



Nome: GABARITO

Matrícula: _____

Prova 2

1) (5.0) No MIPS simplificado desenvolvido durante o curso várias instruções bastante úteis estão ausentes. Mantendo a compatibilidade do código em linguagem de máquina ao MIPS original: 1.1) modifique adequadamente o caminho de dados multiciclo, 1.2) redefina os sinais de controle (se necessário), e 1.3) desenhe a máquina de estados necessária, para implementar, além das instruções já existentes (lw, sw, add, sub, and, or, slt, beq e j), as instruções abaixo:

a) (1.0) jr \$rs

b) (2.0) xori \$rt, \$rs, Imm # Imediato de 16 bits

c) (2.0) sllv \$rd, \$rs, \$rt # R[rd]=R[rs]<<R[rt] Considere: OpCode=000000₂ e Funct=000100₂

2) (4.0) Considerando apenas os seguintes tempos de atraso dos blocos operativos de uma CPU MIPS:

Operação Lógica da ULA (and, or): 250ps

Operação Aritmética da ULA (add, sub, slt): 400ps

Leitura do Banco de Registradores: 100ps

Escrita no Banco de Registradores: 200ps

Leitura da memória: 300ps

Escrita na memória: 500ps

Dado o seguinte trecho de programa:

```
...
4 add $t0, $zero, $zero
5 lw $t1, 100($fp)      # $t1=1
5 lw $t2, 104($fp)      # $t2=100
5 lw $t4, 108($fp)      # $t4=4
LOOP: 5 lw $t3, 0($s0)    } HAZARD DADO
      4 add $t3, $t3, $t3 } HAZARD DADO
      4 sw $t3, 0($s0)
      4 add $s0, $s0, $t4
      4 add $t0, $t0, $t1 } HAZARD DADO
      4 slt $t5, $t0, $t2 } HAZARD DADO e CONTROLE
      3 beq $t5, $zero, OUT } HAZARD DADO e CONTROLE
      3 j LOOP           } HAZARD CONTROLE
OUT:  4 add $t0, $zero, $zero
...
```

a) (0.5) Para uma implementação uniciclo do MIPS: Qual a maior frequência de clock utilizável? Qual o tempo de execução?

b) (0.5) Para uma implementação multiciclo do MIPS: Qual a maior frequência de clock utilizável? Qual o tempo de execução?

c) (0.5) Para uma implementação em Pipeline ideal: Qual a maior frequência de clock utilizável? Qual o tempo de execução?

d) (0.5) Identifique no programa acima, todos os hazards existentes, classificando-os em estrutural, dados ou controle.

e) (1.0) Qual o tempo de execução para uma implementação em Pipeline real considerando que todos os hazards foram corrigidos apenas com o uso de bolhas. (considere jump executado em 2 ciclos, previsão do desvio como não-tomado e avaliação no 2º ciclo)

