



Arquitectura de Computadores

Ano Lectivo de 2009/2010

2º Semestre

Repescagem do 2º Teste 2 de Julho de 2010 Duração: 1h30+0h30

- **O teste é sem consulta**, apenas tem disponível o anexo que lhe deverá ter sido entregue com o teste. Por favor, não escreva nesse anexo e devolva-o no final do teste.
- Resolva o teste no próprio enunciado, o espaço reservado para cada pergunta é suficiente para a sua resposta. Tenha em atenção que cada grupo deve ficar em folhas separadas. Utilize as costas das folhas para rascunho.
- Identifique todas as folhas que entregar, **folhas não identificadas não serão cotadas!**
- Responda ao teste com calma. Se não sabe responder a uma pergunta, passe à seguinte e volte a ela no fim.

I. (1,5 + 2 + 2,5 = 6 val.)

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

1. Indique quais as micro-operações realizadas pela micro-instrução 945112B9h.

Operações realizadas (em RTL):

NOTA1: Utilize apenas as posições da tabela que considere necessárias.

NOTA2: Pode usar como rascunho o espaço disponibilizado junto ao rodapé da página.

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

$CAR \leftarrow F2, M[SP] \leftarrow SD, SP \leftarrow SP - 1, \text{Flags } Z, N$

Indique na tabela seguinte a codificação de uma micro-instrução que faz com que isso aconteça ou justifique que tal não é possível.

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		SRI	SR2	IAK	FM				CALU				MA	MB	M2	MRB	RB				WM		WR	MD	MAD	RAD									
1					LS	MCOND		CC	LI	LF	CONST/NA																										

--

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

3. Independentemente da largura da palavra do processador utilizado (ex: 16-bits, 32-bits, 64-bits etc.), muitas arquiteturas de processadores permitem a realização da mesma operação aritmética sobre vários operandos de 8-bits, armazenados numa única palavra. Desta forma, torna-se possível acelerar a execução de programas, através do processamento em simultâneo de várias amostras do sinal a processar.

Pretende-se implementar no processador P3 a instrução SET8U op1, op2. Esta instrução realiza a inicialização do octeto menos significativo (LSB) e do octeto mais significativo (MSB) do operando op1 com o valor do octeto menos significativo de op2:

$$[MSB(op1)] = [LSB(op1)] = [LSB(op2)] \quad (1)$$

Assume-se que todos os operandos correspondem a números sem sinal. Para além de realizar a inicialização indicada, esta instrução deverá ainda actualizar a flag Z, de acordo com o valor retornado em op1. Ilustra-se, de seguida, uma possível invocação da operação SET8U, bem como do valor resultante da respectiva execução.

```

...
MOV    R1, 1Ah    ; R1 = 001Ah
SET8U  R4, R1     ; R4 = 1A1Ah
...

```

Preencha a tabela com o micro-programa correspondente à fase de execução da instrução SET8U. Deverá utilizar exclusivamente as micro-instruções disponibilizadas na tabela abaixo. Apenas deverá indicar na folha de resposta o número de cada uma das micro-instruções utilizadas.

SET8U_0:	
SET8U_1:	R10<-0FFh
SET8U_2:	
SET8U_3:	
SET8U_4:	R8<-008h
SET8U_5:	
SET8U_6:	
SET8U_7:	!z?CAR<-SET8U_5
SET8U_8:	
SET8U_9:	

NOTA: Nas posições livres indique o número da micro-instrução (de acordo com a lista abaixo).

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

Nº	Micro-Instrução
19	SD<-SD ∧ RD, Flag Z
20	SD<-SD ⊕ RD, Flag Z
21	RD<-RD ∨ SD, Flag Z
22	RD<-RD ∧ SD, Flag Z
23	RD<-RD ⊕ SD, Flag Z
24	SD<-RD
25	SD<-EA
26	RD<-SD
27	RD<-EA
28	EA<-SD
29	EA<-RD
30	SD<-SD ∨ R10
31	SD<-SD ∧ R10
32	SD<-SD ⊕ R10
33	EA<-EA ∨ R10
34	EA<-EA ∧ R10
35	EA<-EA ⊕ R10
36	RD<-shl RD

Nº	Micro-Instrução
37	RD<-shr RD
38	EA<-shl EA
39	EA<-shr EA
40	RD<-shl RD, Flag Z
41	RD<-shr RD, Flag Z
42	EA<-shl EA, Flag Z
43	EA<-shr EA, Flag Z
44	R8<-R8 + 1
45	R8<-R8 + 1, Flag Z
46	R8<-R8 - 1
47	R8<-R8 - 1, Flag Z
48	R10<-R10 + 1
49	R10<-R10 + 1, Flag Z
50	R10<-R10 - 1
51	R10<-R10 - 1, Flag Z
52	SD<-SD+R8
53	RD<-RD+R8
54	EA<-EA+R8

