



NOME		NÚMERO	
------	--	--------	--

1. (1,5+1,5+2 valores) Considere o seguinte programa, que deve completar. Preencha apenas as linhas que entender serem necessárias.

ECRA	EQU	8000H
PLACE	2000H	
pilha:	TABLE	100H

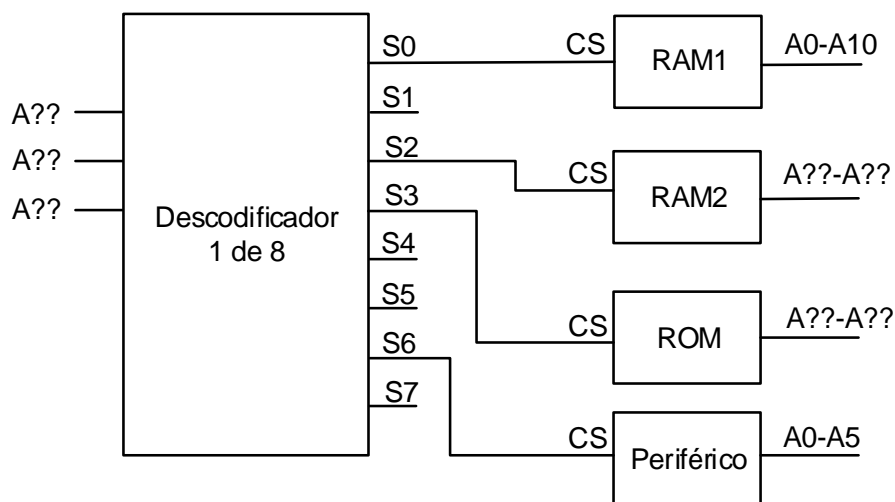
PLACE	0	
	MOV SP,	
	MOV BTE,	
	MOV R2,	ECRA

rotInt0:		
rotInt2:		

- a) Do lado esquerdo, complete a zona de dados e o programa principal com o necessário para a pilha e as interrupções 0 e 2 funcionarem corretamente. O corpo do programa principal deve terminar em ciclo infinito (salto para a própria instrução);
- b) Do lado direito, programe as rotinas de interrupção, que alteram o primeiro byte do ecrã (PixelScreen, periférico de 8 bits), escrevendo simplesmente um valor no respetivo endereço. A rotina de interrupção 0 escreve AAH nesse endereço (■■■■■). De igual modo, a rotina de interrupção 2 escreve 55H no mesmo endereço (■■■■■). O efeito visual no ecrã é os pixels oscilarem entre uma posição e outra, ao ritmo das interrupções. O endereço do ecrã mantém-se no R2 durante todo o programa;
- c) Suponha agora que os pinos do PEPE da interrupção 0 e 2 estão ligados ao mesmo sinal externo, com período de 1 segundo. Qual é o efeito visual no ecrã (o que é que o utilizador vê, concretamente)? Justifique.

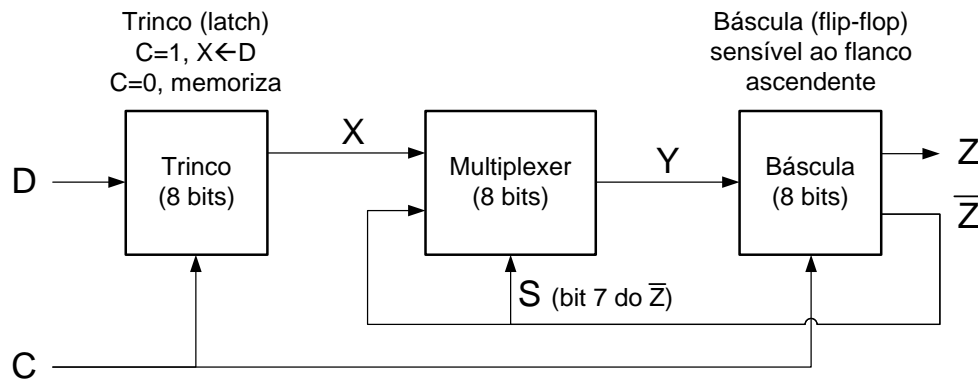
2. (1 valor) Uma transmissão de dados por um barramento série assíncrono, com bit de paridade e 2 stop bits, não pode demorar mais de 2 minutos a transmitir 100.000 bytes de dados. Indique qual o ritmo de transmissão necessário (em bits/seg), e se esse ritmo é o mínimo ou o máximo para que esta transmissão ocorra de acordo com os requisitos. Justifique.

