

Arquitetura de Computadores

2007/2008 2º Semestre

2º Teste (A) - 16/06/2008

INSTRUÇÕES:

- A duração da prova é de 1,5 horas.
- Responda apenas nos locais indicados.
- Identifique todas as folhas do teste com o seu número e nome de forma bem legível.
- A cotação das perguntas está indicada entre parênteses à direita do texto.
- Nas perguntas Verdade/Falso ou de escolha múltipla as respostas erradas descontam.

I

As perguntas deste grupo referem-se todas ao processador P3 e são independentes entre si.

1.1 Indique na tabela seguinte, usando linguagem de transferência de registos (RTL), quais as micro-operações realizadas pela microinstrução **7B0E7B9C** h. (2)

(pode usar como rascunho o espaço no fim desta folha)

Operações realizadas (em RTL):	
$CAR \leftarrow F2$	[1/4]
$IAK \leftarrow 0$	(opcional)
$EA \leftarrow EA+C$, flags ZC	[1/4, 1/4]
$M[EA] \leftarrow RD$	[1/4]

Nota: Utilize apenas as posições da tabela que considere necessárias

1.2 Considere que se pretende que as seguintes micro-operações se realizem num único ciclo de relógio:

$$!z ? CAR \leftarrow MLOOP, SBR \leftarrow CAR+1, RD \leftarrow RE, RE \leftarrow RD$$

Assumindo que a micro-rotina *MLOOP* se encontra codificada a partir do endereço 1A0h, indique na tabela seguinte a codificação de uma microinstrução que faz com que isso aconteça ou justifique que tal não é possível. (2)

NOTA: Preencher com 0, 1 ou X para os sinais indiferentes.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
0	M5		S R 1	S R 2	I A K	FM			CALU				M A	M B	M 2	M R B	RB			W M			W R	MD		M A D	RAD									
1				L S	MCOND			C C	L I	L F	CONST/NA																									
	1	0	0	X	X	1	0	0	1	1	0	1	X	X	X	1	1	0	1	0	0	0	0	0	1	1	0	1	1	1	0	1				

Zona de rascunho para a pergunta 1.1 (será totalmente ignorado durante a correcção)

0	1	1	1	1	0	1	1	0	0	0	0	1	1	1	0	0	1	1	1	1	0	1	1	1	0	0	1	1	1	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

1.3 Pretende-se implementar no processador P3 a instrução “FLOG op1” que efectua o cálculo do maior número natural não superior ao logaritmo na base 2 de um determinado valor inteiro op1 (o resultado será devolvido em op1, como é comum no Assembly do P3).

Considerando que a função logaritmo está apenas definida para valores superiores a zero, convencionou-se que a função a implementar deverá retornar o valor -1 sempre que o operando de entrada for não positivo. Formalmente, a função é definida através da seguinte expressão:

$$\text{flog}(n) = \begin{cases} \lfloor \log_2(n) \rfloor, & n \geq 1 \\ -1, & n \leq 0 \end{cases} \quad n \in \mathbb{Z}$$

Após a sua execução, deverão ser actualizadas as flags Z e N.

Preencha a tabela seguinte com o micro-programa correspondente à fase de execução da instrução FLOG. Deverá utilizar exclusivamente as microinstruções disponibilizadas na lista abaixo. Apenas deverá indicar o número de cada uma das microinstruções utilizadas. (3)

FLOG0:	8	[SBR<-CAR+1, CAR<-F1]
FLOG1:		R8<-R0+R0, Flags ZN
FLOG2:	29	[RD+R0]
FLOG3:		z?CAR<-FLOG12
FLOG4:	17	[shl RD]
FLOG5:		c?CAR<-FLOG12
FLOG6:	11	[RD<-shr RD]
FLOG7:		z?CAR<-FLOG10
FLOG8:	48	[R8<-R8+1, Flags ZN]
FLOG9:		CAR<-FLOG6
FLOG10:	40	[RD<-R8]
FLOG11:	4	[CAR<-WB]
FLOG12:		RD<-R8, Flags ZN, CAR<-WB

Nota: Nas posições livres indique o número da instrução (de acordo com a lista abaixo)

Nº	Micro-Instrução
1	CAR<-IF0
2	CAR<-IF1
3	CAR<-IH0
4	CAR<-WB
5	CAR<-F1
6	CAR<-F2
7	CAR<-CAR+1
8	SBR<-CAR+1, CAR<-F1
9	SBR<-CAR+1, CAR<-F2
10	SBR<-CAR+1, CAR<-WB
11	RD<-shr RD
12	RD<-shl RD
13	SD<-shr SD
14	SD<-shl SD
15	EA<-shr EA
16	EA<-shl EA
17	shl RD
18	shr RD

