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

Arquitectura de Computadores I — Exame Tipo – 2002/03

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. Qual das seguintes afirmações, que comparam os vários mecanismos de transferência de dados numa operação de Input/Output (I/O), é verdadeira?

☐ Usando *polling* o CPU é responsável por verificar activamente alterações no estado do controlador de I/O e pela transferência de dados.

☐ A vantagem da utilização do mecanismo de interrupções relativamente ao *polling* é que o primeiro liberta o CPU da transferência de dados.

☐ A utilização do mecanismo de interrupções implica a utilização do CPU para verificar activamente alterações no estado dos controladores de I/O.

☐ O DMA liberta o CPU de qualquer intervenção em todas as operações de I/O.

2. Qual das seguintes afirmações, respeitantes à contribuição do compilador (gcc) para gerar código optimizado (mais rápido), é verdadeira:

☐ O compilador tenta utilizar instruções que minimizem o período do relógio (T_{cc}), minimizando assim o tempo de execução.

☐ O compilador minimiza o número de instruções (#I) a executar.

☐ O compilador minimiza o número de instruções a executar (#I) e o número de acessos à memória, este último para diminuir o CPI médio.

☐ O compilador empenha-se apenas na diminuição do número de saltos condicionais, conseguindo assim maximizar a *hit rate*, pois o programa optimizado exibe maior localidade espacial.

3. O desempenho do CPU pode ser medido usando a fórmula $T_{exec} = \#I * CPI * T_{cc}$. Qual das seguintes afirmações é verdadeira:

☐ O tempo de execução de qualquer programa diminui na mesma proporção em que aumenta a frequência do relógio do processador.

☐ Um algoritmo com menor número de instruções (#I) do que outro funcionalmente equivalente, resulta invariavelmente num programa mais rápido.

☐ Para o mesmo processador obtem-se um programa mais rápido usando um algoritmo e um compilador que reduza o número de instruções e que minimize o CPI.

☐ O desenho de processadores com instruções mais complexas permite reduzir o número de instruções geradas pelo compilador e, consequentemente, obter tempos de execução menores.

4. Qual das seguintes afirmações é verdadeira:

☐ Os processadores com um desenho CISC conseguem maior desempenho que as alternativas RISC devido ao menor número de instruções dos respectivos programas, conseguidos usando instruções mais complexas.

☐ Os processadores com arquitectura RISC apostam na simplicidade do conjunto de instruções e na regularidade do formato das mesmas, para permitir o desenho de Unidades de Controlo mais simples que as alternativas CISC.

☐ Os modos de endereçamento mais complexos, existentes nos processadores CISC (e não nos RISC), permitem obter melhores tempos de execução, devido à redução do número de instruções necessárias para calcular endereços.

☐ Os processadores CISC, com uma arquitectura mais complexa que os RISC, têm também mais registos, permitindo diminuir o número de acessos à memória.

NOME: _____ Nº:_____

5. Indique os vários estágios de execução da instrução `lw $s0, $t1(100)`, explicitando os elementos utilizados e as operações realizadas em cada fase (ver figura anexa), e indique o valor dos sinais de controlo `RegDst, RegWrite, ALUSrc, PCSrc, MemWrite, MemRead e MemToReg`.

NOME: _____ Nº:_____

6. Explique os vários passos que têm que ser dados na invocação e retorno de funções e indique de que forma estes são codificados seguindo a convenção adoptada para o IA32.

NOME: _____ **Nº:**_____

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

7. Compare os conjuntos de instruções do IA32 (CISC) e MIPS (RISC), no que respeita a **complexidade dos modos de endereçamento**, **número de instruções**, **formato de instruções** e **número de registos**, analisando o impacto destas características no tempo de execução, nomeadamente nas seguintes métricas: número de instruções, CPI e frequência do relógio.

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 1 é o que se apresenta na Tabela 2.

a) Identifique a funcionalidade dos blocos **1 a 4** de instruções etiquetados.

b) Sugira duas optimizações no bloco **3** e uma correcção em cada um dos blocos **2 e 5**.

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

```
list[i].eq = 1;
```

Assuma que:

- o registo **$s0** contém a variável inteira **i**
- o registo **$t0** contém o apontador para a variável inteira **list**

```
typedef struct {
  char eq;
  int db;
  } dBase;

int calcEq (dBase *list, int n)
{ int i=1;

  while (i<n)
  {
    if (list[i].db==list[i-1].db)
      list[i].eq = 1;
    else
      list[i].eq = 0;
    ++i;
  }
}
```

```
calcEq:
  pushl %ebp
  movl %esp, %ebp
  pushl %ebx
  subl $4, %esp              # B 1
  movl $1, -8(%ebp)          # B 1
.L3:
  movl -8(%ebp), %eax        # B 2
  cmpl 12(%ebp), %eax        # B 2
  jg .L5                     # B 2
  jmp .L4                    # B 2
.L5:
  movl -8(%ebp), %eax        # B 3
  imull $8, %eax, %ebx       # B 3
  movl 8(%ebp), %ecx         # B 3
  movl -8(%ebp), %eax        # B 3
  movl %eax, %eax            # B 3
  sall $3, %eax              # B 3
  addl 8(%ebp), %eax         # B 3
  leal -8(%eax), %edx        # B 3
  movl 4(%ecx,%ebx), %eax    # B 3
  cmpl 4(%edx), %eax         # B 3
  jne .L6                    # B 3
  movl -8(%ebp), %eax
  imull $8, %eax, %edx
  movl 8(%ebp), %eax
  movb $1, (%eax,%edx)
  jmp .L7
.L6:
  movl -8(%ebp), %eax        # B 4
  imull $8, %eax, %edx       # B 4
  movl 8(%ebp), %eax         # B 4
  movb $0, (%eax,%edx)       # B 4
.L7:
  leal -8(%ebp), %eax
  incl (%eax)
  jmp .L3
.L4:                         # B 5
  leave                      # B 5
  ret                        # B 5
```

NOME: _____ Nº:_____

2. Considere um programa escrito em assembly do MIPS com as características apresentadas na tabela, executado numa máquina sem *cache misses*, com uma frequência do relógio de 100 MHz.

Instrução	Nº instruções	CPI
Lw	900	4
Sw	900	4
add/addi	100	3
slt/slti	100	2

a) Qual o CPI global e o tempo de execução deste programa?

b) Se a *miss rate* de acesso às instruções for 5% e de acesso aos dados 12%, com uma *miss penalty* de 100ns, qual o CPI global e o tempo de execução do programa?

c) O processador desta máquina foi substituído por outro com uma frequência 2 vezes superior, mantendo-se constantes todos os outros parâmetros do sistema. Qual o CPI global e o tempo de execução do programa?

3. Considere uma máquina com um espaço de endereçamento de 32 *bits*, uma cache com uma capacidade de 64 Kbytes de dados, palavras de 32 *bits*, blocos de 16 palavras e uma organização *8-way set associative*.

a) Qual a distribuição dos *bits* do endereço pela *tag*, índice, *block offset* e *byte offset* para seleccionar o *byte* correcto na *cache*?

b) Qual a capacidade total da *cache,* contando com os campos de controlo (*tag* e *valid bit*)?