

NOME	NÚMERO
------	--------

1. (1 + 2 + 1 valores) Considere o seguinte programa, que usa rotinas de interrupção. Os pinos de interrupção 1 e 3 ligam a relógios com um determinado período, pedindo uma interrupção em cada flanco ascendente.

PLACE	1000H	
pilha:	TABLE	100H
fim_pilha:		
contador:	WORD	0
tab:	WORD	rot_int_1
	WORD	rot_int_3
PLACE	0000H	
	MOV	SP, fim_pilha
	EI	
	MOV	R1, contador
ciclo:	MOV	R2, [R1]
	ADD	R2, 1
	MOV	[R1], R2
	JMP	ciclo
rot_int_1:	...	; processa int 1
	RET	
rot_int_3:	...	; processa int 3
	RET	

[illegible]

a) Verificou-se que este programa tem erros que impedem o funcionamento correto das interrupções 1 e 3. Reescreva o programa na tabela do lado direito, corrigindo esses erros (o conteúdo das rotinas de interrupção não é relevante). Preencha apenas as linhas que entender serem necessárias;

b) Agora com o programa a funcionar corretamente, suponha que com os relógios desligados (sem gerarem interrupções) a variável “contador” consegue ser incrementada 1000 vezes em cada segundo. As rotinas de interrupção 1 e 3 demoram 10 e 20 milissegundos a executar, respetivamente. Se ligar apenas o relógio que liga à interrupção 1, com um período de 100 milissegundos, quantas vezes é que a variável “contador” será incrementada em cada segundo, em média?

vezes

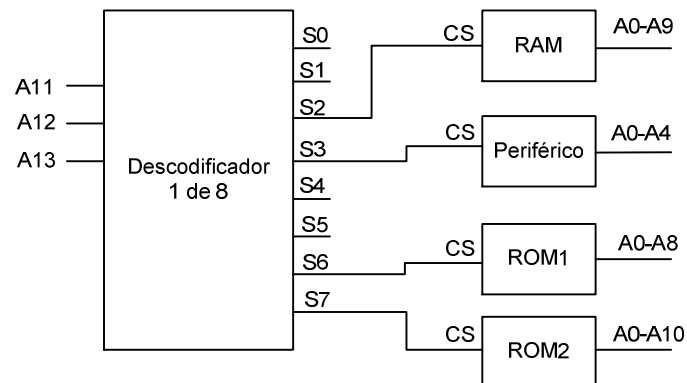
c) Suponha agora que, mantendo o relógio da interrupção 1 a funcionar (100 milissegundos de período), liga também o relógio da interrupção 3. Os tempos de execução das rotinas de interrupção 1 e 3 são 10 e 20 milissegundos, respetivamente. Qual é aproximadamente o período mínimo do relógio da interrupção 3 para que a variável “contador” seja incrementada pelo menos 500 vezes em cada segundo? Sugestão: pense em termos de percentagem de tempo gasto em cada atividade (ciclo do programa principal e rotinas de interrupção).

milissegundos

2. (2 + 1 valores) Os portáteis mais recentes incluem não apenas um disco mas também um SSD (Solid State Drive). Suponha que vai comprar um com Windows 10 Home, mas claro que vai instalar o Windows 10 Education. O disco tem um 1 TByte de capacidade, 7200 rotações/minuto, 6 ms de *seek-time* (procura de pista) e taxa de leitura de dados de 100 MBytes/seg. O SSD tem uma capacidade de 128 GBytes, um tempo de acesso de 0,2 ms e taxa de leitura de dados de 200 MBytes/seg. Considera-se que o setor é a unidade de acesso nos dois casos e o seu tamanho é 1 KByte.
- a) Imaginemos que o arranque do Windows requer a leitura de 4000 setores, distribuídos pelo dispositivo (disco ou SSD) de forma aleatória, a que se somam cerca de 15 segundos de execução de programas em memória. Estime o tempo que demorará aproximadamente o arranque do Windows, no caso do disco e do SSD;

- b) Em qual dispositivo (disco ou SSD) deverá instalar o Windows, e qual dispositivo deverá usar para guardar os seus ficheiros? Justifique.

3. (3 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 tabela sobre cada dispositivo, indicando 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)
RAM	A0-A9			
Periférico	A0-A4			
ROM1	A0-A8			
ROM2	A0-A10			

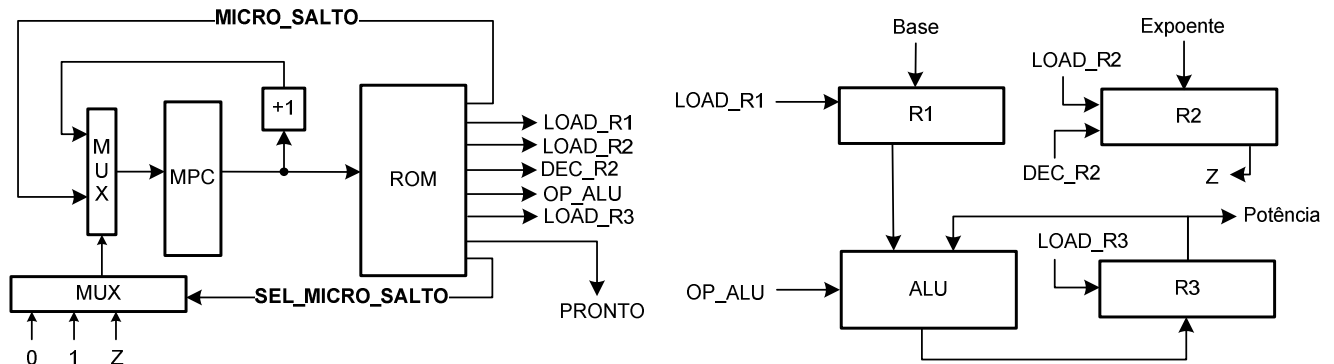
4. (2 valores) Considere a seguinte tabela de verdade, relativa a uma função de quatro entradas e uma saída. Simplifique a respetiva função, preenchendo a tabela de Karnaugh e escrevendo a expressão algébrica mais simplificada que lhe é equivalente.

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

		CD			
		00	01	11	10
AB	00				
	01				
	11				
	10				

Z =

5. (2 + 1 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 Z está ativo (vale 1) quando R2 é 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	LOAD_R1	LOAD_R2	DEC_R2	OP_ALU	LOAD_R3	PRONTO	SEL_MICRO_SALTO	MICRO_SALTO
0	R1 ← Base R2 ← Expoente R3 ← 1								
1	R3 ← R3 * R1								
2	R2 ← R2 - 1								
3	(R2 = 0): MPC ← 5								
4	MPC ← 1								
5	PRONTO ← 1 MPC ← 5								

- b) Quantos bits de largura deve ter no mínimo o sinal MICRO_SALTO?

6. (1,5 + 1,5 valores) Considere uma cache de dados de mapeamento direto, com capacidade para 128 blocos de 8 palavras cada, para um processador com 16 bits de endereço com endereçamento de byte (uma palavra = dois endereços).

- a) Quantos bits deve ter a etiqueta?

- b) Suponha que o tempo de acesso em caso de *hit* e de *miss* é de 4 ns e 30 ns, respetivamente. Se a *hit rate* média for de 80%, qual o tempo médio de acesso?

 ns

7. (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 1000H e 4000H. As páginas virtuais 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:

- 2AFE0H
- AC18AH
- AC78CH
- 348E4H
- 2AF8CH
- AC7E4H
- AC7E0H
- AC18CH
- 348F4H
- 56B4AH
- 2AF58H

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 1000H) 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			



NOME		NÚMERO	
------	--	--------	--

1. (1 + 2 + 1 valores) Considere o seguinte programa, que usa rotinas de interrupção para alterar o valor de um display hexadecimal.

