T1:_____
T2:_____
T3:_____
T4:_____
T5:_____
T6:_____
T7:_____
T: _____
P:_____

Arquitectura de Computadores I - 2002/03 Exame 1ª chamada – 13.Jan.03

NOME: _____ **Nº:**_____

TEÓRICA

As questões devem ser respondidas na própria folha do enunciado. As questões 1 a 4 são de escolha múltipla, e apenas uma das respostas está correcta, valendo 1 valor. Uma resposta errada desconta 1/3 de valor. As questões 5 e 6 valem 2 valores cada. A questão 7 (4 valores) só deve ser respondida pelos alunos que, justificadamente, não fizeram a componente teórico-prática.

1. A estrutura da grande maioria dos computadores modernos apresenta uma hierarquização dos barramentos. Esta surge porque:

☐ Os controladores de periféricos mais lentos não podem estar ligados a barramentos rápidos, pois não são capazes de acompanhar a sua velocidade de funcionamento.

☐ A utilização de barramentos mais lentos permite reduzir os custos da máquina.

☐ A existência de diferentes barramentos, com diferentes velocidades, mas interligados entre si, isola o tráfego entre diferentes componentes do sistema diminuindo contenções no acesso aos barramentos, permitindo simultaneamente reduzir o comprimento dos barramentos mais rápidos.

☐ Diferentes controladores são desenhados especificamente para diferentes barramentos; a máquina deve ter vários barramentos diferentes para permitir a ligação de uma grande diversidade de controladores.

2. O MIPS (Milhões de Instruções Por Segundo) é uma métrica frequentemente publicitada pelos fabricantes de máquinas 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 pode ser usado para comparar o desempenho do mesmo programa em dois CPUs com o mesmo *Instruction Set* e diferentes frequências do relógio.

☐ O MIPS de pico é uma medida confiável do máximo desempenho que os programas do utilizador atingem numa máquina.

☐ O MIPS nativo pode ser usado para avaliar o desempenho de dois programas diferentes no mesmo CPU.

3. Considere o *datapath single cycle* do MIPS. Podemos afirmar que:

☐ Esta solução é promissora do ponto de vista do desempenho, pois todas as instruções levam apenas um ciclo a completar (CPI=1).

☐ Um dos grandes inconvenientes desta abordagem é que o período do relógio (T_{cc}) deve ser tão longo como a instrução mais demorada.

☐ Esta solução obriga ao desenho de um *control path* e respectiva unidade de controlo extremamente complexos.

☐ Esta solução permite uma grande economia de componentes, pois cada um pode ser utilizado várias vezes durante o mesmo ciclo do relógio, isto é, durante a execução da mesma instrução.

4. Considere uma máquina com um espaço de endereçamento de 32 *bits*, uma cache de 256 Kbytes, linhas de 16 *palavras*, palavras de 4 bytes e mapeamento directo. Qual a distribuição dos *bits* do endereço para seleccionar o byte correcto na cache:

☐ *Tag* = 14; Índice = 11; *Block Offset* = 4; *Byte Offset* = 3

☐ *Tag* = 14; Índice = 13; *Block Offset* = 3; *Byte Offset* = 2

☐ *Tag* = 14; Índice = 12; *Block Offset* = 4; *Byte Offset* = 2

☐ *Tag* = 15; Índice = 11; *Block Offset* = 4; *Byte Offset* = 2

NOME: _____ Nº:_____

5. Diga o que entende por localidade nos acessos à memória e discuta de que forma esta propriedade é responsável pelo sucesso da utilização de *caches* na redução do tempo de acesso à memória.

NOME: _____ **Nº:**_____

6. Comente a seguinte afirmação de um engenheiro responsável pela concepção do conjunto de instruções de uma nova família de processadores, indicando se é verdadeira ou não: "Devemos conceber um conjunto de instruções tão simples e reduzido quanto possível, permitindo o desenho de processadores com um CPI médio extremamente baixo, pois assim conseguiremos reduzir o tempo de execução de qualquer programa."

NOME: _____ **Nº:**_____

Apenas para os alunos que, justificadamente, não fizeram a componente teórico-prática.

7. Os conjuntos de instruções de processadores com arquitecturas RISC diferem daqueles apresentados por arquitecturas CISC em vários aspectos. Indique quais as principais diferenças, quais as razões que justificam estas 2 abordagens diametralmente opostas e de que forma estas diferenças se reflectem no código *assembly* gerado pelos compiladores. Use como exemplos o *assembly* do IA32 e do MIPS.

