

Arquitectura de Computadores
2011/2012 2º Semestre
2º Teste - Repescagem - 27/06/2012

INSTRUÇÕES:

- A duração da prova é de 2 horas.
- Identifique todas as folhas do teste com o seu número e nome de forma bem legível.
- Responda apenas nos locais indicados. As folhas serão separadas e corrigidas isoladamente.
- 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 **EC5555F9** h. (2)

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

Operações realizadas (em RTL):	
$R9 \leftarrow 0555\text{ h}$	[1/4]
$RE \leftarrow R9$	[1/4]
$SBR \leftarrow CAR+1$	[1/4]
$CAR \leftarrow F2$	[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:

$$SD \leftarrow SD-1, \text{ Flags Z}, CAR \leftarrow SBR, M[SD] \leftarrow RD$$

Indique na tabela seguinte a codificação de uma micro-instrução 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				
0	0	1	X	X	0	1	0	0	0	0	0	1	0	0	0	0	X	1	1	1	0	1	1	1	0	0	1	1	0	1	1

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

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

1.3 Pretende-se implementar no processador P3 a instrução “SETM op1, op2” que preenche com FFFFh o número de posições de memória indicado por op2, começando no endereço indicado por op1 (ver exemplos abaixo). A instrução SETM afecta op1 que fica a apontar para a posição a seguir à última que foi preenchida com FFFFh.

O operando op2 está representado em binário puro (valor ≥ 0). Se op2 for 0, a instrução limita-se a activar a flag Z e termina (op1 mantém o mesmo valor que tinha antes).

Exemplos: SETM R1, R2

Se R1 = 8000h e R2 = 5 → Preenche 5 posições de memória com FFFFh, começando em 8000h (8000h a 8004h); R1 (op1) fica com o valor 8005h.

Se R1 = 8000h e R2 = 0 → Apenas activa a flag Z.

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

SETM0:	SBR←CAR+1, CAR←F2
SETM1:	23 SD←R0, Flags Z
SETM2:	z?CAR←SETM9
SETM3:	50 R9←R0
SETM4:	R9←!R9
SETM5:	46 M[RD]←R9
SETM6:	RD←RD+1
SETM7:	27 SD←SD-1
SETM8:	!z?CAR←SETM5
SETM9:	4 CAR←WB

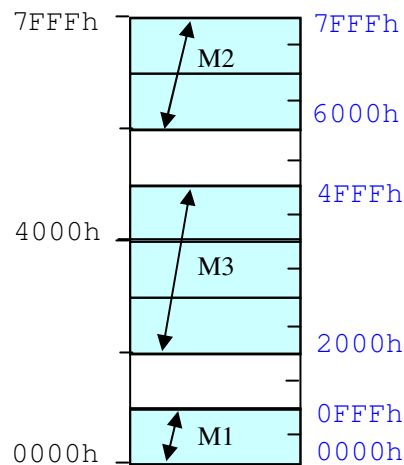
Nº	Micro-Instrução
1	CAR←IF0
2	CAR←IF1
3	CAR←IH0
4	CAR←WB
5	CAR←F1
6	CAR←F2
7	CAR←CAR+1
8	SBR←CAR+1, CAR←F1
9	SBR←CAR+1, CAR←F2
10	SBR←CAR+1, CAR←WB
11	RD←shr RD
12	RD←shl RD
13	SD←shr SD
14	SD←shl SD
15	EA←shr EA
16	EA←shl EA
17	shl RD
18	shr RD

Nº	Micro-Instrução
19	shl SD
20	shr SD
21	shl EA
22	shr EA
23	SD←R0, Flags Z
24	SD←R8
25	SD←R9, Flags Z
26	SD←SD+1
27	SD←SD-1
28	SD←SD-1, Flags ZN
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	M[SD]←R9
46	M[RD]←R9
47	M[R9]←R0
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 o seguinte mapa de memória:



a) Indique o número de linhas de endereço do sistema. (0,5)

15 (2¹⁵ = 32 K palavras)

b) Para cada uma das memórias representadas (M1 a M3) indique a respectiva dimensão em K palavras. (0,5)

M1: 4 K

M2: 8 K

M3: 12 K

c) Indique, em hexadecimal, os endereços iniciais e finais das memórias M1 e M3. (1)

M1: Início: 0000 h

Fim: 0FFF h

M3: Início: 2000 h

Fim: 4FFF h

d) Indique as funções de selecção das memórias M1 a M3. (1)

M1: -A14.-A13.-A12

M2: A14.A13

M3: -A14.A13 + A14.-A13.-A12

(página propositadamente em branco; pode usar para rascunho)