HexaDisplay EQU 8000H

PLACE 1000H

pilha: TABLE 100H

fim_pilha:

PLACE 0

MOV R1, 1 ; contador


```
rot0:  PUSH    R2
        MOV     R2, HexaDisplay
        EI
        SUB     R1, 1 ; decrementa
        MOVB    [R2], R1 ; atualiza display
        POP     R2
        RFE
```

```
rot2:  PUSH    R2
        MOV     R2, HexaDisplay
        ADD     R1, 2 ; incrementa
        MOVB    [R2], R1 ; atualiza display
        POP     R2
        RFE
```

- 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 (rotinas no lado direito). Preencha apenas os espaços que entender serem necessários. O corpo do programa principal deve terminar em ciclo infinito (salto para a própria instrução);

- b) Suponha que os pedidos de interrupção 0 e 2 se alternam, com um segundo de intervalo, e que a interrupção 0 é a primeira a ser pedida. Indique de seguida a sequência dos 8 primeiros valores que aparecem no display;

<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
----------------------	----------------------	----------------------	----------------------	----------------------	----------------------	----------------------	----------------------

- c) Suponha agora que os pinos do PEPE da interrupção 0 e 2 ligam ao mesmo sinal externo (logo, as interrupções são pedidas exatamente ao mesmo tempo), que origina uma interrupção em cada segundo. Indique de seguida a sequência dos 8 primeiros valores que aparecem no display (recomeçando o programa). Justifique as diferenças face ao caso anterior.

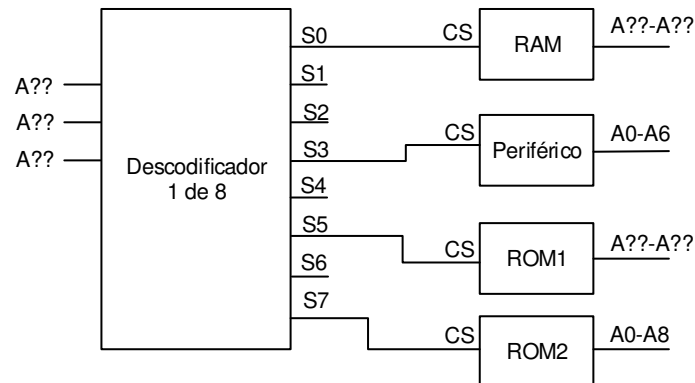
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
----------------------	----------------------	----------------------	----------------------	----------------------	----------------------	----------------------	----------------------

<div></div>

2. (2 + 1 valores) Agora que estamos quase no Natal decidiu renovar o seu portátil, substituindo o disco por um SSD (Solid State Drive). O seu disco tem 5400 rotações/minuto, 6 ms de *seek-time* (procura de pista) e taxa de leitura de dados de 100 MBytes/seg. O SSD tem um tempo de acesso de 0,1 ms e taxa de leitura de dados de 200 MBytes/seg. Considera-se que o setor é a unidade de leitura nos dois casos.
- a) Para ter uma ideia de quanto o seu PC vai melhorar, fez um pequeno *benchmark* que lê 1000 setores de 1 KByte cada um, distribuídos pelo dispositivo de forma aleatória, quer no disco, quer no SSD. Estime o tempo que demorará aproximadamente a execução do seu *benchmark* no caso do disco e do SSD e qual a ordem de grandeza de quantas vezes (N) o SSD será mais rápido.

- b) Isto quer dizer que o seu PC vai correr as aplicações N vezes mais rápido, em que N é o valor calculado na alínea anterior? Justifique;

3. (3 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				
RAM		4 K		
Periférico	A0-A6		3000H	
ROM1		1 K		
ROM2	A0-A8			

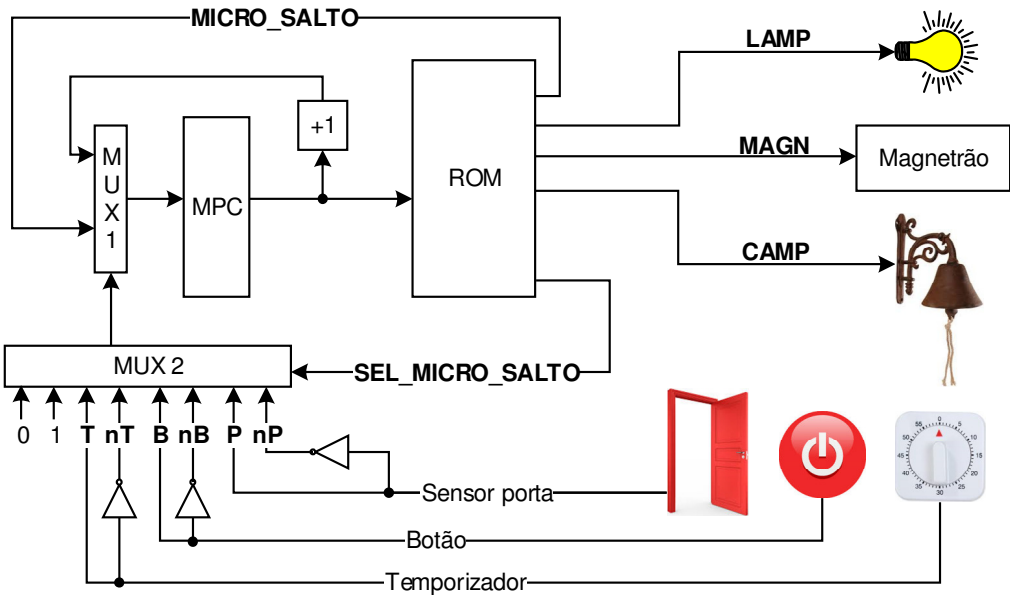
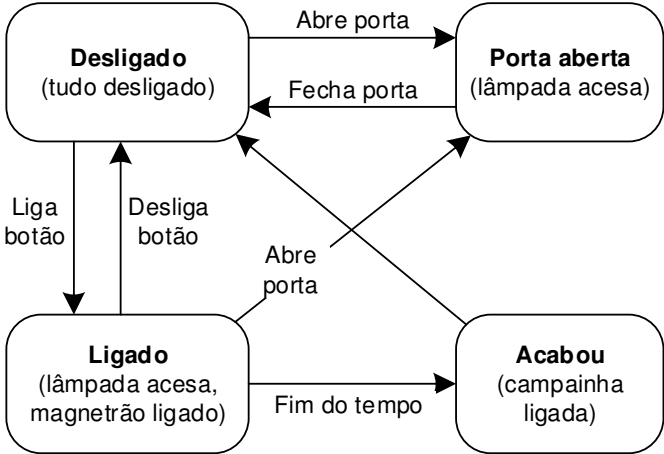
4. (2 valores) Considere a seguinte tabela de verdade, relativa a uma função de quatro entradas e uma saída. Simplifique a respetiva função, preenchendo a tabela de Karnaugh e escrevendo a expressão algébrica mais simplificada que lhe é equivalente.

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

		CD			
		00	01	11	10
AB	00				
	01				
	11				
	10				

Z =

5. (2 + 1 valores) Pretende-se implementar um controlador microprogramado para um micro-ondas. O diagrama de estados seguinte indica os 4 estados possíveis, as saídas relevantes em cada estado e os eventos que podem fazer o controlador transitar de estado. Há três sensores (**P** - porta aberta, **B** - botão ligado e **T** - tempo acabou) e três dispositivos a controlar (lâmpada, magnetrão e campainha de fim de tempo). O temporizador de funcionamento é mecânico (só o sensor indica o fim). As negações nos sinais dos sensores permitem testar as condições negadas.



a) Preencha a tabela seguinte com os valores necessários para implementar o controlador. Cada estado é implementado por um ciclo que testa cada um dos sensores relevantes, podendo mudar de estado (saltar para a primeira microinstrução doutro estado). Indique apenas os sinais relevantes em cada ciclo de relógio e deixe em branco as restantes células.

Endereço na ROM	Estado	Microinstruções	LAMP	MAGN	CAMP	SEL_MICRO_SALTO	MICRO_SALTO
0	Desligado	Muda para Ligado se botão ligar					
1		Mantém estado se porta fechada					
2	Ligado	Muda para Porta Aberta se porta abrir					
3		Muda para Acabou se fim de tempo					
4		Caso contrário, mantém estado					
5	Porta Aberta	Muda para Desligado se porta fechar					
6		Caso contrário, mantém estado					
7	Acabou	Muda para Desligado					

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

6. (1 + 2 valores) Suponha que a *cache* do PEPE (processador com 16 bits de endereço, endereçamento de byte) é de mapeamento direto, usa blocos de 8 palavras e tem 7 bits de etiqueta.

a) Quantos blocos pode a cache armazenar, se estiver completamente cheia?