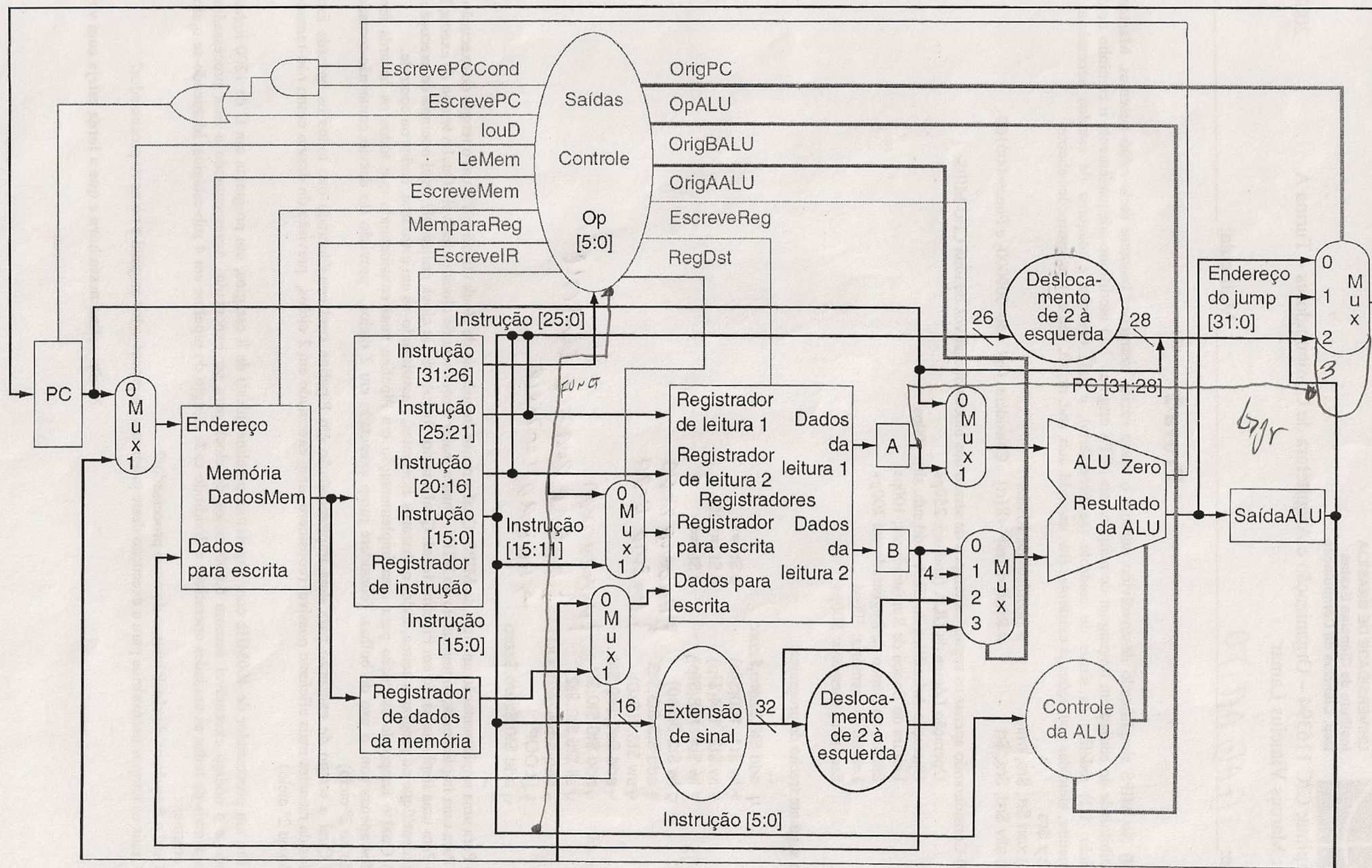
f) (1.0) Qual o tempo de execução para uma implementação em Pipeline real considerando que todos os hazards foram corrigidos da maneira mais eficiente possível (considere jump executado em 2 ciclos, previsão do desvio como não-tomado e avaliação no 2º ciclo)

3) (1.5) Em um processador de 400MHz com arquitetura pipeline (ideal) de 8 estágios, um programa em C de 1200 linhas é compilado e o código executável demora 6.000 ciclos de clock para ser executado. Acrescentando a esse processador 3 unidades a mais de todas as unidades operativas e dividindo cada estágio do pipeline em 4 sub-estágios. Mantendo-se o clock externo original:

a) (0.5) Qual a frequência de clock interno do novo processador?

b) (1.0) Quais os tempos necessários para a execução deste programa no processador original e no novo processador?

Muita calma nessa hora e que a força esteja com vc!!!



