



d0 d1 / d2 d3 d4 d5 d6

Nome: _____ Matrícula: /

Prova 2

1) (6.0) Na implementação do processador MIPS uniciclo desenvolvida durante o curso, várias instruções bastante úteis estão ausentes. Mantendo a compatibilidade do código em linguagem de máquina à ISA MIPS:

1.1)(2.5) Modifique adequadamente o caminho de dados do verso desta folha para implementar as seguintes instruções:

- a)(0.5) jal LABEL # PC=LABEL ; R[\$ra]=PC+4
- b)(0.5) jr \$rs # PC=R[\$rs]
- c)(0.5) lui \$rt,IMM # R[\$rt]={IMM,16b'0} ; \$rs=0x00
- d)(0.5) ori \$rt,\$rs,IMM # R[\$rt]=R[\$rs] or {16b'0,IMM}
- e)(0.5) addi \$rt,\$rs,IMM #R[\$rt]=R[\$rs]+{16{IMM[15]},IMM}

1.2)(2.5) Uma das funções do Coprocessador 0 (*System Control Coprocessor*) no MIPS é o tratamento de exceções. No tratamento da exceção simplificado visto em aula, o registrador EPC armazena o endereço da instrução que gerou a exceção e o registrador CAUSE indica a causa da exceção, transferindo o controle para a rotina de tratamento de exceção.

A fim de incorporar estas funcionalidades no processador MIPS uniciclo desenvolvido em aula modifique o caminho de dados no verso desta folha de modo a implementar:

- f)(0.5) exceção overflow # EPC=PC; CAUSE=0x30; PC=0x00001000
- g)(0.5) exceção instrução inválida # EPC=PC; CAUSE=0x28; PC=0x00001000
- h)(0.5) syscall # EPC=PC; CAUSE=0x20; PC=0x00001000 Opcode/Funct=0x00/0x0C
- i)(0.5) mfc0 \$rt,\$rd # \$rt=\$rd (CAUSE=13, EPC=14,...); \$rs=0x00; Opcode/Funct=0x10/0x00
- j)(0.5) eret # PC=EPC+4 Opcode/Funct=0x10/0x18

1.3)(1.0) Especifique os requerimentos da Unidade de Controle, isto é, defina todos os sinais de entrada e respectivos sinais de saída, para a CPU completa (ISA com as instruções antigas vistas em aula e as novas de a) a j)).

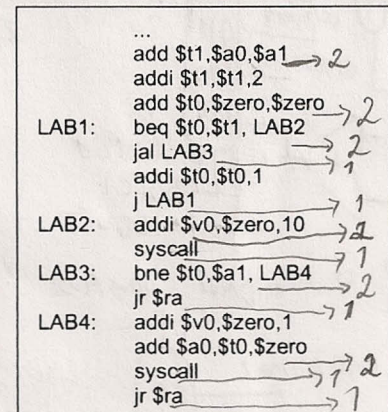
2) (4.0) Considerando apenas os seguintes tempos de atraso das unidades operativas do caminho de dados de uma CPU MIPS:

- Operação com a ULA: 100ps
- Leitura dos Bancos de Registradores: 80ps
- Escrita nos Bancos de Registradores: 100ps
- Leitura da memória: 120ps
- Escrita na memória: 150ps

Com relação ao trecho de programa em assembly MIPS ao lado, onde os registradores **\$a0 = d5** e **\$a1=d6**.

Considere que a rotina de tratamento de exceção print int consista sempre de 25 instruções tipo-R sem nenhum hazard, e que a rotina de exit consista de um branch (não tomado) seguido de jump, sem hazards de dados. O comando syscall requer que os argumentos necessários já estejam disponíveis no banco de registradores.

Responda as questões abaixo especificando como as instruções não implementadas em aula (addi, jal, bne, jr, syscall) foram consideradas em cada caso:



a)(0.5) Para a implementação uniciclo, qual será a maior frequência de clock utilizável? Qual o tempo de execução deste trecho de código neste caso?

b)(0.5) Para a implementação multiciclo vista em aula, qual será a maior frequência de clock utilizável? Qual o tempo de execução deste trecho de código neste caso? Justifique os requerimentos das instruções não implementadas em aula.