- b) Suponha que o processador acedeu ao endereço 1234H, ficando o respetivo valor em cache, e que logo a seguir acedeu a outro endereço, do qual conhece apenas os dígitos de maior e menor peso (8 e A). Indique possíveis dígitos intermédios do endereço, em duas situações:

- O segundo valor acedido retira o primeiro da cache

8			A	H
---	--	--	---	---

- O segundo valor acedido não retira o primeiro da cache

8			A	H
---	--	--	---	---

7. (2 valores) Imagine um processador com endereçamento de byte, capaz de endereçar um espaço virtual de 000 000H até FFF FFFH, enquanto o espaço de endereçamento físico vai de 0000H até FFFFH, mas só há RAM de 0000H até 8000H. As páginas virtuais têm uma dimensão de 100H bytes. A TLB é totalmente associativa de 8 entradas e tem atualmente o conteúdo da tabela seguinte (algumas posições estão vazias, isto é, não inicializadas).

Posição da TLB	Bit validade	N.º página virtual (hexadecimal)	N.º página física (hexadecimal)
0	0	3B9	1F
1	1	207	31
2	1	2A0	3E
3	1	1EF	0F
4	1	4B8	1F
5	0	0C3	1D
6	1	C31	1B
7	0	A25	0C

Preencha a tabela seguinte para este computador e para este conteúdo da TLB.

Número de bits do espaço virtual	
Número de páginas virtuais	
Número de páginas físicas	
Endereço virtual que corresponde ao endereço físico 1F3AH	



NOME		NÚMERO	
------	--	--------	--

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

ECRA	EQU	8000H
PLACE	1000H	
pilha:	TABLE	100H
PLACE	0	
	MOV SP,	
	MOV BTE,	
	MOV R1, ECRA	

rotInt1:		
rotInt2:		

- a) Do lado esquerdo, complete a zona de dados e o programa principal com o necessário para a pilha e as interrupções 1 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 (pixel screen, periférico de 8 bits). A interrupção 1 coloca 0FH nesse byte (). A interrupção 2 coloca F0H no mesmo byte (). O efeito visual no ecrã é os pixels a 1 (a preto) oscilarem entre uma posição e outra, ao ritmo das interrupções. O endereço do ecrã mantém-se no R1 durante todo o programa;
- c) Suponha agora que os pinos do PEPE da interrupção 1 e 2 ligam ao mesmo sinal externo (logo, as interrupções são pedidas exatamente ao mesmo tempo). Qual é o efeito visual no ecrã (o que é que o utilizador vê, concretamente)? Justifique.

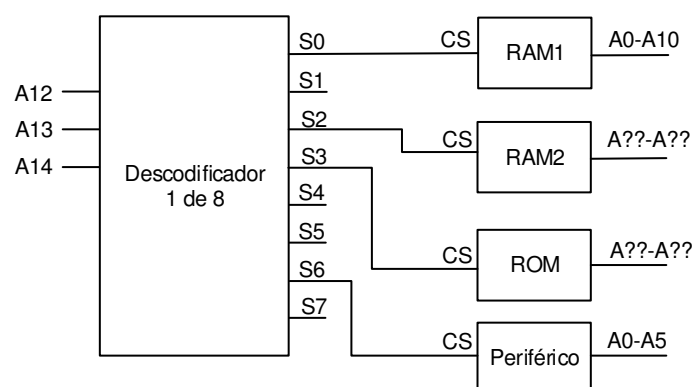
--

2. (1 valor) Uma transmissão de dados por um barramento série assíncrono, com bit de paridade e 2 stop bits, a um ritmo de transmissão de 1000 bits/seg, demorou 4 minutos a completar-se. Indique o número máximo de bytes de dados que puderam ser transmitidos. Justifique.

--

3. (2 valores) Agora que está no IST (parabéns!), decidiu comprar um portátil novo. O modelo mais interessante tem duas variantes, uma com disco de 5400 rotações/minuto, 6 ms de *seek-time* (procura de pista) e taxa de transmissão de dados de 50 MBytes/seg, e a outra com 7200 rotações/minuto, 8 ms e 100 MBytes/seg. De resto, os portáteis são iguais. Para decidir qual escolher, fez um pequeno *benchmark* que lê 1000 setores de 1 KByte cada um (igual nos dois discos), distribuídos de forma aleatória pelo disco. Quanto tempo demorará aproximadamente cada portátil a executar o seu *benchmark*? Qual será então a sua escolha? Justifique.

4. (3 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 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)
RAM1	A0-A10			
RAM2		4 K		
ROM		1 K		
Periférico	A0-A5			

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 simplificada.

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

		CD			
		00	01	11	10
AB	00				
	01				
	11				
	10				

Z =

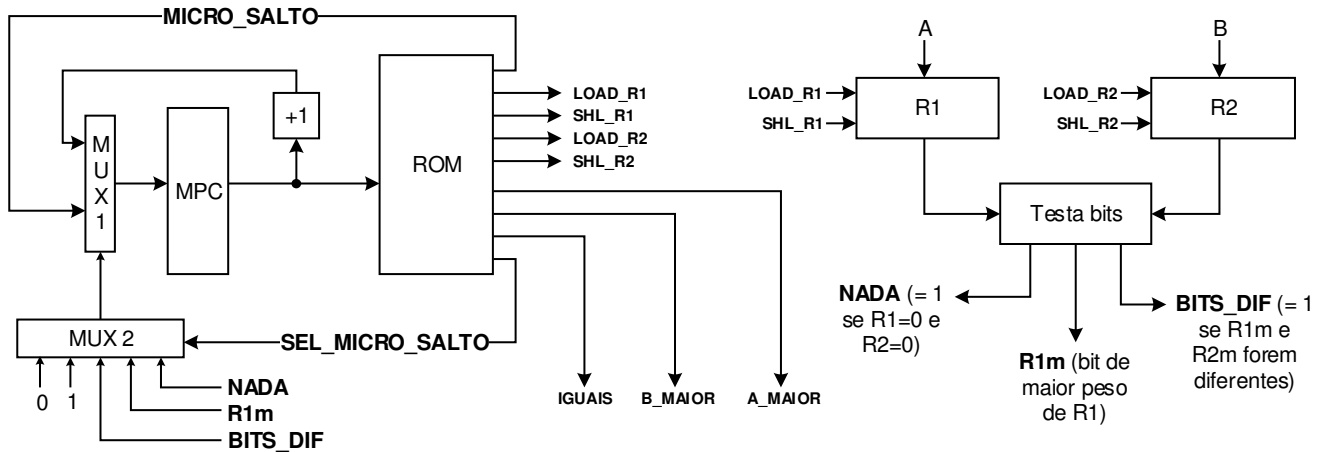
6. (1+1 valores) Suponha que a *cache* do PEPE (processador com 16 bits de endereço, endereçamento de byte) é de mapeamento direto, com uma capacidade de 256 palavras (blocos de 8 palavras).

a) Quantos bits deve ter a etiqueta?

b) Suponha que o tempo que o processador demora a obter uma palavra, em leitura, é de 3 ns e 30 ns em caso de *cache hit* e de *miss*, respetivamente, e que o processador está em ciclo a aceder alternadamente aos endereços 1000H e 2000H (e a mais nenhum outro). Qual o tempo médio de acesso nesta situação?

 ns

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

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

8. (3 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 de 0000H até 4000H. As páginas físicas têm uma dimensão de 100H bytes. A TLB é totalmente associativa de 8 entradas e tem atualmente o conteúdo da tabela seguinte (algumas posições estão vazias, isto é, não inicializadas).

Posição da TLB	Bit validade	N.º página virtual (hexadecimal)	N.º página física (hexadecimal)
0	0	3B9	0C
1	0	207	31
2	1	2A0	3E
3	0	1EF	0F
4	1	4B8	25
5	0	0C3	1D
6	1	C31	1B
7	1	A25	0C

Preencha a tabela seguinte para este computador e para **este conteúdo** da TLB.

N.º bits do espaço virtual	
N.º bits do espaço físico	
N.º páginas virtuais	
N.º <u>máximo</u> de páginas que podem estar carregadas simultaneamente na RAM (considerando que toda a RAM está disponível para conter páginas)	
N.º de páginas virtuais cuja tradução virtual/físico não precisa de aceder à tabela de páginas	
Endereço virtual que corresponde ao endereço físico 0C31H	



NOME		NÚMERO	
------	--	--------	--

1. (1+1+1 valores) Considere o seguinte programa, que deve completar. Preencha apenas o que entender necessário.