III

3.1 O troço de programa seguinte é executado num sistema com o processador P3 com caches de dados e de código. A cache de dados tem 2 palavras, bloco de 2 palavras, é de mapeamento directo e segue uma política *write-back* com *write-allocate*.

```

MASK      EQU      1h
SIZE      EQU      1000h

DATA      ORIG      8000h
          TAB      SIZE

          ORIG      0000h
          MOV      R1, SIZE
          MOV      R2, DATA
          MOV      R3, MASK
          CMP      R1, R0
          BR.Z     END
LOOP:     TEST     R2, R3
          BR.NZ    ODD
          INC      M[R2]
ODD:      INC      M[R2]
          INC      R2
          DEC      R1
          BR.NZ    LOOP
END:      BR      END
    
```

a) Assinale abaixo, com um círculo, qual a taxa de sucesso na cache de dados ao executar o programa. (1)



A: ~ 0%	B: ~ 33%	C: ~ 50%	D: ~ 83%	E: ~ 92%	F: Nenhuma das opções anteriores.
---------	----------	----------	----------	----------	-----------------------------------

Nos endereços pares faz dois INC e nos endereços ímpar faz um (notar que cada INC faz dois acessos à mem. (lê, inc., escreve). Inicialmente Miss e carrega bloco (2 posições) e Hit (primeiro INC), depois Hit e Hit para o segundo INC (tudo isto na mesma posição da cache). Passa para o endereço seguinte (ímpar) e faz um INC a que corresponde Hit e Hit (2ª posição do bloco). Depois repete. Total hits = 5 / Total acessos = 6 → ~83%

b) Se a dimensão da cache de dados passar para 1024 palavras, mantendo as demais características, indique que alteração teria na taxa de sucesso. (0,5)



A: Aumenta	B: Diminui	C: Mantém
------------	------------	-----------

Inicialmente cache apenas tem uma linha (1 bloco) com duas palavras. Passando para 1024 palavras, passa a ter 512 linhas (blocos). No entanto, o vector DATA não cabe todo na cache (1000h → 13 bits → $2^{13} = 8k$ palavras). Assim, o comportamento descrito antes mantém-se, em cada iteração do ciclo, pelo que a taxa de sucesso se mantém.

c) Indique, para a cache de código, qual a configuração que permite a melhor taxa de sucesso e minimiza a dimensão da cache. (0,5)

- A: 1024 palavras, bloco de 1 palavra
- B: 512 palavras, bloco de 4 palavras
- C: 256 palavras, bloco de 4 palavras
- D: 16 palavras, bloco de 8 palavras
- E: 8 palavras, bloco de 8 palavras
- F: 4 palavras, bloco de 4 palavras

O ciclo que é repetido (entre Loop: ... e BR.NZ Loop) contém 7 instruções, em que cada uma ocupa 1 palavra de mem. O ciclo ocupa pois 7 palavras de código. Por outro lado, notar que essas 7 palavras cabem dentro de um mesmo bloco de 8 posições (as instruções antes do ciclo ocupam exactamente 8 palavras; por isso, TEST R2,R3 fica na primeira posição do bloco, BR.NZ ODD fica na segunda posição, etc). Assim, basta a cache ter um bloco com 8 palavras para todo o ciclo caber lá, originando uma taxa de sucesso de ~100%. As demais opções ou não são tão boas ou implicam que a cache tenha uma maior dimensão.

3.2 Considere um sistema em que o tempo de acesso à cache de código é 80ns e o tempo de acesso à memória primária é 250ns. Sabendo que a introdução da cache de código permitiu aumentar a rapidez em 2 vezes (tempo de acesso passou para metade), calcule a taxa de sucesso nessa cache. (1)



A: 50%	B: 62%	C: 70%	D: 82%	E: 90%	F: Nenhuma das opções anteriores.
--------	--------	--------	--------	--------	-----------------------------------

Sem cache $t=250$ ns. Com cache passou a metade (125ns).

$T_{\text{acesso}} = T_{\text{cache}} + \text{miss_rate} * T_{\text{mem.principal}} \rightarrow 125 = 80 + \text{miss_rate} * 250$ $\text{miss_rate} = 45 / 250 = 0,18$
 $\text{hit_rate} = 1 - \text{miss_rate}$ $\text{hit_rate} = 1 - 0,18 = 0,82 \rightarrow 82\%$

3.3 Considere um sistema de memória virtual paginada. O espaço de endereçamento virtual é de 32k palavras. O campo de deslocamento do endereço virtual tem 5 bits. O espaço de endereçamento físico é de 1k palavras.