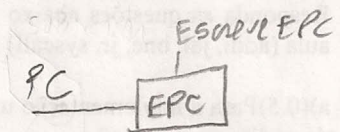
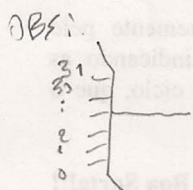
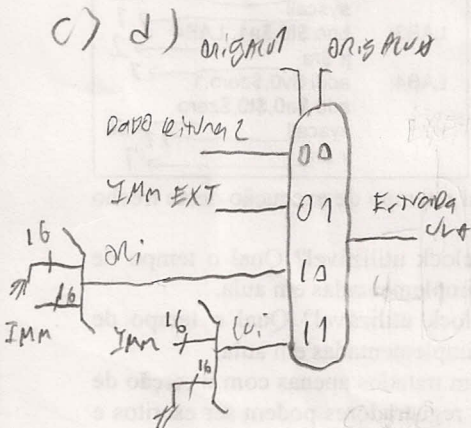
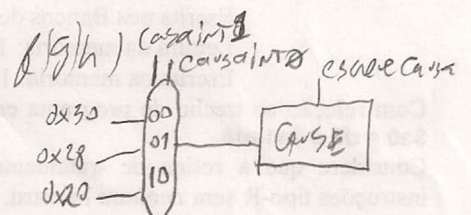
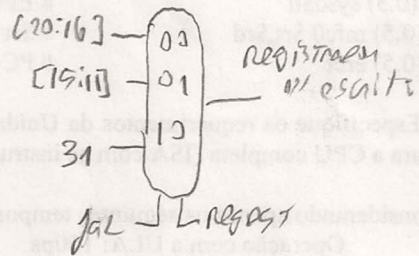
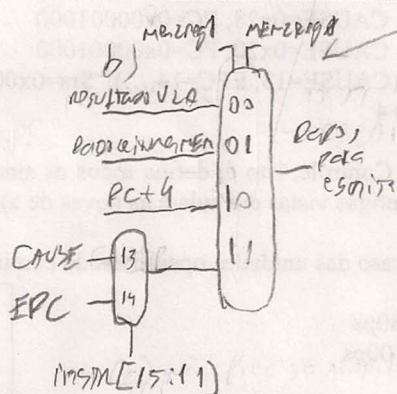
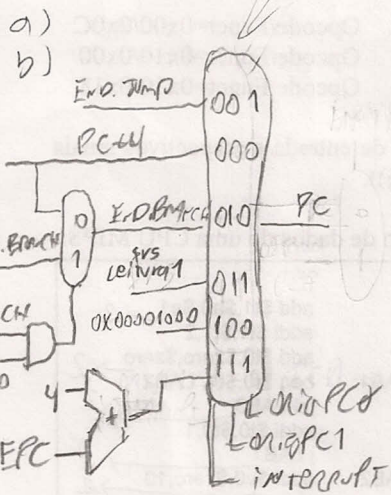