PLACE	2000H	
registo:	WORD	0FFFFH
pilha:		100H
PLACE	0	
	MOV SP,	
	MOV BTE,	
int2:	PUSH	

- a) Complete a zona de dados, com o necessário para a pilha e a interrupção 2 funcionarem corretamente;
- b) Complete o programa principal, com o necessário para a pilha e a interrupção 2 funcionarem corretamente. O corpo do programa principal deve terminar em ciclo infinito (salto para a própria instrução);
- c) Complete a rotina de interrupção 2 ("int2"), que deve deslocar a variável "registo" de 1 bit para a direita.

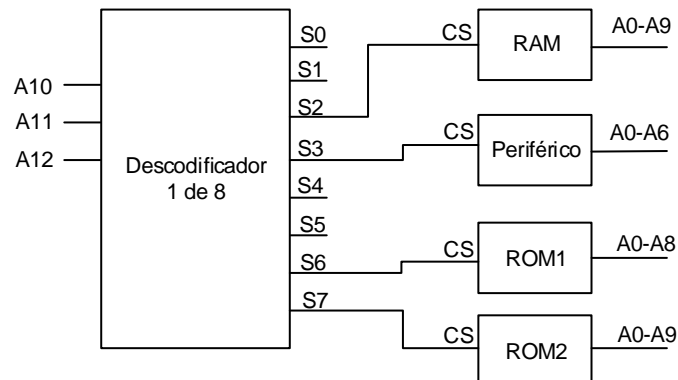
2. (1 valor) Numa transmissão de dados por um barramento série assíncrono, com bit de paridade e 2 stop bits, a um ritmo de transmissão de 1000 bits/seg, indique o tempo mínimo para transmitir 30.000 bytes de dados.

--

3. (2 valores) Tem de decidir entre dois processadores qual o melhor para correr a sua aplicação. O processador A tem um relógio de 2 GHz, a duração média de cada instrução é de 2 ciclos de relógio por instrução, e o compilador gera 200 Mega instruções. Por sua vez o processador B tem um relógio de 3 GHz, a duração média de cada instrução é de 3 ciclos de relógio por instrução, e o compilador gera 220 Mega instruções. Indique qual o melhor processador. Justifique.

--

4. (3 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 o tamanho de cada dispositivo (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	Tamanho (bytes)	Início	Fim
RAM			
Periférico			
ROM1			
ROM2			

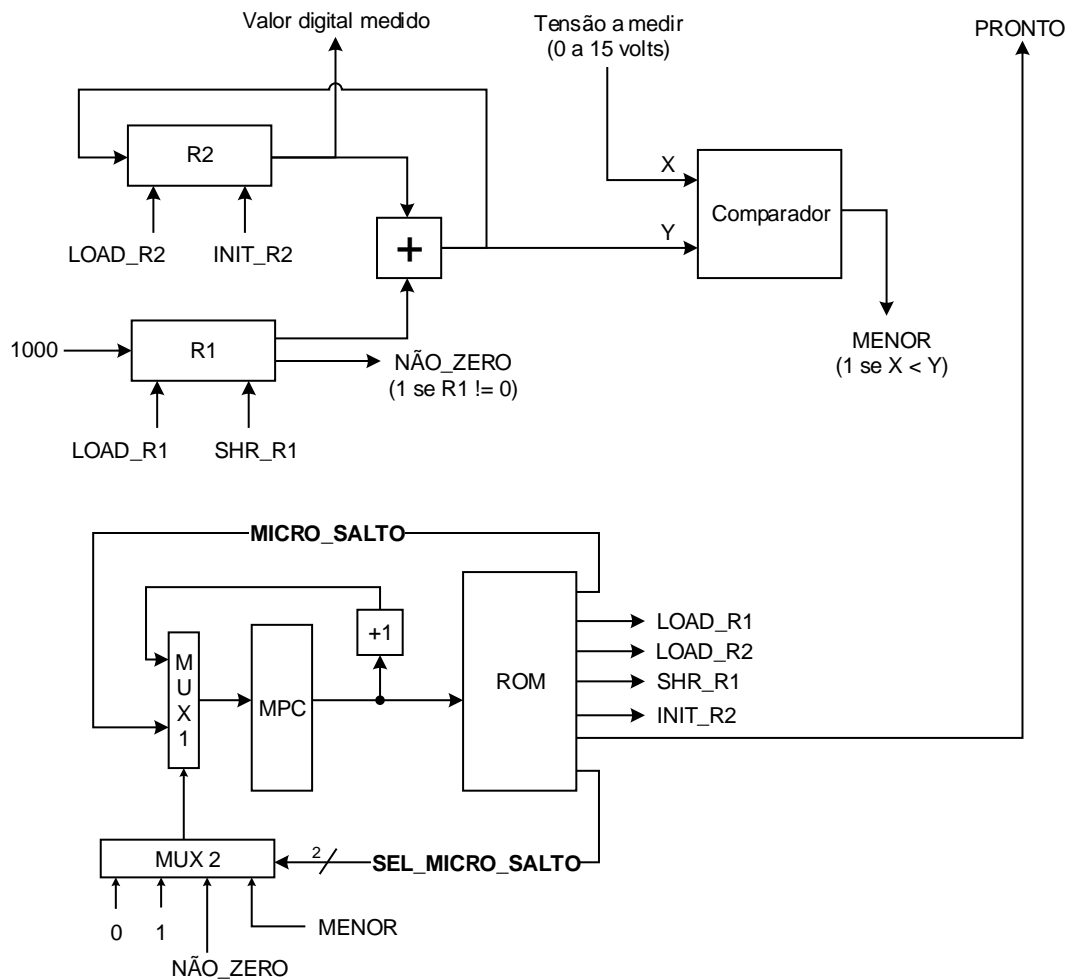
5. (3 valores) Considere a seguinte tabela de verdade, relativa a uma função de quatro entradas e uma saída. Simplifique a respetiva função, preenchendo a tabela de Karnaugh e escrevendo a expressão algébrica simplificada.

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

		CD			
		00	01	11	10
AB	00				
	01				
	11				
	10				

Z =

6. (2+1 valores) 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.



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

Endereço	Microinstrução (RTL)	LOAD_R1	LOAD_R2	SHR_R1	INIT_R2	PRONTO	SEL_MICRO_SALTO	MICRO_SALTO
0	$R1 \leftarrow 1000b;$ $R2 \leftarrow 0;$							
1	$(X < Y) : MPC \leftarrow 3$							
2	$R2 \leftarrow R2 + R1;$							
3	$R1 \leftarrow R1 \gg 1$							
4	$(R1 \neq 0) : MPC \leftarrow 1$							
5	$PRONTO \leftarrow 1;$ $MPC \leftarrow 5$							

- b) Quantos bits de largura deve ter no mínimo o sinal MICRO_SALTO?

7. (1+1 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	
Índice	
Palavra dentro do bloco	
Byte dentro da palavra	

- 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 insucesso (miss rate) da *cache* com este programa?

%

8. (3 valores) Considere um processador com 32 bits de endereço, endereçamento de byte e suporte para memória virtual com páginas de 4K bytes. Assuma que a memória física tem uma capacidade de 4 Mbytes 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	28H

Dimensão do espaço virtual	
Número de páginas virtuais	
Número de páginas físicas	

Endereço virtual	Endereço físico
	3AB6CH
23BAE4AH	
	28813 H
23BAA18H	

NOME		NÚMERO	
------	--	--------	--

1. (1,5 + 1,5 valores) Considere o seguinte programa, que usa rotinas de interrupção para alterar o valor de um display hexadecimal.

```

PLACE      1000H
contador:   WORD      6
pilha:      TABLE    100H
fim_pilha:

```

```
PLACE      0
```

```
MOV      R2, contador
```

```
fim:      JMP      fim
```

rot1:

```
MOV      R1, [R2]
```

```
SUB      R1, 2      ; decrementa
```

```
MOV      [R2], R1 ; atualiza contador
```

rot3:

```
MOV      R1, [R2]
```

```
ADD      R1, 1      ; incrementa
```

```
MOV      [R2], R1 ; atualiza contador
```

- a) Complete a zona de dados, o programa principal e as rotinas de interrupção (do lado direito) com o necessário para as interrupções 1 e 3 funcionarem corretamente;
- b) Suponha que os pinos das interrupções 1 e 3 (sensíveis ao flanco ascendente) estão ligadas ao mesmo relógio, cuja frequência é de 1 Hertz. Assumindo que o primeiro pedido de interrupção é feito no instante 0 (quando o programa arranca), indique de seguida a sequência dos 8 primeiros valores que as rotinas de interrupção escrevem no contador, bem como o número da interrupção em que cada valor é escrito e o tempo em segundos (desde que o programa arranca) em que aproximadamente essa escrita ocorre.