Nº	Micro-Instrução
19	shl SD
20	shr SD
21	shl EA
22	shr EA
23	SD+R0
24	SD+R8
25	SD+R9
26	SD+R10
27	SD+EA
28	SD+RD
29	RD+R0
30	RD+R8
31	RD+R9
32	RD+R10
33	RD+SD
34	RD+EA
35	SD<-R8
36	SD<-R9

Nº	Micro-Instrução
37	SD<-R10
38	SD<-RD
39	SD<-EA
40	RD<-R8
41	RD<-R9
42	RD<-R10
43	RD<-SD
44	RD<-EA
45	R8<-R0
46	R8<-R8+1
47	R8<-R8-1
48	R8<-R8+1, Flags ZN
49	R8<-R8-1, Flags ZN
50	R9<-R0
51	R9<-R9+1
52	R9<-R9-1
53	R9<-R9+1, Flags ZN
54	R9<-R9-1, Flags ZN

II

2.1 Considere um sistema com um processador de 16 bits com 20 linhas de endereço – de A19 (maior peso) a A0 (menor peso). Neste sistema existem duas zonas de memória e uma zona para entradas/saídas (E/S):

- ROM M1 – Possui 508 k palavras e ocupa a gama de endereços mais elevados.
- RAM M2 – Possui 512 k palavras e ocupa a gama de endereços mais baixos.
- E/S M3 – Ocupa 4k palavras a partir de 80000h (inclusive).

a) Assinale abaixo, com um círculo, a letra que corresponde às funções de descodificação destas zonas de memória. (1,5)

(Notas: MS – memory select ; -X = not X)

- A: MS1 = A19; MS2 = -A19; MS3 = A19.-A18.-A17.-A16.-A15.-A14.-A13.-A12
 B: MS1 = A19.A18; MS2 = -A19; MS3 = A19.-A18.-A17.-A16.-A15.-A14.-A13.-A12
 C: MS1 = A19.-MS3; MS2 = -A19; MS3 = A19.-A18.-A17.-A16.-A15.-A14.-A13
 D: MS1 = A19.-MS3; MS2 = -A19; MS3 = A19.-A18.-A17.-A16.-A15.-A14.-A13.-A12
 E: MS1 = -A19; MS2 = A19; MS3 = A19.-A18.-A17.-A16.-A15.-A14.-A13.-A12
 F: Nenhuma das funções anteriores está correcta.

b) Pretende-se que o espaço de memória reservado para as entradas/saídas tenha capacidade para alojar 16 controladores de periféricos com as seguintes características (por ordem crescente de endereços, PS – *peripheral select*):

	Controlador	Palavras de controlo e estado	Palavras de dados
PS1 = MS3.-W.-X.-Y.-Z	Interrupções	2	0
PS2 = MS3.-W.-X.-Y.Z	Rede local	64	2k em FIFO em <i>hardware</i>
PS3 = MS3.-W.-X.Y.-Z	Disco	128	4k em FIFO em <i>hardware</i>
PS4 a PS16	Outros	Menos de 64 palavras para controlo, estado e dados por cada controlador	

Nas funções de descodificação W, X, Y e Z são sinais de endereços do processador (A19-A0). Indique a linha de endereço que corresponde a cada letra. (0,5)

	Linha de endereço
W	A11
X	A10
Y	A9
Z	A8

2.2 O processador P11 tem um conjunto de instruções logicamente igual ao do P3 mas todas as instruções são codificadas em duas palavras. O P11 tem um espaço de endereçamento de 64k palavras de 16 bits e caches separadas para código e dados.

A cache de dados tem mapeamento directo, tem capacidade de 4k palavras e blocos de 4 palavras. A cache é *write-back*.

A cache de código será detalhada mais abaixo.