NOME: _____ **Nº:**_____

PRÁTICA

As questões práticas devem ser respondidas em folha separada. A questão 1 vale 4 valores, as questões 2 e 3 valem 2 valores cada.

1. Considere que o código assembly IA32 (sem optimização) obtido após compilar a função da tabela seguinte é o que se apresenta na coluna direita da mesma tabela.

a) Identifique a funcionalidade dos blocos **1 a 6** de instruções etiquetados.

b) Sugira uma optimização no bloco **3** e outra no bloco **7**.

```
typedef struct {
   int  num;
   char v;
   } dType;

char somar (
   dType *lst, int min,int max )
   {
   int  med;
   char r;
   if (min != max)
      {
      med = (max-min) >> 1;
      r = somar(lst, min, med)+
          somar(lst, med+1, max);
      }
   else
      r = lst[min].v;
   return (r);
   }
```

```
somar:
   pushl %ebp
   movl  %esp, %ebp
   pushl %ebx                 # B 1
   subl  $8, %esp             # B 1

   movl  12(%ebp), %eax       # B 2
   cmpl  16(%ebp), %eax       # B 2
   je    .L3                  # B 2

   movl  12(%ebp), %edx       # B 3
   movl  16(%ebp), %eax       # B 3
   subl  %edx, %eax           # B 3
   sarl  $1, %eax             # B 3
   movl  %eax, -8(%ebp)       # B 3
   subl  $4, %esp
   pushl -8(%ebp)
   pushl 12(%ebp)
   pushl 8(%ebp)
   call  somar
   addl  $16, %esp            # B 7
   movb  %al, %bl             # B 7
   subl  $4, %esp             # B 7

   pushl 16(%ebp)             # B 4
   movl  -8(%ebp), %eax       # B 4
   incl  %eax                 # B 4
   pushl %eax                 # B 4
   pushl 8(%ebp)              # B 4
   call  somar                # B 4
   addl  $16, %esp
   movb  %al, %al
   leal  (%eax,%ebx), %eax    # B 5
   movb  %al, -9(%ebp)        # B 5
   jmp   .L4                  # B 5
.L3:
   movl  12(%ebp), %eax       # B 6
   imull $8, %eax, %edx       # B 6
   movl  8(%ebp), %eax        # B 6
   movb  4(%eax,%edx), %al    # B 6
   movb  %al, -9(%ebp)        # B 6
.L4:
   movsbl -9(%ebp),%eax
   movl   %eax, %eax
   movl   -4(%ebp), %ebx
   leave
   ret
```

NOME: _____ Nº:_____

c) Escreva em *assembly* do MIPS o código correspondente à seguinte linha de código

r = lst[min].v;

Assuma que:

- o registo **$s0** contém a variável **r** (tipo char)
- o registo **$t1** contém o apontador para a variável estruturada **lst**
- o registo **$t2** contém a variável inteira **min**

2. Considere o seguinte programa, escrito em assembly do MIPS:

```
      li $t0, 0x40000
ciclo:
      subi $t0, $t0, 1
      bne $t0, $ao, ciclo
```

a) Identifique as pseudo-instruções e reescreva o código usando apenas instruções nativas.

b) Converta o *assembly* para o nível máquina, apresentando os resultados finais em hexadecimal. Apresente também todos os passos intermédios.

3. A máquina M, com uma frequência de 1 GHz, apresenta um CPI global de 5 ciclos/instrução ao executar o programa P, cuja distribuição de tipos de instrução se apresenta na tabela. Para este programa a máquina apresenta uma *miss rate* de instruções de 5% e uma *miss rate* de dados de 10%. O CPI_{CPU} é de 1.5 ciclos/instrução.

Tipo Instrução	Num.
Inteiros	$3*10^7$
Vírg. Flutuante	$2*10^7$
Acesso à memória	$5*10^7$

a) Calcule a *miss penalty* exibida por esta máquina. Apresente o resultado em nanosegundos.

b) Suponha que a frequência da máquina M aumenta para 1,333 GHz. Qual deverá ser a *miss penalty* (em nanosegundos) desta versão de M para que o CPI global se mantenha em 5 ciclos/instrução, sabendo que todos os outros parâmetros da máquina e do programa se mantêm?

c) Porque é que um aumento na frequência do relógio pode implicar um aumento do CPI e que consequências é que este facto tem no tempo de execução dos programas?