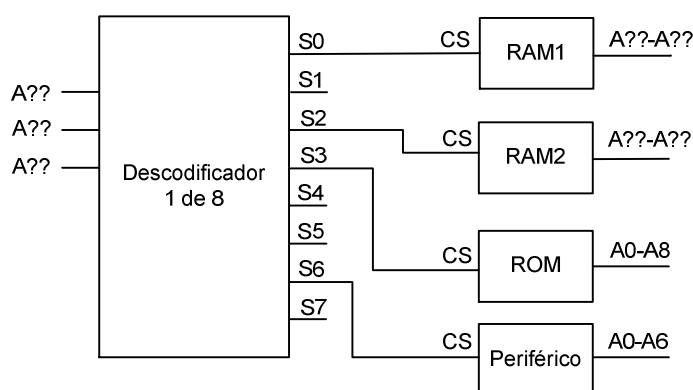
Tempo	<div style="border: 1px solid black; width: 40px; height: 20px;"></div>	<div style="border: 1px solid black; width: 40px; height: 20px;"></div>	<div style="border: 1px solid black; width: 40px; height: 20px;"></div>	<div style="border: 1px solid black; width: 40px; height: 20px;"></div>	<div style="border: 1px solid black; width: 40px; height: 20px;"></div>	<div style="border: 1px solid black; width: 40px; height: 20px;"></div>	<div style="border: 1px solid black; width: 40px; height: 20px;"></div>	<div style="border: 1px solid black; width: 40px; height: 20px;"></div>
Interrupção	<div style="border: 1px solid black; width: 40px; height: 20px;"></div>	<div style="border: 1px solid black; width: 40px; height: 20px;"></div>	<div style="border: 1px solid black; width: 40px; height: 20px;"></div>	<div style="border: 1px solid black; width: 40px; height: 20px;"></div>	<div style="border: 1px solid black; width: 40px; height: 20px;"></div>	<div style="border: 1px solid black; width: 40px; height: 20px;"></div>	<div style="border: 1px solid black; width: 40px; height: 20px;"></div>	<div style="border: 1px solid black; width: 40px; height: 20px;"></div>
Valor	<div style="border: 1px solid black; width: 40px; height: 20px;"></div>	<div style="border: 1px solid black; width: 40px; height: 20px;"></div>	<div style="border: 1px solid black; width: 40px; height: 20px;"></div>	<div style="border: 1px solid black; width: 40px; height: 20px;"></div>	<div style="border: 1px solid black; width: 40px; height: 20px;"></div>	<div style="border: 1px solid black; width: 40px; height: 20px;"></div>	<div style="border: 1px solid black; width: 40px; height: 20px;"></div>	<div style="border: 1px solid black; width: 40px; height: 20px;"></div>

2. (1,5 + 1,5 valores) Uma transmissão de dados é feita por um barramento série assíncrono, com bit de paridade, 2 stop bits e um ritmo de transmissão de 10.000 bits/seg.

- a) Indique qual o tempo mínimo necessário para enviar 10.000 bytes de dados. Justifique;

- b) Suponha agora que um programa demora T_1 segundos a executar, gastando 60% deste tempo em processamento e no fim transmite os 10.000 bytes da alínea anterior, usando para tal o tempo mínimo possível, o que corresponde a 40% do tempo de execução do programa. Se o ritmo de transmissão de dados duplicar (dos 10.000 bits/seg para 20.000 bits/seg), mantendo-se todas as restantes condições, o tempo total de execução do programa passou a ser T_2 . Qual a melhoria $M=T_1/T_2$ verificada no desempenho? Justifique.

3. (3 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. Pretende-se que cada saída do descodificador fique ativa em 1 K endereços consecutivos (dimensão da fatia de endereços de cada saída). Preencha a tabela com os bits de endereço a que o descodificador e cada dispositivo devem ligar, a capacidade (decimal) e os endereços de início e de fim (em hexadecimal) em que cada dispositivo está ativo (não considerando endereços de acesso repetido - espelhos).



Dispositivo	Bits de endereço	Capacidade (bytes) (decimal)	Início (hexadecimal)	Fim (hexadecimal)
Descodificador				
RAM1				01FFH
RAM2		1 K		
ROM	A0-A8			
Periférico	A0-A6			

4. (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 simplificada.

A	B	C	D	Z
0	0	0	0	1
0	0	0	1	1
0	0	1	0	0
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	0
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				
	01				
	11				
	10				

Z =

5. (1 + 2 valores) Suponha que a *cache* do PEPE (processador com 16 bits de endereço, endereço de byte) é de mapeamento direto, com uma capacidade de 1024 palavras e blocos de 8 palavras.

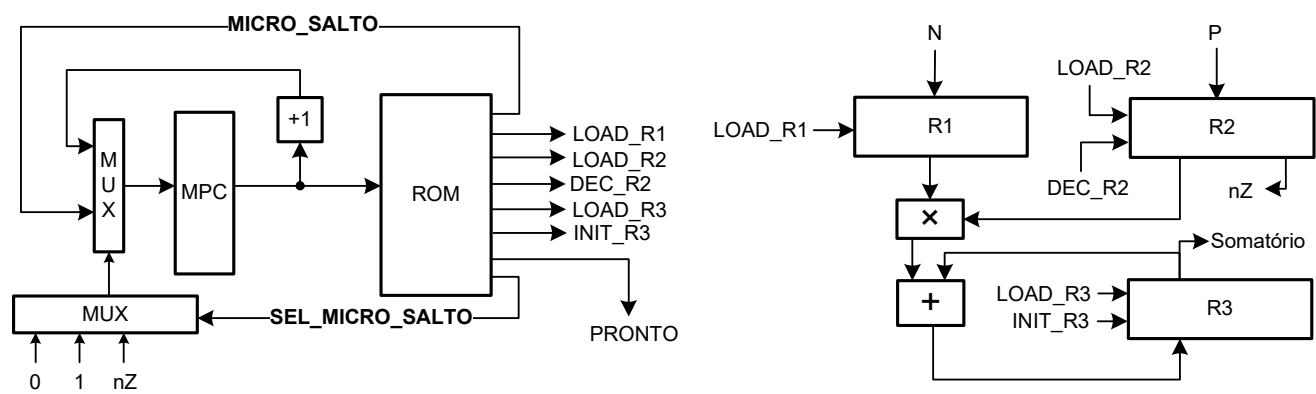
- a) Quantos bits são precisos para indicar qual o bloco em que cada palavra se situa (campo índice)?

- b) Suponha que o tempo de acesso em caso de *hit* e de *miss* é de 5 ns e 40 ns, respetivamente. Se o tempo médio de acesso for de 8,5 ns, qual é a *hit rate* média? Justifique.

6. (2 + 1 valores) Pretende-se construir um circuito microprogramado que implemente o somatório

$$\sum_{i=0}^{P-1} N * (P - i)$$

em que N, P > 1. O diagrama seguinte descreve o circuito. Os registos R1 e R2 recebem N e P, respetivamente. O registo R3 vai acumulando o resultado das somas sucessivas. O sinal DEC_R2 decrementa R2 de uma unidade. O sinal INIT_R3 inicializa R3 a zero. 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	INIT_R3	LOAD_R3	SEL_MICRO_SALTO	MICRO_SALTO
0	R1 ← N R2 ← P R3 ← 0								
1	R3 ← R3 + R1 * R2								
2	R2 ← R2 - 1								
3	(R2 != 0): MPC ← 1								
4	PRONTO ← 1 MPC ← 4								

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

7. (1,5 + 1,5 valores) Pretende executar um programa que necessita de 12 Mbytes de memória. O computador de que dispõe tem apenas 512 KBytes de memória física, mas o seu processador suporta memória virtual com páginas virtuais de 4 Kbytes (com endereçamento de byte).

a) Preencha a tabela seguinte com os valores que decorrem desta informação.

