

NOME: \_\_\_\_\_

Nº: \_\_\_\_\_

As questões devem ser respondidas nas folhas de enunciado. As questões 1 são de escolha múltipla e uma (e só uma) das respostas está correcta, valendo 0,75 valores. Uma resposta errada desconta 0,25 valores. As questões 2 e 3 valem 2 valores cada. A questão 4 vale 1,5 valores.

**1a** Considere um programa P com  $10^5$  instruções, em que 40% das instruções implicam um acesso à memória. Executado na máquina M, P demora 1ms a executar, exhibe uma *miss rate* de instruções de 4% e um  $CPI_{CPU}$  de 2 ciclos. M tem uma *cache* com linhas de 4 palavras e um acesso à memória central tem uma latência de 60 ns mais 10 ns por palavra. Sabendo que a frequência do relógio de M é de 500 MHz, qual dos seguintes valores corresponde à *miss rate* de dados?

- 2,5%     5%     6%     10%

1: \_\_\_  
2: \_\_\_  
3: \_\_\_  
4: \_\_\_  
Σ: \_\_\_

**1b** Considere a struct em C: `struct {short int a; char b[2];} v[80]`. Qual das seguintes instruções (*assembly* do IA32) implementa a instrução C “`x = v[19].b[1]`”. Admita que `short int` ocupa 2 bytes, `%eah` está associado a `char x`, `%ebx` contém o endereço base de `vector` e `%ecx` contém o valor 19?

- `movb 2(%ebx,%ecx,4), %eah`     `movb 3(%ebx,%ecx,4), %eah`  
 `movb 2(%ebx,%ecx,8), %eah`     `movb 3(%ebx,%ecx,2), %eah`

**1c** O MIPS (Milhões de Instruções Por Segundo) é uma métrica frequentemente publicitada pelos fabricantes para anunciar o desempenho dos respectivos CPUs. Qual das seguintes afirmações é verdadeira?

- O MIPS nativo pode ser usado para avaliar o desempenho de CPUs com diferentes Instruction Sets.  
 O MIPS nativo não pode ser usado para avaliar o desempenho do mesmo programa em dois CPUs com o mesmo Instruction Set, o mesmo CPI e diferentes frequências do relógio.  
 O MIPS de pico corresponde à máxima taxa de execução de instruções que o processador pode atingir, não sendo atingível para programas comuns.  
 O MIPS nativo pode ser usado para avaliar o desempenho de dois programas diferentes no mesmo CPU.

**1d** A convenção seguida no MIPS para uso dos registos `$s` e `$t` na invocação de procedimentos resulta de ...

- diferenças na implementação destes registos que limitam o modo como cada tipo de registo pode ser usado.  
 os nomes dos registos determinarem a forma exclusiva como podem ser usados, nomeadamente, “s” de seguro e “t” de temporário.  
 impossibilidade de salvaguardar o conteúdo dos registos `$t`.  
 tentativa de diminuir os acessos à memória, requeridos pelas operações com a *stack*, estabelecendo quais os registos que são salvaguardados pela função invocada e quais os que o são pela função que invoca.

**2.** Na compilação de uma função, escrita em C, o compilador gera código para criar, para cada invocação uma estrutura designada por *activation record*, constituída por vários campos e que mantém o contexto da função. Descreva esta estrutura e o papel de cada um dos seus campos, tal como é gerada pelo `gcc`. Para facilitar a sua resposta represente-a graficamente juntamente com o apontador que para ela é mantido.


3. Relativamente à hierarquia de memória, discuta os diferentes esquemas de colocação de blocos na cache, indicando as vantagens e desvantagens de cada um deles. Diga o que entende por localidade e indique de que forma esta é responsável pelo sucesso das caches na redução do tempo de acesso à memória.


4. Indique os vários estágios de execução da instrução `addi $t2, $t3, -12`, explicitando os elementos do datapath single cycle do MIPS utilizados (ver figura anexa), as operações realizadas em cada fase, e indique o valor dos sinais de controlo `RegDst`, `RegWrite`, `ALUSrc`, `PCSrc`, `MemWrite`, `MemRead` e `MemToReg`.


**PRÁTICA**

A questão 5 vale 4 valores, a questão 6 vale 2,5 valores e a questão 7 vale 2 valores.

