

NOME		NÚMERO	
------	--	--------	--

1. (1 + 2 + 2 valores) Considere o seguinte programa do PEPE-16.

```

PLACE      2000H
var:        WORD      5
pilha:      TABLE    100H
fim_pilha:

```

rots_int:	WORD	int_0
	WORD	0
	WORD	0
	WORD	int_3

```

PLACE      0
MOV        R1, var

```

MOV	SP, fim_pilha
MOV	BTE, rots_int
EI0	
EI3	
EI	

```

fim:        JMP        fim

```

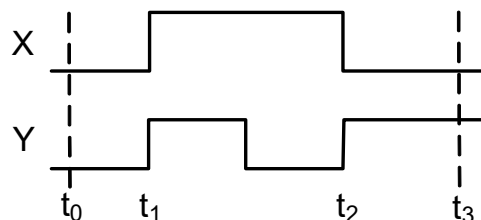
```

int_0:      PUSH      R2
            MOV        R2, 3
            CALL       rotina
            POP        R2
            RFE
int_3:      PUSH      R2
            MOV        R2, 1
            CALL       rotina
            POP        R2
            RFE
rotina:     PUSH      R3
            MOV        R3, [R1]
            DI
            ADD        R3, R2
            MOV        [R1], R3
            POP        R3
            RET

```

- a) Complete os espaços em branco com o necessário para as interrupções 0 e 3 funcionarem corretamente;
- b) Suponha que os sinais X e Y ligam aos pinos de interrupção 0 e 3 do PEPE, respetivamente, e têm a evolução temporal indicada na figura seguinte. Os flancos dos sinais X e Y nos instantes t_1 e t_2 ocorrem simultaneamente.

Sabendo que o programa começa a executar em t_0 , que entre t_1 e t_2 decorrem cerca de 4 segundos e que a evolução temporal está aproximadamente à escala, preencha a tabela seguinte com informação sobre os acessos à variável “var” entre t_0 e t_3 (tempo aproximado em que ocorre, se é leitura ou escrita, qual o valor lido ou escrito e qual a interrupção que causou o acesso). Respeite a ordem temporal dos acessos.



- c) Responda novamente à mesma questão da alínea b), mas agora substituindo a instrução DI da “rotina” pela instrução EI (que faz o oposto da instrução DI).

Alínea b) (DI)			
Tempo	Leitura/escrita	Valor	Interrup.
t1	Leitura	5	0
t1	Escrita	8	0
t1	Leitura	8	3
t1	Escrita	9	3
t2	Leitura	9	3
t2	Escrita	AH	3

Alínea c) (EI)			
Tempo	Leitura/escrita	Valor	Interrup.
t1	Leitura	5	0
t1	Leitura	5	3
t1	Escrita	6	3
t1	Escrita	8	0
t2	Leitura	8	3
t2	Escrita	9	3

2. (2 valores) Como vai ter uma boa nota neste teste, merece um portátil novo, para correr aquele jogo novo em que cada *frame* gasta 30% do seu tempo em CPU e 70% em GPU (processador gráfico). Está indeciso entre um (A) que consegue 100 *frames/seg* neste jogo e outro (B) cujo CPU é cerca de 50% (1,5 vezes) mais rápido e cujo GPU é cerca de 30% (1,3 vezes) mais lento. Assumindo que todos os restantes fatores são iguais nos dois portáteis, qual portátil escolheria, A ou B? Justifique, calculando quantos *frames/seg* o portátil B conseguirá neste jogo (contas aproximadas!).

No portátil A, cada frame demora um tempo T_A :

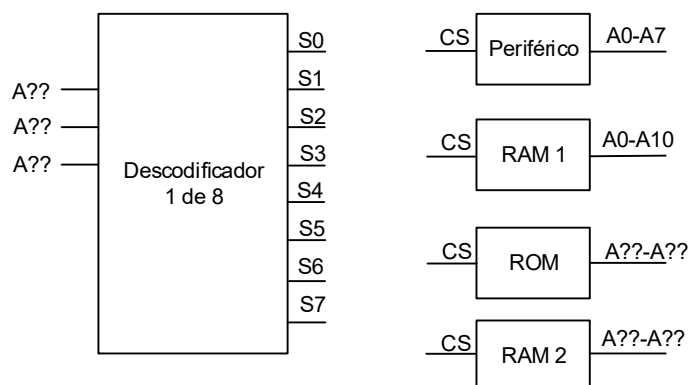
$$T_A = 0,3 * T_A + 0,7 * T_A \quad (\text{tempo CPU} + \text{tempo GPU})$$