N.º mínimo de bits do espaço virtual	
N.º de páginas físicas existentes	

- b) Suponha que a TLB é totalmente associativa de 8 entradas e a memória física disponível para o programa está localizada a partir do endereço 30000H. Após reset, o processador acede aos seguintes endereços virtuais:

203B8H
4B35AH
3A1A0H
1E5BEH
4BFFEH
208FCH

Indique, na tabela a seguir, um possível estado do conteúdo da TLB imediatamente após estes acessos. Arbitre o que for necessário (não há solução única) e preencha apenas o que for relevante.

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

NOME	NÚMERO
------	--------

1. (2,5+1,5 valores) Considere o seguinte programa.

Periferico	EQU	5000H
PLACE	1000H	
X:	WORD	0
pilha:	TABLE	100H
fim_pilha:		
tab:	WORD	rotInt1
PLACE	0	
	MOV	SP, pilha
	MOV	BTE, tab
	MOV	R1, X
	EI	
ciclo:	MOV	R0, [R1]
	ADD	R0, 1
	MOV	[R1], R0
	JMP	ciclo
rotInt1:	MOV	R2, Periferico
	MOVB	R0, [R2]
	SHR	R0, 2
	MOVB	[R2], R0
	RET	

[illegible]

- a) O objetivo do programa principal é incrementar a variável X, que deve passar sucessivamente por todos os valores de 0000H a FFFFH (dando depois a volta e repetindo). A rotina de interrupção processa e atualiza o valor lido de um periférico. Este programa contém erros. Reescreva o programa na tabela do lado direito com os erros corrigidos, assinalando com uma cruz na coluna da direita as instruções que diferirem do original.
- Use apenas as linhas que precisar.

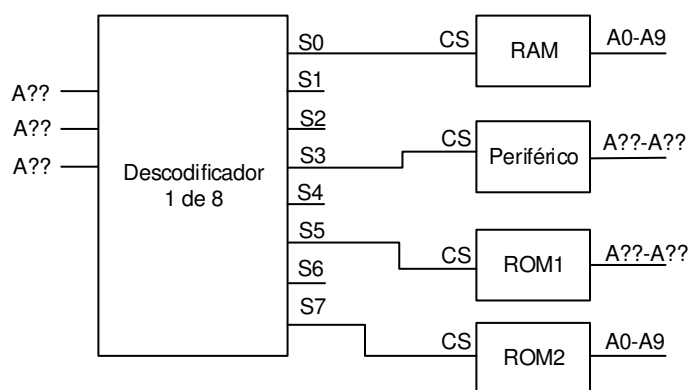
- b) Imagine que, sem ocorrência de interrupções, o programa principal demora 1 segundo a dar a volta ao valor da variável X (desde 0000H até 0000H novamente). Suponha agora que as interrupções são geradas periodicamente, que cada execução da rotina de interrupção demora 1 milissegundo e que o tempo de dar a volta ao valor da variável duplicou. Qual a frequência dos pedidos de interrupção neste caso?

pedidos de interrupção por segundo

2. (1 valor) Uma transmissão de dados por um barramento série assíncrono, com bit de paridade e 2 stop bits, efetua-se a um ritmo de transmissão de 1000 bits/seg. O recetor demora 10 milissegundos a processar cada byte recebido. Será que o recetor consegue processar de forma contínua os bytes enviados pelo emissor, se este o fizer ao ritmo máximo permitido pelo barramento? Justifique.

3. (2 valores) Suponha que duas pessoas com tipos de atividades diferentes vão a uma loja de informática comprar um portátil. Só há dois modelos. O modelo X tem um processador mais rápido (em *benchmark*) que o do Y, mas em compensação o Y tem um disco mais rápido (em *benchmark*) do que o do X. Uma das pessoas é administrativa e essencialmente usa editores de texto e e-mail. A outra trabalha com processamento de vídeo, conversores de formato AVI para MP4, etc. Que modelo deve cada uma destas pessoas escolher? Justifique.

4. (3 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 tabela com os bits de endereço a que quer o descodificador quer cada dispositivo deve ligar, bem como 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).



Bits do descodificador

Dispositivo	Bits de endereço	Capacidade em bytes (decimal)	Início (hexadecimal)	Fim (hexadecimal)
RAM	A0-A9			
Periférico		64	1800H	
ROM1		512		
ROM2	A0-A9			

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 simplificada.

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

		CD			
		00	01	11	10
AB	00				
	01				
	11				
	10				

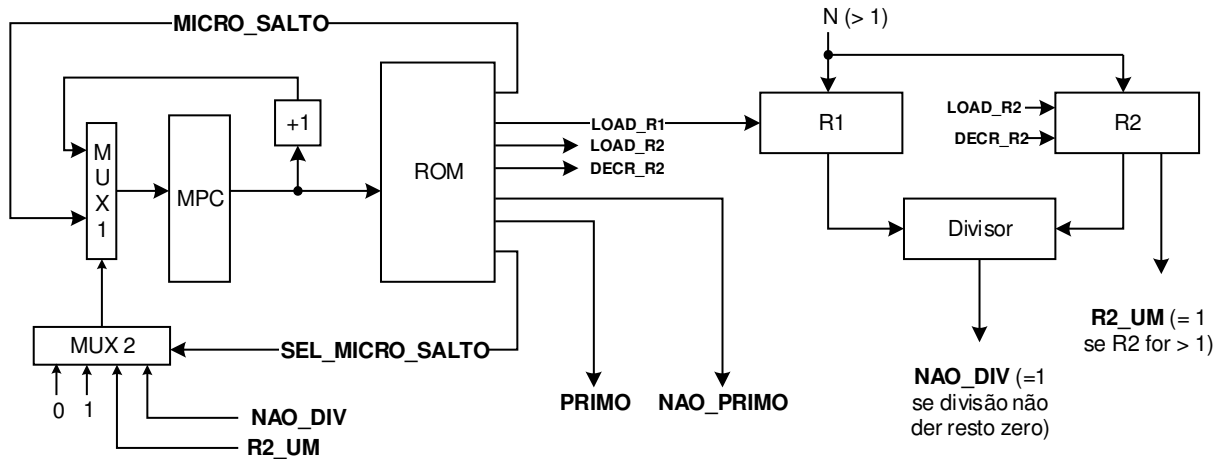
Z =

6. (1+1 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) Quantas linhas tem a cache?

b) Indique qual a linha em que a palavra com o endereço 1044H deverá ficar localizada, quando carregada na cache.

7. (2+1 valores) Considere o circuito seguinte, que permite testar se um dado número N (maior que 1) é primo. Depois de carregar o valor N nos registos R1 e R2, o algoritmo vai dividindo sucessivamente N por N-1, N-2, etc, até chegar a 1 (R2_UM = 1), o que acontecer indica que o número é primo. Pode também terminar quando o resto da divisão der zero (NAO_DIV = 0), caso em que o número não é primo. Nessa altura, um dos sinais de saída (PRIMO ou NAO_PRIMO) 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 algoritmo. Indique apenas os sinais relevantes em cada ciclo de relógio e deixe em branco as restantes células.

Endereço	Microinstrução (RTL)	LOAD_R1	LOAD_R2	DECR_R2	PRIMO	NAO_PRIMO	SEL_MICRO_SALTO	MICRO_SALTO
0	$R1 \leftarrow N;$ $R2 \leftarrow N;$							
1	$R2 \leftarrow R2 - 1$							
2	$(R2_UM = 1) : MPC \leftarrow 5$							
3	$(NAO_DIV = 1) : MPC \leftarrow 1$							
4	$NAO_PRIMO \leftarrow 1;$ $MPC \leftarrow 4$							
5	$PRIMO \leftarrow 1;$ $MPC \leftarrow 5$							

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

8. (1,5+1,5 valores) Imagine um processador com endereçamento de byte, capaz de endereçar um espaço virtual de 000000H até FFFFFFFH, enquanto o espaço de endereçamento físico vai de 00000H até FFFFFFFH. As páginas físicas têm uma dimensão de 1000H bytes.

a) Preencha a tabela seguinte com os valores que decorrem desta informação.