II. (2,5_{0,5+0,5+0,5+1} + 3,5_{1+0,5+0,5+0,5+1} + 2₁₊₁ = 8 val.)

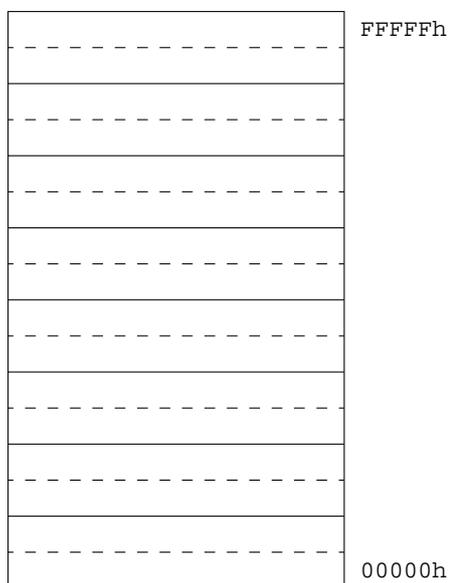
1. Considere um sistema com um processador que possui 20 linhas de endereço – de A19 (maior peso) a A0 (menor peso) – e 8 linhas de dados. Neste sistema existem duas memórias que possuem as seguintes funções de selecção:

$$\text{Mem1} = \overline{A19}.\overline{A18}$$

$$\text{Mem2} = A19.A18.\overline{A17}.\overline{A16}$$

a) Indique a capacidade da memória Mem2: _____

- b) A figura seguinte representa o mapa de memória do sistema. Assinale nessa figura a localização das duas memórias.



c) Indique, em hexadecimal, o último endereço ocupado pela Mem1: _____

- d) Pretende-se adicionar uma terceira memória ao sistema (Mem3) que deverá ocupar um espaço de endereçamento de 256k palavras a partir de 80000h.

Indique a função de selecção dessa memória. Mem3 = _____

2. O processador P11 é uma versão do P3 em que todas as instruções são codificadas em 2 palavras de 16 bits. O P11 possui uma cache para código totalmente associativa, com política LRU, capacidade de 8 palavras e blocos de 4 palavras. No P11 executa-se o programa seguinte que gera um sinal periódico em escada através de um porto de saída (ligado a um conversor digital-analógico). A amplitude deste sinal é controlável a partir do conteúdo de um porto de entrada.

```

1   MAX   EQU   FFF0h
2   OUT   EQU   FFFEh
3
4           ORIG 0000h
5   Loop: MOV   R1, M[MAX]
6   Down: DEC  R1
7           MOV  M[OUT], R1
8           NOP
9           NOP
10          BR.NZ Down
11          BR   Loop

```

a) Qual é a taxa de sucesso na cache de instruções ao executar este programa? (assuma um valor elevado em M[MAX] e note que cada leitura de uma instrução gera sempre 2 acessos à cache)

A: 25%	B: 50%	C: 60%	D: 70%	E: ≈100%	F: Nenhuma das anteriores.
--------	--------	--------	--------	----------	----------------------------

b) Sugira uma, e uma só, alteração à estrutura da cache, sem alterar a sua capacidade, que melhore o desempenho ao executar este programa.

d) Considere que o tempo de acesso à cache é 80ns, o tempo de acesso à memória primária é 200ns e que um acesso à memória primária só é lançado, se necessário, após a verificação da cache.

Calcule o *speed-up* máximo do sistema com cache relativamente a um sistema sem cache. ($Speed-up = \text{Desempenho do sistema sem cache} / \text{Desempenho do sistema com cache.}$)

A: 0,4	B: 0,75	C: 1	D: 2,5	E: 3	F: Nenhuma das anteriores.
--------	---------	------	--------	------	----------------------------

e) Sabendo que numa aplicação se mediu um *speed-up* de 1,2 do sistema com cache relativamente ao sistema sem cache (isto é, com acesso imediato à memória), calcule a taxa de sucesso na cache.

A: 33%	B: 57%	C: 67%	D: 75%	E: 96%	F: Nenhuma das anteriores.
--------	--------	--------	--------	--------	----------------------------

3. Considere um sistema com memória virtual paginada com páginas de 1k palavras e com o seguinte mapa de memória virtual:

0000h – 7FFFh	Código
8000h – EFFFh	Dados
F000h – FFFFh	Entradas/Saídas e vectores de interrupção

a) Atendendo exclusivamente ao espaço total de endereçamento, indique quantas entradas tem a tabela de páginas.

b) Na realidade a tabela de páginas traduz só os endereços entre 0 e EFFFh já que a zona superior não é traduzida. A memória primária instalada ocupa um bloco contíguo de 8k palavras que termina em FFFFh. (O espaço físico também varia entre 0 e FFFFh.)

Preencha as entradas relevantes da tabela de páginas de modo a ser mapeada na memória primária uma aplicação com código de 0 a 07FFh, e dados e pilhas de 8000h a 83FFh.

Entrada	Conteúdo

III. (1 + 1 + 1,5 + 1,5 + 1 = 6 val.)

1. Numa transmissão série síncrona orientada ao bit, foi recebida a seguinte sequência

0111111001111100011111010111111001111110

Ignore a parte do cabeçalho e verificação de erros e assumo a *flag* habitual 01111110. Indique os bits de informação útil recebida. (nota: utilize só as posições da tabela que considere necessárias)

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	

2. Indique, justificando, a principal limitação da transmissão série síncrona orientada ao carácter face à orientada ao bit.

3. Desenhe um diagrama temporal que descreva o protocolo *handshake* iniciado pelo receptor. Indique claramente o nome dos sinais envolvidos e a ordem dos acontecimentos.

4. Indique, justificando, a principal vantagem e a principal desvantagem de se usarem caches *write-through* com um sistema DMA.

5. Indique, justificando, a principal vantagem e a principal desvantagem de se usar transferências por bloco com DMA.