



Nome: _____ Matrícula: _____

Prova 1

1) (4.0) Hoje em dia, a linha que separa o software do hardware é extremamente tênue. Considerando que a instrução `div` do MIPS não tivesse sido implementada em hardware.

a)(2.0) Escreva a implementação de uma pseudo-instrução que a realize:

`div $t0, $t1, $t2, $t3` $\# \$t0 = \$t2/\$t3$ $\$t1 = \$t2 \% \$t3$ Obs.: Considere $\$t2$ e $\$t3 > 0$

b)(1.0) Em um microprocessador uniclo (cada instrução é executada em 1 ciclo) de 200MHz de frequência de clock, qual o tempo estimado para a execução do trecho de código ao lado:

c)(1.0) Considerando que uma implementação em hardware necessite 33 ciclos de clock para realizar a operação `div` de dois números quaisquer. Qual o fator de desempenho obtido na execução do trecho ao lado entre as duas implementações?

```
...  
li $t2, 255  
li $t3, 19  
div $t0, $t1, $t2, $t3  
div $t4, $t5, $t2, $t1  
...
```

2) (6.0) Dado o mapa da memória de programa abaixo. Realize o trabalho de engenharia reversa de forma a:

a) (3.0) Traduzir o código em linguagem de máquina para Assembly MIPS;

b) (2.0) Traduzir o código Assembly para uma linguagem de alto nível (C ou Java);

c) (1.0) Entenda o algoritmo e responda: O que representa o valor apresentado na tela?

Endereço	Conteúdo	Assembly MIPS
0x00400000	001000 00000 00010 0000000000000101 0,1	<code>addi \$v0, \$zero, 5</code>
0x00400004	000000 00000 00000 00000 00000 001100 0,1	<code>syscall</code>
0x00400008	000000 00000 00010 00100 00000 100001 0,1	<code>addu \$a0, \$zero, \$v0</code>
0x0040000c	000011 00000 100000000000000001001 0,3	<code>jal 100009h</code>
0x00400010	000000 00000 00010 00100 00000 100001 0,1	<code>addu \$a0, \$zero, \$v0</code>
0x00400014	001000 00000 00010 0000000000000001 0,1	<code>addi \$v0, \$zero, 1</code>
0x00400018	000000 00000 00000 00000 00000 001100 0,1	<code>syscall</code>
0x0040001c	001000 00000 00010 0000000000001010 0,1	<code>addi \$v0, \$zero, 10</code>
0x00400020	000000 00000 00000 00000 00000 001100 0,1	<code>syscall</code>
0x00400024	000000 00000 00000 01000 00000 100001 0,1	<code>addu \$t0, \$zero, \$zero</code>
0x00400028	001000 00000 01100 0000000000000001 0,1	<code>addi \$t4, \$zero, 1</code>
0x0040002c	000000 00100 01100 01001 00000 100010 0,1	<code>sub \$t1, \$a0, \$t4</code>
0x00400030	000100 01001 01100 0000000000000110 0,3	<code>beq \$t1, \$t4, 6</code>
0x00400034	000000 00100 01001 00000 00000 011010 0,1	<code>div \$a0, \$t1</code>
0x00400038	000000 00000 00000 01010 00000 010000 0,1	<code>mfhi \$t2</code>
0x0040003c	000101 01010 00000 0000000000000001 0,3	<code>bne \$t2, \$zero, 1</code>
0x00400040	000000 01000 01100 01000 00000 100000 0,1	<code>add \$t0, \$t0, \$t4</code>
0x00400044	000000 01001 01100 01001 00000 100010 0,1	<code>sub \$t1, \$t1, \$t4</code>
0x00400048	000010 00000 100000000000000001100 0,3	<code>j 10000Ch</code>
0x0040004c	000000 00000 01000 00010 00000 100001 0,1	<code>addu \$v0, \$zero, \$t0</code>
0x00400050	000000 11111 00000 00000 00000 001000 0,2	<code>jr \$ra</code>

3) (1.0) Escreva a equação que define o fator de desempenho de um sistema computacional em relação a outro, explicando seus componentes.

BOA SORTE!

OAC - TURMA A

2008/1

1ª PROVA

GABARITO

1) DIV \$t0, \$t1, \$t2, \$t3

a) CONSIDERANDO O ALGORITMO: $\text{int DIV}(\text{int } A, \text{int } B)$

{ int D=0; R=A;

add \$t0, \$zero, \$zero

add \$t1, \$t2, \$zero

loop: SLT \$at, \$t1, \$t3 R < B

bne \$at, \$zero, Fim

sub \$t1, \$t1, \$t3

addi \$t0, \$t0, 1

J LOOP

Fim:

add \$t1, \$t1, \$t3
M^o Ciclos = 4 + 5x (int \$t2/\$t3)

while (R >= B)

{

R = R - B;

D = D + 1;

}

return D, R;

b) addi \$t2, \$zero, 255

→ 1 ciclo

addi \$t3, \$zero, 19

→ 1 ciclo, $255/19$

div \$t0, \$t1, \$t2, \$t3

→ $4 + 5 \times 13 = 69$

div \$t4, \$t5, \$t2, \$t1

→ $4 + 5 \times \frac{255}{8} = 159$

$\frac{255}{8}$

TOTAL: 230 CICLOS

Logo: $t_1 = \text{M}^o \text{ Ciclos} \times T$

$t_1 = 230 \times \frac{1}{200M} = 1,15 \mu s$

c) M^o TOTAL = 1 + 1 + 33 + 33 = 68 CICLOS

Logo: $t_2 = 68 \times \frac{1}{200M} = 340 ns$

$\eta = \frac{t_1}{t_2} = 3,38$

Logo 203,38 vezes mais rápido que 1

2) 9) na FUTA

b) main()

```
{ int n;
```

```
scanf("%d", &n);
```

```
printf("%d", n-DIV(h));
```

```
exit(1);
```

```
}
```

```
int n-DIV (int n)
```

```
{ int i, j, k, M;
```

```
int i, j, k, M;
```

```
i=0;
```

```
M=1;
```

```
j=n-M;
```

```
for (j=n-M; j!=M; j=j-M)
```

```
{ k=n % j;
```

```
if (k==0)
```

```
i=i+M;
```

```
}
```

```
return i;
```

```
}
```

c) mostra o número de divisores (hauos que 1) que um n° lido no teclado possui.

3) $\eta = \frac{t_1}{t_2}$ → tempo de execução na máquina 1

t_2 → " " " " " 2

2 é η vezes mais rápido que 1

$T = N \cdot \text{instruções} \times \text{CPI} \times T$

└

└ N° de instruções

└ período de clock

└ ciclos por instrução