N.º bits do espaço virtual	
N.º bits do espaço físico	
N.º páginas virtuais	
N.º páginas físicas	

- b) Suponha que a TLB é totalmente associativa de 8 entradas e que, a certa altura, o seu conteúdo é o indicado na tabela seguinte. Com este conteúdo (sem o alterar), indique, na tabela a seguir, os endereços físicos a que o processador acede quando realiza acessos aos endereços virtuais indicados, explicando sucintamente o que acontece na tradução de virtual para físico em cada um destes acessos.

Posição da TLB	Bit validade	N.º página virtual (hexadecimal)	N.º página física (hexadecimal)
0	0	3B9	0C
1	0	207	31
2	1	0A2	3E
3	0	1EF	0F
4	1	4B8	25
5	0	0C3	1D
6	1	C31	1B
7	1	025	0C

Endereço virtual	Endereço físico	Justificação do que acontece
C314AFH		
0A2F04H		
2071D3H		
4B802EH		



NOME		NÚMERO	
------	--	--------	--

1. (1+1+1 valores) Considere o seguinte programa, que deve completar. Preencha apenas o que entender necessário.

PLACE	2000H	
saldo:	WORD	200
ilha:		100H
PLACE	0	
	MOV SP,	
	MOV BTE,	
rot1:	...	; código omitido
rot2:	PUSH	

- a) Complete a zona de dados, com o necessário para a pilha e a interrupção 1 e 2 funcionarem corretamente. Para simplificar, o código da rotina de interrupção 1 foi omitido;
- b) Complete o programa principal, com o necessário para a pilha e a interrupção 2 funcionarem corretamente. O corpo do programa principal deve terminar em ciclo infinito (salto para a própria instrução);
- c) Complete a rotina de interrupção 2 ("int2"), que deve duplicar o valor da variável "saldo" (na memória).

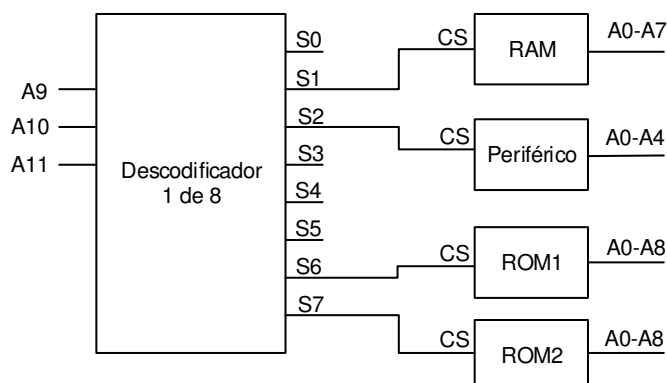
2. (1 valor) Considere um sistema com transferência de dados por DMA em regime de bloco (transferência de uma só vez) e uma interrupção periódica. Cada bloco de dados transferido tem 1000 palavras e cada palavra transferida por DMA demora 0,5 microssegundos. Qual a frequência máxima do sinal de interrupção? Justifique.

--

3. (2 valores) Tem de decidir entre dois processadores qual o melhor para correr a sua aplicação. O processador A tem um relógio de 3 GHz, a duração média de cada instrução é de 1,8 ciclos de relógio por instrução, e o compilador gera 200 Mega instruções. Por sua vez, o processador B tem um relógio de 2 GHz, a duração média de cada instrução é de 2,2 ciclos de relógio por instrução, e o compilador gera 100 Mega instruções. Indique qual o melhor processador. Justifique.

--

4. (3 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 o tamanho de cada dispositivo (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	Tamanho (bytes)	Início	Fim
RAM			
Periférico			
ROM1			
ROM2			

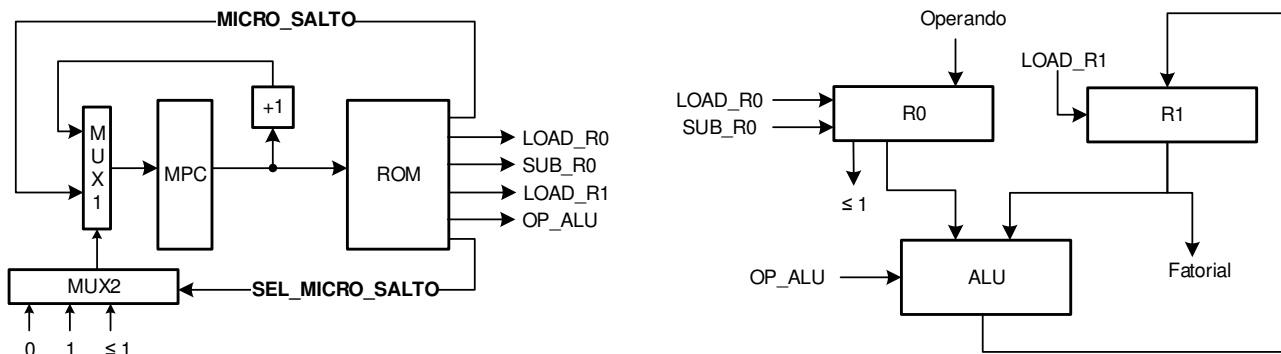
5. (3 valores) Considere a seguinte tabela de verdade, relativa a uma função de quatro entradas e uma saída. Simplifique a respetiva função, preenchendo a tabela de Karnaugh e escrevendo a expressão algébrica simplificada.

A	B	C	D	Z
0	0	0	0	1
0	0	0	1	1
0	0	1	0	0
0	0	1	1	0
0	1	0	0	1
0	1	0	1	1
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
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				
	01				
	11				
	10				

Z =

6. (2+1 valores) O circuito seguinte implementa a operação de fatorial do Operando, usando microprogramação. SUB_R0 decrementa R0 de uma unidade. O sinal ≤ 1 fica ativo (a 1) sempre que $R0 \leq 1$. A ALU suporta 4 operações (especificadas por OP_ALU): na sua saída pode aparecer $R0+R1$, $R0 \cdot R1$, 0 ou 1.



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

Endereço	Microinstrução (RTL)	LOAD_R0	LOAD_R1	SUB_R0	OP_ALU	SEL_MICRO_SALTO	MICRO_SALTO
0	$R0 \leftarrow \text{Operando}; R1 \leftarrow 1$						
1	$(\leq 1): MPC \leftarrow 5$						
2	$R1 \leftarrow R1 * R0$						
3	$R0 \leftarrow R0 - 1$						
4	$MPC \leftarrow 1$						
5	$MPC \leftarrow 5$						

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

7. (1+1 valores) Suponha que a *cache* do PEPE (processador com 16 bits de endereço, endereçamento de byte) é de mapeamento direto, com uma capacidade de 2048 palavras e blocos de 8 palavras.

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

Etiqueta	
Índice	
Palavra dentro do bloco	
Byte dentro da palavra	

- b) Na execução de instruções do tipo MOV R1, [R2], o núcleo do PEPE verificou que, em média, o valor pretendido estava na cache 90% das vezes, caso em que este demorava 3 ns a obter. Medindo-se um tempo médio de acesso de 5,7 ns, qual será o tempo de acesso a um valor quando este não está na cache?

 ns

8. (2+1 valores) Considere um processador com 24 bits de endereço, endereçamento de byte e suporte para memória virtual com páginas de 4K bytes. A memória física tem uma capacidade de 1 Mbyte.

a) Preencha a tabela seguinte:

Dimensão do espaço virtual	
Dimensão da página física	
Número de páginas virtuais	
Número de páginas físicas	

- b) Suponha que a TLB é uma *cache* totalmente associativa de 4 entradas, inicialmente vazia. Preencha o seu conteúdo após os acessos aos endereços virtuais seguintes e assumindo que a TLB será preenchida de cima para baixo, à medida que for sendo necessário.

Acessos realizados	
Endereço virtual	Endereço físico
23BA52H	
17BA4EH	
23BA0CH	
158054H	
47F054H	
17B1FEH	

TLB	
Página virtual	Página física
	34
	5C
	AE
	04



NOME		NÚMERO	
------	--	--------	--

1. (1 + 2 + 1 valores) Considere o seguinte programa, que usa rotinas de interrupção para alterar o valor de um display hexadecimal.

