



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 mult do MIPS não tivesse sido implementada em hardware.

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

mult \$t0, \$t1, \$t2 # \$t0 = \$t1*\$t2 Obs.: \$t1 e \$t2 >= 0 e despreze o overflow

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:

```
...  
li $t1, 255  
li $t2, 63  
mult $t0, $t1, $t2  
mult $t3, $t0, $t1  
...
```

c)(1.0) Considerando que uma implementação em hardware necessite 32 ciclos de clock para realizar a multiplicação de dois números quaisquer. Qual o fator de desempenho obtido na execução do trecho ao lado entre as duas implementações?

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 00000000000000101 0,1	addi \$v0, \$ZERO, 5
0x00400004	000000 00000 00000 00000 00000 001100 0,1	syscall
0x00400008	000000 00000 00010 00100 00000 100001 0,1	addu \$a0, \$ZERO, \$v0
0x0040000c	000011 0000010000000000000000001001 0,3	jal 100009h
0x00400010	000000 00000 00010 00100 00000 100001 0,1	addu \$a0, \$ZERO, \$v0
0x00400014	001000 00000 00010 0000000000000000001 0,1	addi \$v0, \$ZERO, 1
0x00400018	000000 00000 00000 00000 00000 001100 0,1	syscall
0x0040001c	001000 00000 00010 000000000000001010 0,1	addi \$v0, \$ZERO, 10
0x00400020	000000 00000 00000 00000 00000 001100 0,1	syscall
0x00400024	000000 00000 00000 01000 00000 100001 0,1	addu \$t0, \$ZERO, \$ZERO
0x00400028	001000 00000 01100 0000000000000000001 0,1	addi \$t4, \$ZERO, 1
0x0040002c	000000 00100 01100 01001 00000 100010 0,1	sub \$t1, \$a0, \$t4
0x00400030	000100 01001 01100 00000000000000110 0,3	beq \$t1, \$t4, 6
0x00400034	000000 00100 01001 00000 00000 011010 0,1	div \$a0, \$t1
0x00400038	000000 00000 00000 01010 00000 010000 0,1	mtHi \$t2
0x0040003c	000101 01010 00000 0000000000000000001 0,3	bne \$t2, \$ZERO, 1
0x00400040	000000 01000 01100 01000 00000 100000 0,1	add \$t0, \$t0, \$t4
0x00400044	000000 01001 01100 01001 00000 100010 0,1	sub \$t1, \$t1, \$t4
0x00400048	000010 0000010000000000000000001100 0,3	j 10000Ch
0x0040004c	000000 01000 01100 00010 00000 101010 0,1	slt \$v0, \$t0, \$t4
0x00400050	000000 11111 00000 00000 00000 001000 0,2	jr \$ra

3) (1.0) Escreva a equação que define o desempenho de um sistema computacional, explicando o significado de seus fatores.

1ª PROVA
CABARITO1) `MULT $t0, $x1, $x2`

a) considerando o Algoritmo:

```

add $t0, $zero, $zero
add $at, $zero, $zero
LOOP: BEQ $x1, $at, Fim
      add $t0, $t0, $x2
      addi $at, $at, 1
      j LOOP

```

Fim;

`int mult (int A, int B)``{ int i, C = 0;``for (i = 0; i < A; i++)` `C = C + B;``return C; }`nº ciclos: $3 + 4 \times \$x1$ Obs.: pode ser + eficiente
verificar qual é >b) `addi $t1, $zero, 255`

→ 1 ciclo

`addi $t2, $zero, 63`

→ 1 ciclo

`MULT $t0, $t1, $t2`→ $3 + 4 \times 255 = 1023$ `MULT $t3, $t0, $t1`→ $3 + 4 \times 1023 = 4095$

TOTAL: 65288 ciclos

Logo: $t_1 = \text{nº ciclos} \times T$

$$t_1 = 65288 \times \frac{1}{200M} = 326,44 \mu s$$

c) `MULT` de 32 ciclos:

$$\text{nº TOTAL ciclos} = 1 + 1 + 32 + 32 = 66$$

$$\text{Logo } t_2 = 66 \times \frac{1}{200M} = 330 ns$$

$$\eta = \frac{t_1}{t_2} = 973,68 \quad \text{Logo } \eta \text{ é } 973,68 \text{ vezes mais rápido que } 1$$

2) a) na Folha

b) main()

{ int n

scanf("%d", &n);

printf("%d", e_primo(n));

exit();

}

int e_primo(int n)

{ int i, 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<m);

}

c) mostra: 1 → se o número entrado no teclado é primo
0 → se não for primo

3) $\text{desempenho}_x = \frac{1}{\text{tempo}_x}$

tempo_x → tempo de execução na máquina x

$\text{tempo}_x = N \times \text{instruções} \times \text{CPI} \times T$

↳ período de clock
↳ ciclos por instrução
↳ nº de instruções