- 5.** Considere o bloco de programa, escrito em linguagem C, que calcula a nota mínima numa lista de alunos.

5a:	___
5b:	___
5c:	___
5d:	___
Σ:	___

<pre>typedef struct{   char nome[22];   int nota; }aluno;  aluno alunos[128];</pre>	<pre>int NotaMinima(aluno * la,int tot){   int i,min;   min=la[0].nota;   for (i=1;i&lt;tot;i++){     if (la[i].nota&lt;min){       min=la[i].nota;     }   }   return min; }</pre>
---	---

- (a) Complete o programa em baixo, nas linhas marcadas de 1 a 6, de modo a corresponder a uma compilação válida do programa em C.

<pre>NotaMinima:   pushl %ebp   movl %esp, %ebp   subl \$8, %esp   movl 8(%ebp), %eax   movl 24(%eax), %eax   movl %eax, -8(%ebp)   /*1*/ movl \$1, _____   .L6:     movl -4(%ebp), %eax   /*2*/ cmpl _____, %eax   /*3*/ jge _____     movl -4(%ebp), %edx     movl %edx, %eax     sall \$3, %eax     subl %edx, %eax     leal 0(,%eax,4), %edx     movl 8(%ebp), %eax     movl 24(%edx,%eax), %eax</pre>	<pre>    cmpl -8(%ebp), %eax     jge .L8      movl -4(%ebp), %edx     movl %edx, %eax     sall \$3, %eax     subl %edx, %eax     leal 0(,%eax,4), %edx     movl 8(%ebp), %eax     movl 24(%edx,%eax), %eax   /*4*/ movl %eax, _____   .L7:   /*5*/ movl _____, %eax     leave     ret   .L8:     leal -4(%ebp), %eax   /*6*/ incl _____     jmp .L6</pre>
--	---

- (b) Identifique e explique as instruções responsáveis pela seguinte linha no programa em c:  
if (la[i].nota<min)


(c) Identifique e explique as instruções responsáveis pela seguinte linha no programa em c:

```
min=la[i].nota;
```


(d) Que alterações seriam necessárias fazer, no programa assembly, caso o array nome da estrutura aluno passasse a ter 28 caracteres.


NOME: \_\_\_\_\_

Nº: \_\_\_\_\_

6. Considere o seguinte fragmento de código, escrito em assembly do MIPS:

\$L2:

```
addu    $t0, $t1, $t2
lb      $s2, 0($t0)
slt     $v0, $s2, 50
bne     $v0, $0, $L2    # QB
```

\$L3:

```
la      $s1, 0xC00AA00
sb      $a0, 28($fp)
move    $sp, $a0
```

6a:	___
6b:	___
6c:	___
7a:	___
7b:	___
Σ:	___

a) Reescreva o código fornecido, usando apenas instruções nativas do MIPS.

b) Converta a instrução assinalada com "# QB" para código máquina do MIPS, apresentando o resultado final em hexadecimal e incluindo na resposta todos os passos intermédios usados.

c) Converta para assembly do MIPS a instrução máquina **0x 0224082B** (em hexadecimal), apresentando os cálculos intermédios.

7. Considere um programa P1 escrito em assembly do MIPS com as características apresentadas na tabela seguinte, executado numa máquina M1 com uma *miss rate* de instruções de **6%**, uma *miss rate* de dados de **12%** e uma frequência do relógio de **1.333 GHz**.

Tipo de Instruções	Nº Instruções	CPI <sub>CPU</sub>
LW / SW	$1 \cdot 10^6$	4.0
ADD / SUB / SLT	$4 \cdot 10^6$	1.5
LB / SB	$2 \cdot 10^6$	3.5
BEQ / BNE	$3 \cdot 10^6$	2.0

- a) Qual a *miss penalty* (em ns) apresentada pela máquina M1, se o tempo de execução deste programa for **46.06ms** ?

- b) Considere agora uma máquina M2, em tudo idêntica a M1, excepto que utiliza um novo processador MIPS com frequência do relógio de **1.5GHz** e um CPI<sub>CPU</sub> **20% pior** que o do processador da máquina M1. Com base no desempenho que se obtém ao executar o programa P1, valerá a pena fazer a actualização de M1 para M2? Justifique a resposta.