PLACE 1000H
contador: WORD 6
pilha: TABLE 100H
fim_pilha:

PLACE 0

MOV R2, contador

EI3

EI

fim: JMP fim

rot1: PUSH R1
MOV R1, [R2]
ADD R1, 2 ; incrementa
MOV [R2], R1 ; atualiza contador
POP R1
RFE

rot3: PUSH R1
MOV R1, [R2]
EI1
SUB R1, 1 ; decrementa
MOV [R2], R1 ; atualiza contador
POP R1
RFE

- a) Do lado esquerdo, complete a zona de dados e o programa principal com o necessário para a pilha e as interrupções 1 e 3 funcionarem corretamente (rotinas no lado direito);
- b) Suponha que os pedidos de interrupção 1 e 3 se alternam, com um segundo de intervalo, e que a interrupção 1 é a primeira a ser pedida. Indique de seguida a sequência dos 8 primeiros valores que as rotinas de interrupção escrevem no contador (e o número da interrupção em que cada valor é escrito). Justifique os dois primeiros;

Interrupção							
Valor							

--

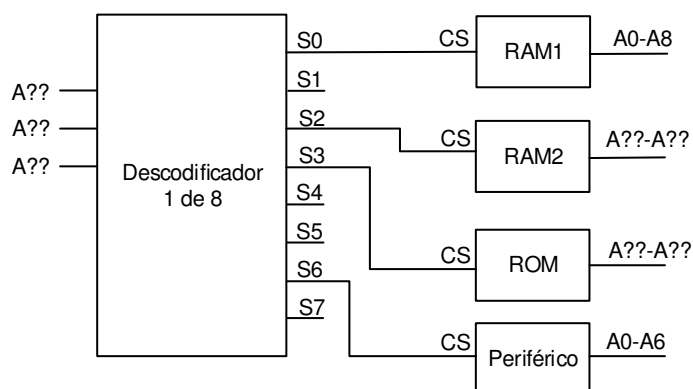
- c) Repita a alínea anterior (recomeçando o programa), mas supondo agora que na rotina rot3 a instrução EI1 passa para imediatamente antes do POP R1. Justifique as diferenças face ao caso anterior.

Interrupção							
Valor							

--

2. (2 valores) Uma transmissão de dados é feita por um barramento série assíncrono, com bit de paridade e 2 stop bits. O recetor demora 50 milissegundos a processar cada byte recebido, mas pode ir recebendo o byte seguinte (e apenas esse) durante esse processamento, processando-o imediatamente após processar o anterior. Supondo que o emissor usa um ritmo de transmissão de 1000 bits/seg, indique qual o tempo mínimo, em média, que o emissor tem de esperar entre acabar de enviar um byte e poder começar a enviar o seguinte. Justifique.

3. (3 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 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)
Descodificador				
RAM1	A0-A8			
RAM2		1 K	1000H	
ROM				1FFFH
Periférico	A0-A6			

4. (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 simplificada.

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

		CD			
		00	01	11	10
AB	00				
	01				
	11				
	10				

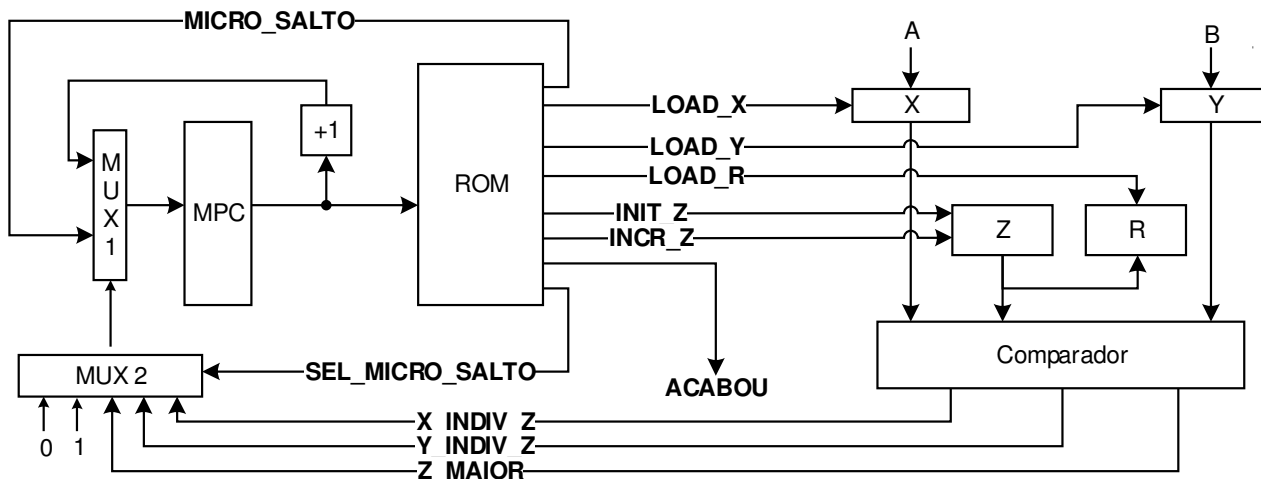
Z =

5. (1 + 2 valores) Suponha que a *cache* do PEPE (processador com 16 bits de endereço, endereço de byte) é de mapeamento direto, com uma capacidade de 512 palavras (blocos de 8 palavras).

a) Quantos blocos tem a cache?

- b) Suponha que um programa num PEPE sem cache gasta 20% do seu tempo de execução T em acessos de dados à memória e que num PEPE com cache os acessos à memória em caso de *cache hit* demoram 10% do tempo dos acessos em caso de *cache miss*. Assumindo uma *hit rate* de 80%, qual será o tempo de execução T_c do mesmo programa no PEPE com *cache*, em função de T? Justifique.

6. (2 + 1 valores) Considere o circuito seguinte, que permite obter no registo Z o mdc (máximo divisor comum) entre dois números A e B (positivos e maiores que 1). Depois de carregar os valores A e B nos registos X e Y, o algoritmo inicializa Z e R a 1 e vai testando sucessivamente se X e Y são divisíveis por Z, caso em que atualiza R com o valor de Z. Termina quando Z for maior que X ou que Y, altura em que R conterá o mdc, o sinal ACABOU é ativado e o programa fica em salto infinito na mesma microinstrução.



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

Endereço	Microinstrução (RTL)	LOAD_X	LOAD_Y	INIT_Z	INCR_Z	LOAD_R	ACABOU	SEL_MICRO_SALTO	MICRO_SALTO
0	$X \leftarrow A;$ $Y \leftarrow B;$ $Z \leftarrow 1;$								
1	$R \leftarrow Z$								
2	$(Z_MAIOR = 1) : MPC \leftarrow 7$								
3	$(X_INDIV_Z = 1) : MPC \leftarrow 6$								
4	$(Y_INDIV_Z = 1) : MPC \leftarrow 6$								
5	$R \leftarrow Z$								
6	$Z \leftarrow Z + 1;$ $MPC \leftarrow 2$								
7	$ACABOU \leftarrow 1;$ $MPC \leftarrow 7$								

- b) Se usarmos uma ROM de microprograma com 16 bits de largura, quantos bits **NÃO** serão necessários? Justifique.

7. (1,5 + 1,5 valores) Imagine um processador com endereçamento de byte, capaz de endereçar um espaço virtual de 000 000H até FFF FFFH, enquanto o espaço de endereçamento físico vai de 00000H até FFFFFH. As páginas físicas têm uma dimensão de 4 Kbytes.

a) Preencha a tabela seguinte com os valores que decorrem desta informação.

N.º bits do espaço virtual	
N.º bits do espaço físico	
N.º páginas virtuais	

- b) Suponha que a TLB é totalmente associativa de 8 entradas e há apenas 64 Kbytes de memória física, localizada a partir do endereço 10000H. Após reset, o processador acede aos seguintes endereços virtuais:

207 3B8H
4B8 35AH
0A2 1A0H
1EF 5BEH
4B8 FFEH
207 8FCH

Indique, na tabela a seguir, um possível estado do conteúdo da TLB imediatamente após estes acessos. Arbitre o que for necessário (não há solução única) e preencha apenas o que for relevante.

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