(10 bits, dos quais 5 são para o deslocamento)

a) Indique a dimensão das páginas. (0,5)

$$2^5 = \underline{32 \text{ palavras}}$$

b) Indique quantas páginas tem o espaço de endereçamento virtual. (0,5)

$$32k = 2^{15} \quad 15-5 = 10 \quad 2^{10} = \underline{1 \text{ K}}$$

c) Pretende-se aceder ao endereço virtual 3F31h. Indique, em hexadecimal, o índice da tabela de páginas que será consultado. (0,5)

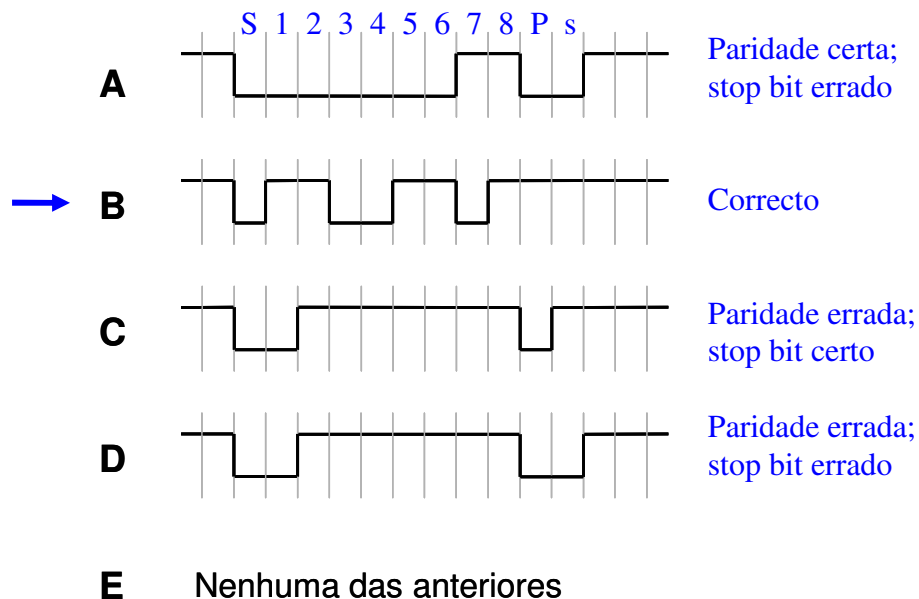
$$\underline{011 \ 1111 \ 0011 \ 0001} \quad \underline{1F9 \text{ h}}$$

d) No acesso ao endereço virtual 3F31h foi lida da tabela de páginas o valor 7h. Indique em hexadecimal o endereço físico acedido. (0,5)

$$\begin{array}{ccccccc} 011 & 1111 & 001 & 1 & 0001 & & \\ & 00 & 111 & 1 & 0001 & & \\ \hline & & & & & & \end{array} \quad \underline{0F1 \text{ h (ou F1 h)}}$$

IV

4.1 Considere uma comunicação série assíncrona com as seguintes características específicas: 8 bits de dados, paridade par e 1 stop bit. Analise os diagramas temporais abaixo e assinale com um círculo a letra correspondente ao diagrama que satisfaz às características indicadas. (1,5)



4.2 Pretende-se transferir um ficheiro com 2400 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. Indique qual o tempo mínimo necessário à transferência do ficheiro. (1)

$$2400 * (1+8+1) / 2400 = 240 / 24 = 10 \text{ s}$$

4.3 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*.

Considere que se pretende transmitir a seguinte sequência de bits

0 1 1 1 1 1 1 1 0 1 1 1 1 1 0 1 1 1 1 1 0 1

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

- A: 0 1 1 1 1 1 1 0 0 1 1 1 1 1 0 1 1 0 1 1 1 1 1 0 1 0 1 1 1 1 1 1 0
- B: 0 1 1 1 1 1 1 0 0 1 1 1 1 1 0 1 1 0 1 1 1 1 1 0 0 1 0 1 1 1 1 1 1 0
- C: 0 1 1 1 1 1 0 1 1 0 1 1 1 1 1 1 0 0 1
- D: 0 1 1 1 1 1 0 1 1 0 1 1 1 1 1 1 0 1
- E: Nenhuma das anteriores

4.4 Apresentam-se em seguida várias afirmações. Assinale quais as verdadeiras (use **V**) e quais as falsas (use **F**). (1,5)

F	A: Considere dois sistemas que interagem usando um protocolo de duplo <i>handshake</i> . Não é possível ao sistema emissor saber quando o sistema receptor leu a informação.
F	B: Numa cache existe informação adicional de controlo associada a cada bloco. O bit de controlo <i>dirty</i> identifica que os dados não são válidos e que necessitam ser lidos de memória primária.
V	C: Um processador que tenha apenas uma linha de interrupção pode ser ligado a vários periféricos e todos eles podem gerar interrupções.
V	D: Durante uma transferência por DMA o processador pode continuar a executar instruções caso o sistema disponha de cache.
F	E: Num sistema com DMA, o processador perde imediatamente o acesso aos barramentos assim que o DMA necessitar efectuar uma transferência.
F	F: Num sistema com DMA é preferível ter uma cache que siga uma política de escrita <i>write-through</i> em vez de <i>write-back</i> .