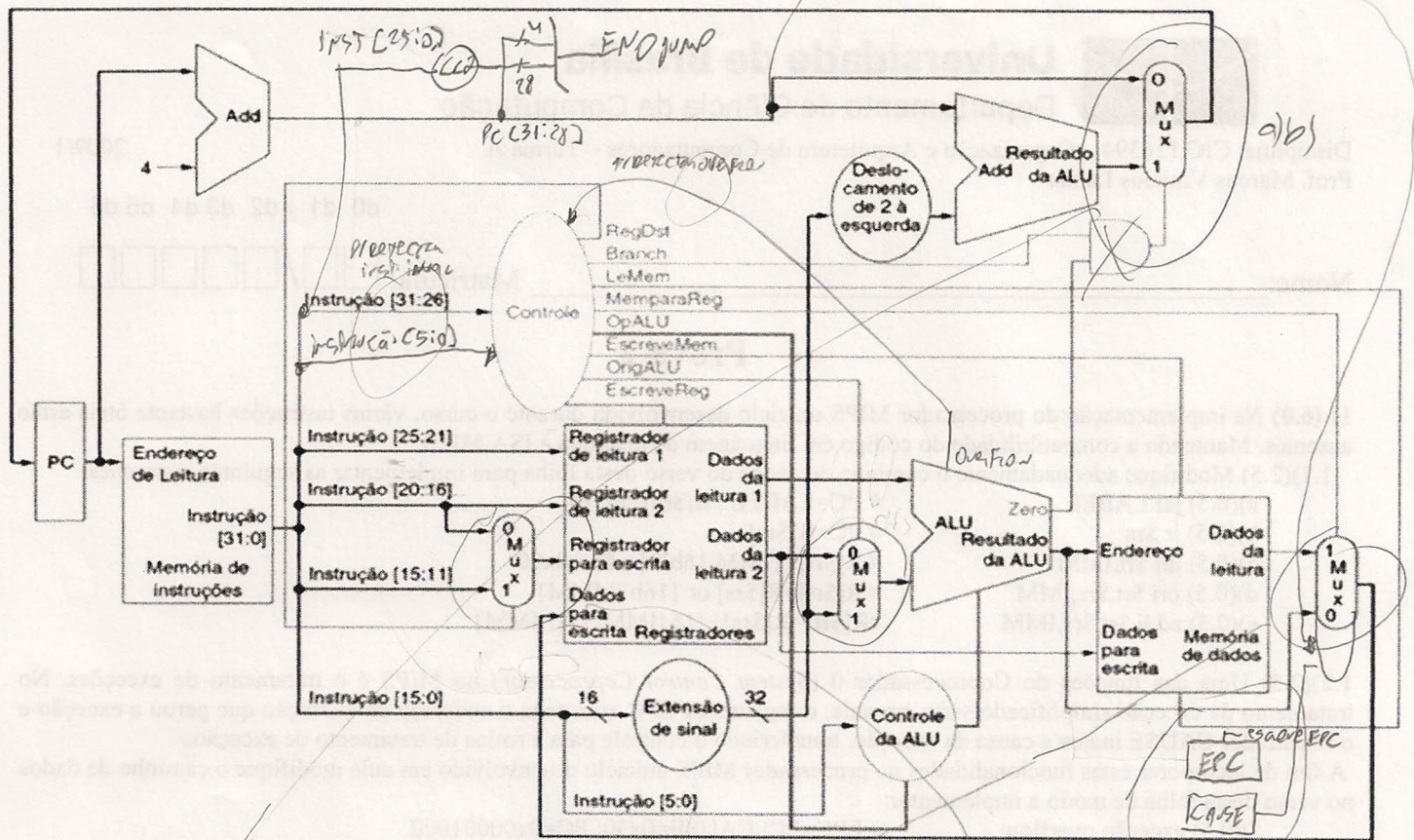
c)(0.5) Para uma implementação com pipeline ideal, qual será a maior frequência de clock utilizável? Qual o tempo de execução deste trecho de código neste caso? Justifique os requerimentos das instruções não implementadas em aula.