Considere ambas as caches inicialmente vazias. Neste sistema executa-se o seguinte programa.

```

1  MASK    EQU    01h
2  SIZE    EQU    1000h

3
4  A       TAB    SIZE

5
6         ORIG   0000h
7         MOV    R7, MASK
8         MOV    R4, SIZE
9         DEC    R4
9  Loop:   TEST   R7, M[R4+A]
10        BR.Z   R
11        SHL   M[R4+A], 1
12        BR    Next
13  R:     SHR   M[R4+A], 1
14  Next:  DEC   R4
15        BR.NN Loop

```

a) Assinale abaixo, com um círculo, a letra que corresponde ao valor aproximado do *hit rate* na cache de dados ao executar este programa? **(1,5)**

A: 83%	B: 92%	C: 33%	D: 50%	E: 75%	F: Nenhuma das opções anteriores.
--------	---------------	--------	--------	--------	-----------------------------------



b) Sem alterar a capacidade da cache de dados de que forma é possível melhorar significativamente o *hit rate*? **(1)**

<p>→ A: Aumentando o número de vias de associatividade.</p> <p>B: Aumentando a dimensão do bloco.</p> <p>C: Mudando a política de escrita.</p> <p>D: Já não é possível melhorar o <i>hit rate</i>.</p> <p>E: Mudando a política de substituição.</p> <p>F: Nenhuma das opções anteriores.</p>

c) A cache de código tem 4 (quatro) palavras, blocos de 1 palavra e é totalmente associativa. Assinale abaixo, com um círculo, a letra que corresponde ao *hit rate* nesta cache ao executar o programa. **(0,5)**

A: 0%	B: 25%	C: 33%	D: 50%	E: 75%	F: Nenhuma das opções anteriores.
--------------	--------	--------	--------	--------	-----------------------------------



Número: _____ Nome: _____

d) Assinale abaixo, com um círculo, a letra que corresponde à dimensão do bloco que maximizava o *hit rate* na cache de código, mantendo as restantes características da cache. (0,5)

A: 1 palavra.	B: 2 palavras.	C: 3 palavras.	D: 4 palavras.	E: O <i>hit rate</i> não varia com a dimensão do bloco.	F: Nenhuma das opções anteriores.
---------------	----------------	----------------	----------------	---------------------------------------------------------	-----------------------------------



e) Assinale abaixo a dimensão mínima que deverá ter a cache de código, mantendo-se as suas restantes características originais, para que o *hit rate* ao executar este programa seja próximo de, ou até, 100%? (0,5)

A: 1 palavra.	B: 2 palavras.	C: 4 palavras.	D: 8 palavras.	E: 16 palavras.	F: Nenhuma das opções anteriores.
---------------	----------------	----------------	----------------	-----------------	-----------------------------------



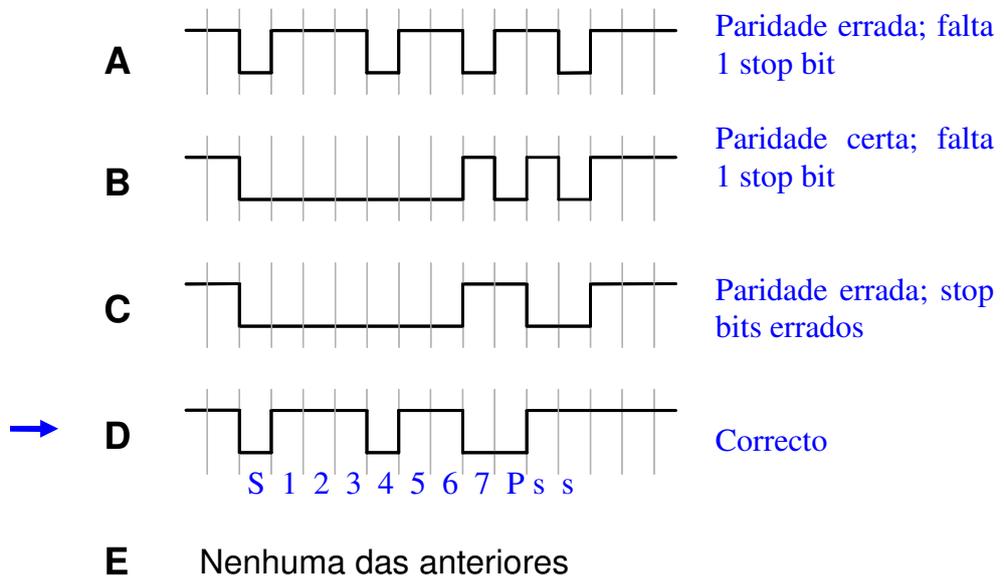
f) Num sistema sem cache o tempo de acesso à memória é 100 ns. Assinale abaixo o tempo de acesso à cache abaixo do qual é vantajosa a inclusão de cache entre o processador e a memória ao executar um programa com *hit rate* de 80%. Só é lançado o acesso à memória após o teste do *hit/miss* na cache. (1)

→	A: É sempre vantajoso incluir cache.
	B: Tempo de acesso à cache < 40 ns.
	C: Tempo de acesso à cache < 60 ns.
	D: Tempo de acesso à cache < 80 ns.
	E: Tempo de acesso à cache < 90 ns.
	F: Nenhuma das opções anteriores está correcta.