3. (2 valores) Considere o seguinte sistema de decodificação de endereços utilizado por um processador de bus de dados de 8 bits e bus de endereços de 16 bits. Preencha a tabela com os bits de endereço a que cada dispositivo deve ligar, a sua capacidade (decimal) e os endereços de início e de fim (em hexadecimal) em que esse dispositivo está ativo (não considerando endereços de acesso repetido - espelhos).



Dispositivo	Bits de endereço	Capacidade (bytes) (decimal)	Início (hexadecimal)	Fim (hexadecimal)
Decodificador				
RAM1	A0-A10			
RAM2		1 Ki	2000H	
ROM		4 Ki		
Periférico	A0-A5			

4. (2 valores) Considere o seguinte circuito, em que os sinais D, X, Y e Z são barramentos de 8 bits, C é o *clock* (tanto do trinco como da básica) e S é o sinal de seleção do *multiplexer* (S=0 seleciona a entrada X). Assumindo que os sinais D e C evoluem ao longo do tempo da forma indicada na tabela seguinte, acabe de preencher o resto da tabela (escreva todas as células, mesmo que o valor se mantenha). Todos os valores de 8 bits estão representados em hexadecimal (não é preciso colocar o H).



D	3A		7B		53		4E		B6		7D		1F
C	1	0	0	1	1	0	0	1	1	0	0	1	1
S													
X													
Y													
Z	5E												
\bar{Z}													

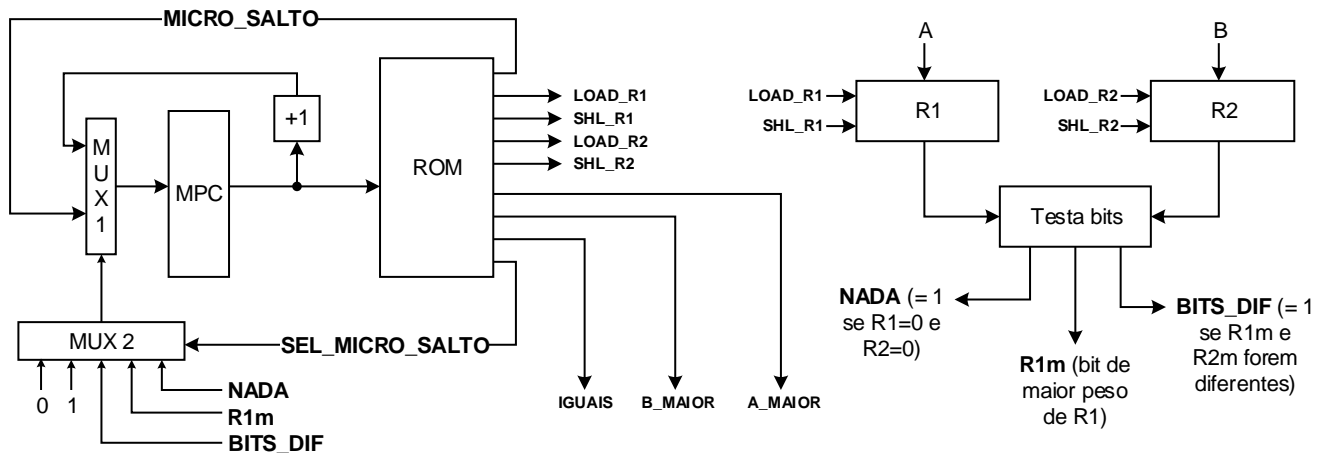
5. (2 valores) Considere a seguinte tabela de verdade, relativa a uma função de quatro entradas e uma saída. Simplifique a respetiva função, preenchendo a tabela de Karnaugh e escrevendo a expressão algébrica mais simplificada que lhe é equivalente.