No portátil B, cada frame demorará, para o mesmo jogo:

$$T_B = (0,3/1,5) * T_A + (0,7*1,3) * T_A = (0,2+0,91) * T_A = 1,11 * T_A$$

O portátil B é assim mais lento que o portátil A, conseguindo só cerca de $100/1,11 \approx 90$ *frames/seg*. Irei escolher o portátil A.

3. (2 valores) Considere o seguinte sistema de descodificação de endereços utilizado por um processador de bus de dados de 8 bits e bus de endereços de 16 bits. Preencha a informação em falta sobre o descodificador e cada dispositivo (bits de endereço a que liga, capacidade, saída do descodificador a que deve ligar e o endereço de fim da gama de endereços em que esse dispositivo está ativo, não considerando endereços de acesso repetido - espelhos).

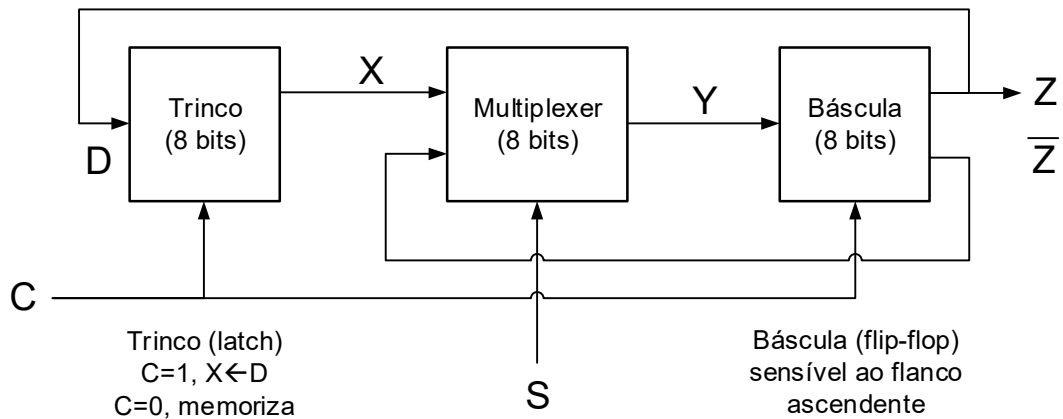


Bits de endereço a que o descodificador deve ligar

A13 .. A15

Dispositivo	Bits de endereço	Capacidade (bytes) (decimal)	Saída do descodificador	Início (hexadecimal)	Fim (hexadecimal)
Periférico	A0-A7	256	S4	8000H	80FFH
RAM 1	A0-A10	2 Ki	S0	0000H	07FFH
ROM	A0-A11	4 Ki	S3	6000H	6FFFH
RAM 2	A0-A8	512	S6	C000H	C1FFH

4. (2 valores) Considere o seguinte circuito, com 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 C e S evoluem ao longo do tempo da forma indicada na tabela seguinte, preencha os valores estáveis no 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).



C	1	0	1	0	0	1	1	0	1	1
S	1	1	1	1	0	0	1	1	1	0
X	60	60	9F	9F	9F	9F	9F	9F	60	60
Y	9F	9F	60	60	9F	9F	60	60	9F	60
Z	60	60	9F	9F	9F	9F	9F	9F	60	60
\bar{Z}	9F	9F	60	60	60	60	60	60	9F	9F

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 verdade a partir da tabela de Karnaugh e escrevendo a expressão algébrica mais simplificada que lhe é equivalente.