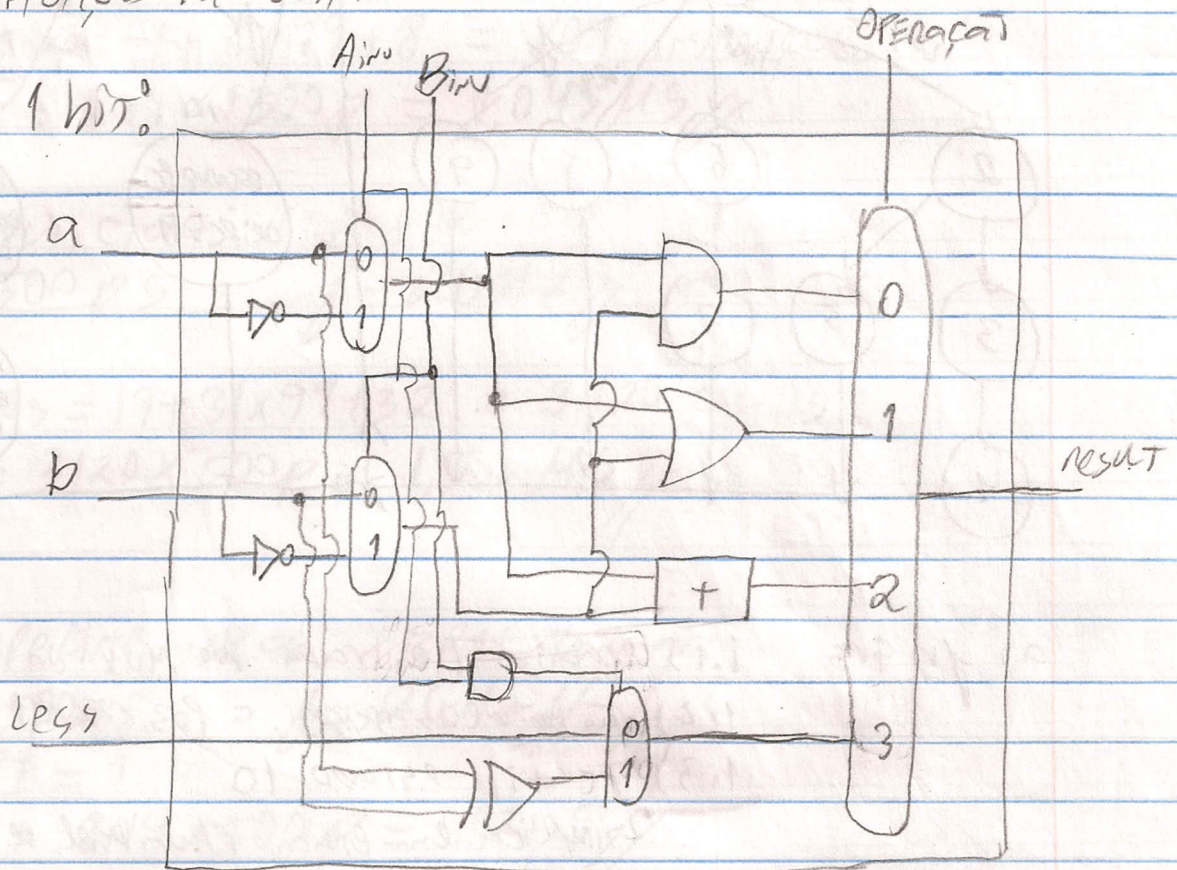
OAC - TURMA - A

2007/2

2: PROVA

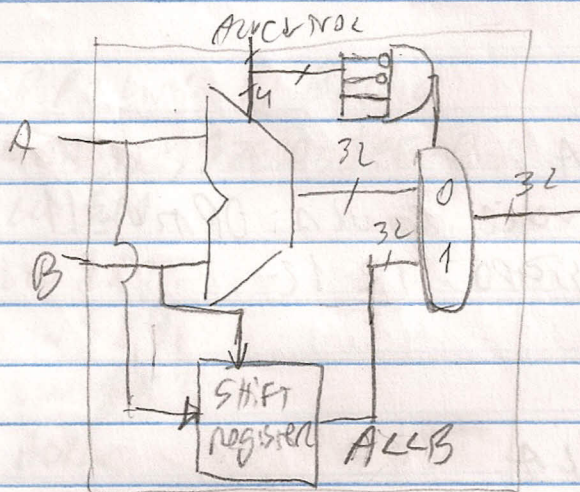
GABARITO

1) MODIFICAÇÕES NO CAMINHO A DADOS: NO VERSO
MODIFICAÇÃO NA ULA:



