavras e uma capacidade de 1k bytes (só a parte dos dados propriamente ditos). V=1 se a linha tiver um valor válido e M=1 se alguma das palavras dessa linha tiver sido modificada.

V	M	Etiqueta	Palavras
...			

a) - Quantas linhas tem a cache? Justifique.

Cada bloco tem 4 palavras ou 8 bytes. Com uma capacidade de 1K bytes, a cache tem 128 blocos (1K / 8), ou 128 linhas.

b) - Suponha que a *cache* está vazia quando o processador faz um acesso de 16 bits em leitura ao endereço 18C4H. Indique o nº da linha da *cache* que é alterada na linha em baixo o conteúdo que fica nessa linha (no caso das palavras, escreva o endereço de cada palavra em vez do seu valor, só para se perceber que palavra fica lá escrita). Indique todos os valores em binário ou hexadecimal.

Nº da linha alterada, em hexadecimal

18H

V	M	Etiqueta	Palavras (endereço maior do lado direito)			
1	0	000110	18C0H	18C2H	18C4H	18C6H

- 4 - Suponha que o PEPE, com os seus 16 bits de endereço e endereçamento de byte, suporta memória virtual com páginas de 256 bytes. Assuma que a memória física é de 4 Kbytes e que a TLB é uma *cache* totalmente associativa de 4 entradas, cujo conteúdo é numa dada altura o seguinte:

Válida	Página alterada	Nº de página virtual	Nº de página física
1	1	24H	03H
0	0	7CH	03H
1	0	63H	0FH
0	0	E8H	08H

- a) - Preencha a tabela seguinte (valores em decimal ou hexadecimal):

	Valor	Breve justificação
Dimensão do espaço virtual	64 K bytes	16 bits de endereço
nº de páginas virtuais	256	64K / 256 (dimensão de cada página)
nº de páginas físicas	16	4K / 256

- b) - Acabe de preencher as correspondências da tabela seguinte:

Endereço virtual	Endereço físico
63CEH	0FCEH
24A6H	03A6H
241EH	031EH
630EH	0F0E H

- c) - Suponha agora que o processador faz uma leitura ao endereço virtual 23E8H. Indique na tabela da página anterior as alterações que terão de ocorrer na TLB (risque e escreva os novos valores, arbitrando o que for necessário) para que o acesso seja efectuado e justifique sucintamente essas alterações na caixa em baixo.

Válida	Página alterada	Nº de página virtual	Nº de página física
1	1	24H	03H
1	0	23H	05H
1	0	63H	0FH
0	0	E8H	08H

A página virtual 23H não está carregada na TLB, pelo que no acesso dá TLB miss. Usando uma das entradas livres, tem de se carregar na TLB a correspondência entre a página virtual 23H e a página física onde está carregada em memória principal (05H, no exemplo), assumindo que esta já está carregada em memória principal (senão dá *page-fault* e isto tem de ser resolvido primeiro).