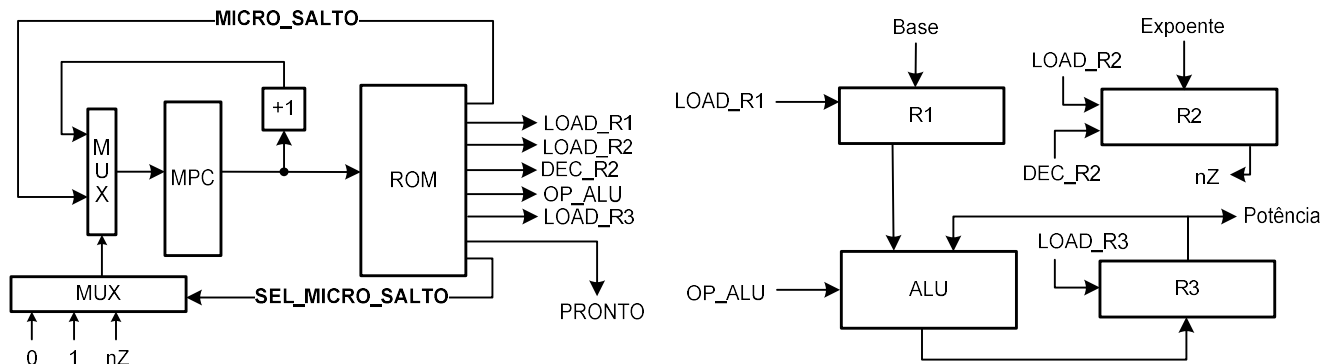
A	B	C	D	Z
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	0
1	0	1	0	1
1	0	1	1	0
1	1	0	0	0
1	1	0	1	1
1	1	1	0	0
1	1	1	1	1

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

Z =

$$\bar{A}\bar{C} + \bar{A}B + BD + \bar{B}\bar{D}$$

6. (1,5 + 0,5 valores) Pretende-se construir um circuito microprogramado que implemente a operação potência (base elevada a expoente) por multiplicações sucessivas da base. O diagrama seguinte descreve o circuito. Os registos R1 e R2 recebem a base e o expoente, respetivamente. O registo R3 vai acumulando o resultado das multiplicações. A saída da ALU pode tomar uma de quatro possibilidades, consoante o valor de OP_ALU: soma (SOMA) e multiplicação (MUL) dos dois operandos, 0 (ZERO) e 1 (UM). O sinal nZ está ativo (vale 1) quando R2 é diferente de zero e o sinal PRONTO é ativado quando o resultado está pronto.



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

Endereço na ROM	Microinstruções	PRONTO	LOAD_R1	LOAD_R2	DEC_R2	OP_ALU	LOAD_R3	SEL_MICRO_SALTO	MICRO_SALTO
0	R1 ← Base R2 ← Expoente		SIM	SIM					
1	R3 ← 1					UM	SIM		
2	R3 ← R3 * R1					MUL	SIM		
3	R2 ← R2 - 1				SIM				
4	(R2 != 0): MPC ← 2							nZ	2
5	PRONTO ← 1 MPC ← 5	SIM						1	5

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

12

7. (1,5 + 1,5 valores) Suponha que a *cache* do PEPE (processador com 16 bits de endereço, endereçamento de byte) é de mapeamento direto, com uma capacidade de 512 palavras (blocos de 4 palavras).

- a) Indique o número de bits de cada um dos campos em que o endereço se divide para acesso à *cache*.

Etiqueta	6
Índice	7
Palavra dentro do bloco	2
Byte dentro da palavra	1

- b) Na execução de instruções do tipo MOV R1, [R2], o núcleo do PEPE verificou que nuns casos o valor de R1 demorava 3 ns a obter, noutros 20 ns, e que em média demorava 6,4 ns. Qual a taxa de sucesso (*hit rate*) da *cache* com este programa?

80 %

8. (2 valores) Considere um processador com 32 bits de endereço, endereçamento de byte e suporte para memória virtual com páginas de 4Ki bytes. Assuma que a memória física tem uma capacidade de 4 Mi bytes e que a TLB é uma *cache* totalmente associativa de 4 entradas, cujo conteúdo é numa dada altura o indicado na tabela da esquerda. Acabe de preencher as outras duas tabelas para este exemplo concreto.

Válida	Página alterada	Nº de página virtual	Nº de página física
1	1	23BAH	68H
0	0	7BAH	3BH
1	1	158FCH	3AH
1	0	2B5DH	3BH

Dimensão do espaço virtual	4 Gi bytes
Número de páginas virtuais	1 Mi
Número de páginas físicas	1 Ki

Endereço virtual	Endereço físico
158FCB6CH	3AB6CH
23BAE4AH	68E4AH
2B5D813H	3B813H
23BAA18H	68A18H