Página deixada em branco propositadamente

III

3.1 Considere uma comunicação série assíncrona com as seguintes características específicas: 7 bits de dados, paridade ímpar e 2 stop bits. Analise os diagramas temporais abaixo e assinale com um círculo a letra correspondente ao diagrama que satisfaz às características indicadas. (1,5)



3.2 Pretende-se transferir um ficheiro com 1200 bytes usando comunicação série assíncrona com as seguintes características específicas: 8 bits de dados, sem bit de paridade e um stop bit. O ritmo de transmissão é de 2400 bps. Assinale abaixo qual o tempo mínimo necessário à transferência do ficheiro. (1)

A: 0,2 s	B: 0,5 s	C: 2 s	D: 5 s	E: 20 s	F: Nenhuma das opções anteriores.
----------	----------	--------	--------	---------	-----------------------------------



3.3 Dois sistemas dialogam entre si usando comunicação série síncrona em que a *flag* delimitadora é constituída pelo padrão 0111 1110 e em que é usada a técnica *bit stuffing*.

Se o receptor receber a sequência

flag 0 1 1 1 1 1 1 0 1 1 1 1 1 0 0 0 *flag*

Assinale abaixo qual a sequência de bits que foi transmitida na linha de comunicação. (1)

A: 0 1 1 1 1 1 1 0 1 1 1 1 1 0 0 0

→ B: 0 1 1 1 1 1 1 0 0 1 1 1 1 1 0 1 0 1 1 1 1 1 1 0 0 0 0 0 1 1 1 1 1 1 1 0

C: 0 1 1 1 1 1 1 0 0 1 1 1 1 1 1 0 1 0 1 1 1 1 1 1 0 0 0 0 1 1 1 1 1 1 1 0

D: 0 1 1 1 1 1 1 0 1 0 1 1 1 1 1 1 0 0 0

E: Nenhuma das anteriores

3.4 Considere um sistema que inclui um microprocessador, um PIC (Programmable Interrupt Controller) e alguns periféricos. Pretende-se que indique e ordene, da lista apresentada a seguir, as acções que ocorrem quando um periférico efectua um pedido de interrupção ao PIC.

- 1 - Periférico efectua pedido de interrupção ao PIC
- 2 - Processador termina a instrução em curso
- 3 - Processador activa linha INT
- 4 - Processador activa linha IAK
- 5 - PIC activa linha INT
- 6 - PIC activa linha IAK
- 7 - Processador verifica se tem as interrupções *enabled*; se sim avança; se não ignora pedido
- 8 - PIC coloca no bus de dados o vector de interrupção
- 9 - PIC coloca no bus de dados o endereço da rotina de tratamento de interrupção
- 10 - Processador inicia a execução da rotina de tratamento de interrupção indicada
- 11 - Processador acede a uma tabela em memória e obtém o endereço da rotina de interrupção
- 12 - Tem início a execução da rotina de tratamento de interrupção

Assinale abaixo, com um círculo, a letra correspondente às acções correctas e ordem correcta.(1,5)

- | | |
|---|-------------------------------------------------|
| | A: 4, 1, 5, 2, 7, 8, 11, 12 |
| | B: 1, 5, 2, 7, 4, 9, 10 |
| | C: 1, 5, 4, 2, 7, 11, 12 |
| | D: 1, 4, 2, 7, 9, 10 |
| → | E: 1, 5, 2, 7, 4, 8, 11, 12 |
| | F: Nenhuma das opções anteriores está correcta. |

3.5 Apresentam-se em seguida várias afirmações. Assinale quais as verdadeiras (use **V**) e quais as falsas (use **F**). (1)

F	A: Um controlador DMA pode ser ligado a qualquer processador
F	B: Uma transferência por DMA é desencadeada por um periférico que activa a linha Bus-Request
V	C: Durante uma transferência por DMA o processador não pode aceder aos barramentos do sistema
F	D: Uma transferência de dados memória/periférico efectuada por interrupção é normalmente mais eficiente que uma transferência efectuada por DMA