XOR ALU CONTROL = 1111

SLV ALU CONTROL = 0011



0000 AND

0001 OR

0010 ADD

0110 SUB

0111 SLT

1100 MUX

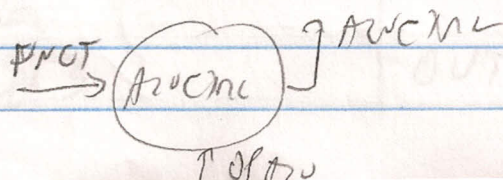
IMPLEMENTATION: XOR

OP ALU = 11 → ALU CONTROL = 1111

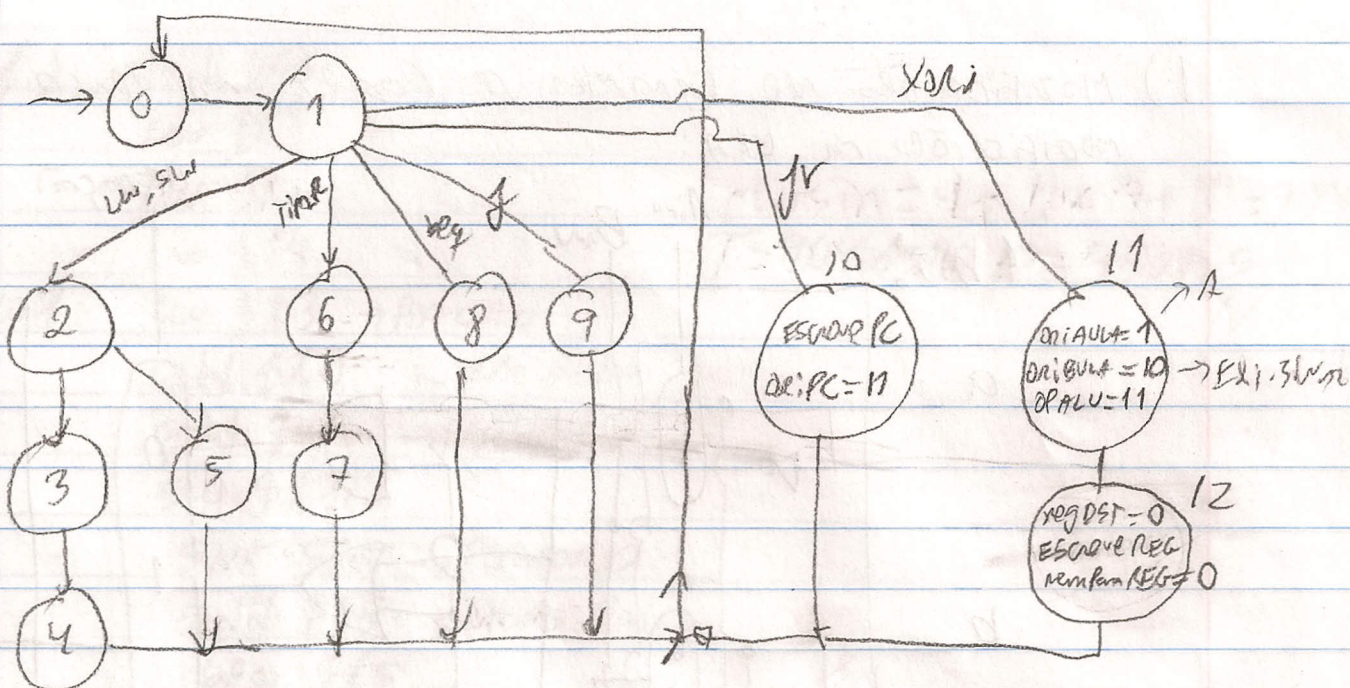
OP ALU = 10 include

EXACT = 000100

→ ALU CONTROL = 0011



MÁQUINAS DE ESTADO



- a) jr & rs
- 1.1) Aumenta 1 entrada no MUX da origem PC
 - 1.2) não há necessidade, a posição 11 do m.a. permanece
 - 1.3) incluir estado 10

Implica em enviar p/ unidade de controle
além do PCOP, também o código Funct
P/ BOP identificar OP: 000000 Funct: 001000

- b) xorl & rt, & rs, im

- 1.1) muda a VLA e o controle
- 1.2) muda controle da ALU: OP ALU = 11
- 1.3) incluir estado 11 e 12

- c) sllv & rd, & rs, & rt

- 1.1) muda AULA
- 1.2) muda controle da ALU p/ Funct: 000100
- 1.3) não muda nada

2) a) UNICICLO: instrução + demorada

$$T = 300 + 100 + 400 + 300 + 200 = 1300 \text{ ps}$$

$$f = \frac{1}{1300 \text{ p}} = 769,2 \text{ MHz}$$

$$\text{CPI} = 1$$

$$\text{Nº instruções} = 4 + 8 \times 99 + 8 = 804 \text{ instruções}$$

$$T = 804 \times 1300 \text{ p} = 1,045 \mu\text{s} //$$

b) MULTICICLO: etapa + demorada

$$T = 500 \text{ ps} \quad f = 20 \text{ MHz} //$$

$$\text{Nº ciclos} = 19 + 31 \times 99 + 32 = 3120 \text{ ciclos}$$

$$T = 3120 \times 500 \text{ p} = 1,56 \mu\text{s} //$$

c) Pipeline ideal: etapa + demorada

$$T = 500 \text{ ps} \quad f = 20 \text{ MHz} //$$

$$\text{CPI} = 1$$

$$T = 804 \times 500 \text{ p} = 402 \text{ ns} //$$

d) na Folha

e) Pipeline c/ Bolha

add \$t0, \$zero, \$zero	add \$s0
lw \$t1	add \$t0
lw \$t2	Bolha
lw \$t4	Bolha
loop: lw \$t3	slt \$t5
Bolha	Bolha
Bolha	Bolha
add \$t3	beq \$t5
Bolha	loop
Bolha	Bolha
sw \$t3	OUT: add \$t0

$$n_{inst} = 4 + 17 \times 99 + 17 = 1704 \text{ instruções}$$

$$T = 1704 \times 500p = 852 \text{ ns}$$

1) add \$t0
lw \$t1

lw \$t2

lw \$t4

loop: lw \$t3 → FORWARD

add \$t0 ← t0 e t3 Both

add \$t3 → FORWARD

sw \$t3

slt \$t5 → FORWARD

and \$t0 ← t0 e t5 Both

beg \$t5

loop

Both

out: add \$t0

$$n_{inst} = 4 + 9 \times 99 + 9 = 904$$

$$T = 904 \times 500p = 452 \text{ ns}$$

3)

a) 4 substituições

$$T = \frac{1}{400M} / 4 = \frac{1}{1,6G} \rightarrow f = 1,6GHz$$

b) origina: $T = 6.000 \times \frac{1}{400M} = 15 \mu s$

Novo: $T = \frac{6.000}{4 \times 4} \times \frac{1}{400M} = 937,5 \text{ ns}$

4 substit → 1+3 drivers

16 ns / ciclo