A	B	C	D	Z
0	0	0	0	
0	0	0	1	
0	0	1	0	
0	0	1	1	
0	1	0	0	
0	1	0	1	
0	1	1	0	
0	1	1	1	
1	0	0	0	
1	0	0	1	
1	0	1	0	
1	0	1	1	
1	1	0	0	
1	1	0	1	
1	1	1	0	
1	1	1	1	

		CD			
		00	01	11	10
AB	00	1	1		1
	01	1	1	1	1
	11		1		
	10	1	1		1

Z =

6. (2+1 valores) Considere o circuito seguinte, que implementa um comparador iterativo de dois números binários sem sinal. Depois de carregar os valores A e B nos registros R1 e R2, o algoritmo vai comparando os bits de maior peso (R1m e R2m). Se forem iguais, desloca os registos de um bit para a esquerda (entrando 0 do lado direito) e testa os novos bits de maior peso. Termina quando esses bits forem diferentes (BITS_DIF = 1), caso em que o bit de maior peso de R1 indica qual é o maior, ou os dois registos forem zero (NADA = 1). Nessa altura, um dos sinais de saída (A_MAIOR, B_MAIOR ou IGUAIS) deve ser colocado a 1 e o programa fica em salto infinito nessa microinstrução.



- a) Preencha a tabela seguinte com os sinais necessários para implementar o comparador. Indique apenas os sinais relevantes em cada ciclo de relógio e deixe em branco as restantes células.

Endereço	Microinstrução (RTL)	LOAD_R1	LOAD_R2	SHL_R1	SHL_R2	IGUAIS	A_MAIOR	B_MAIOR	SEL_MICRO_SALTO	MICRO_SALTO
0	$R1 \leftarrow A;$ $R2 \leftarrow B;$									
1	(NADA = 1) : $MPC \leftarrow 5$									
2	($R1m \neq R2m$) : $MPC \leftarrow 6$									
3	$R1 \leftarrow R1 \ll 1;$ $R2 \leftarrow R2 \ll 1$									
4	$MPC \leftarrow 1$									
5	$IGUAIS \leftarrow 1;$ $MPC \leftarrow 5$									
6	($R1m = 1$) : $MPC \leftarrow 8$									
7	$B_MAIOR \leftarrow 1;$ $MPC \leftarrow 7$									
8	$A_MAIOR \leftarrow 1;$ $MPC \leftarrow 8$									

- b) Quantos bits de largura deve ter no mínimo a ROM de microprograma?

7. (1,5 + 1,5 valores) Considere uma cache de dados de mapeamento direto, com capacidade para 256 blocos de 8 palavras cada, para um processador com 16 bits de endereço com endereçamento de byte.

a) Quantos bits deve ter a etiqueta?

b) Suponha que o tempo de acesso em caso de *hit* e de *miss* é de 5 ns e 40 ns, respectivamente. Se a *hit rate* média for de 90%, qual o tempo médio de acesso?

 ns

8. (2 valores) Imagine um processador com endereçamento de byte, capaz de endereçar um espaço virtual de 00000H até FFFFFH, enquanto o espaço de endereçamento físico vai de 0000H até FFFFH, mas só há RAM entre 4000H e 8000H. As páginas físicas têm uma dimensão de 100H bytes. A TLB é totalmente associativa de 8 entradas e está inicialmente vazia, após o que o processador acedeu aos seguintes endereços virtuais, por esta ordem:

- 348E0H
- AC78AH
- AC18CH
- 56BE4H
- 3488CH
- AC1E4H
- AC1E0H
- AC78CH
- 56AF4H
- 5B658H
- 56A4AH

Preencha a TLB com os valores com que vai ficando, após cada um destes acessos. Assuma que as páginas físicas vão sendo atribuídas por ordem, começando na primeira (a partir do endereço 4000H) e indo até onde for necessário. Coloque X nos campos da TLB cujo valor não seja determinável neste exemplo.

Posição da TLB	Bit validade	N.º página virtual (hexadecimal)	N.º página física (hexadecimal)
0			
1			
2			
3			
4			
5			
6			
7			