d)(1.0) Qual o tempo de execução da implementação com pipeline, se todos os hazards forem tratados apenas com inserção de bolhas? Indique no programa os locais e o número de bolhas necessário. Considere que os registradores podem ser escritos e lidos no mesmo ciclo, que o branch não previsto é avaliado na etapa EX, e que o jump necessita sempre 2 ciclos.

e)(1.0) Qual o tempo de execução da implementação com pipeline, se os hazards forem tratados eficientemente pelo processador com forwarding e/ou bolhas e/ou execução fora de ordem? Desenhe o pipeline esquemático indicando as instruções, bolhas e forwards sugeridos. Considere que os registradores podem ser escritos e lidos no mesmo ciclo, que o branch é avaliado na etapa ID e previsto como não-tomado.

f)(0.5) Ao final da execução do trecho de programa, o que vai estar escrito na tela?

Boa Sorte!!!



OAC - TURMA A

2009/1

2ª PROVA

GABARITO

SINAR	TIPO-R	LW	SW	REG	JAL	IV	LV	ORI	ADDI	OVER	INS INV	SYSGL	MFCO	ERET
RESRST	1	0	X	X	0	X	0	0	0	NA	0	X	0	X
ORIALV1	0	0	0	0	X	X	1	1	0	NA	0	X	X	X
ORIALO0	0	1	1	0	X	X	1	0	1	NA	0	X	X	X
MEM2REG1	0	0	X	X	1	X	0	0	0	NA	0	X	1	X
MEM2REG0	0	1	X	X	0	X	0	0	0	NA	0	X	1	X
ESCRVREG	1	1	0	0	1	0	1	1	1	NA	0	0	1	0
LEMEM	0	1	0	0	0	0	0	0	0	NA	0	0	0	0
ESCRMEM	0	0	1	0	0	0	0	0	0	NA	0	0	0	0
BRANCH	0	0	0	1	0	0	0	0	0	NA	0	0	0	0
ALUOP1	1	0	0	0	X	X	1	1	0	NA	0	X	X	0
ALUOP0	0	0	0	1	X	X	1	1	0	NA	0	X	X	0
JAL	0	0	0	0	1	0	0	0	0	NA	0	0	0	0
INTERRUPT	0	0	0	0	0	0	0	0	0	1	1	1	0	1
ORIGPC1	0	0	0	1	0	1	0	0	0	0	0	0	0	1
ORIGPC0	0	0	0	0	1	1	0	0	0	0	0	0	0	1
CAUSAINT1	X	X	X	X	X	X	X	X	X	0	0	1	X	X
CAUSAINT0	X	X	X	X	X	X	X	X	X	0	1	0	X	X
ESCRVCAUSA	0	0	0	0	0	0	0	0	0	1	1	1	0	0
ESCRVEPC	0	0	0	0	0	0	0	0	0	1	1	1	0	0

MODIFICAÇÃO CONTROLE DA

OPALU

FUNÇÃO

* Caso uma interrupção gere overflow

00

+

O controle deve ativar os sinais

01

-

Para os acima, onde NA

10

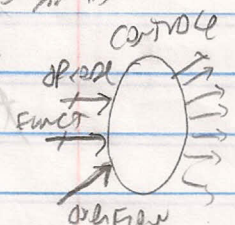
FUNCT

significa: Não alterar

11

OR

Le modo a escrever os resultados
negativos que overflow



2)

a) instruções + carga: LW, SW, TIR, R, beg, J

$$LW = 120 + 80 + 100 + 120 + 100 = 520ps$$

$$\text{Logo: } T = 520ps \rightarrow f = 1,923 \text{ GHz} //$$

$$X1 = a8 + a1 + 2$$

$$N^{\circ} \text{ instruções} = 3 + \overset{\text{beg}}{(X1+1)} + \overset{f}{X1} \overset{\text{bne}}{(1+1+4+25+1+1)} + \overset{\text{add}}{1} \overset{\text{exit}}{1} + 2 -$$

-3-25 \rightarrow 1 vez mais massa na RLA

$$N^{\circ} \text{ instr} = 34X1 - 20$$

$$t_{\text{instr}} = (34X1 - 20) \times 520p = //$$

b) MULTICIRCU : etapa mais lenta

$$\text{Escrita na memória } 150ps \rightarrow f = 6,66 \text{ GHz} //$$

considerando: addi = 4 ciclos \approx add

jal = 3 ciclos \approx f

bne = 3 ciclos \approx beg

lv = 3 ciclos \approx f

syscall = 3 ciclos \approx f

4x1m, 10m

$$N^{\circ} \text{ ciclos} = 130X1 - 83$$

$$t_{\text{multicirc}} = (130X1 - 83) \times 150p = //$$

c) Pipeline ideal: $f = 6,66 \text{ GHz} //$

$N^{\circ} \text{ instruções} \times T$

$$t_{\text{pipe ideal}} = (34X1 - 20) \times 150p = //$$

d) Pipe ss com bolhas

$$N^{\circ} \text{ ciclos com bolhas} = 43 \times X1 - 14$$

$$t_{\text{bolhas}} = (43X1 - 14) \times 150p = //$$

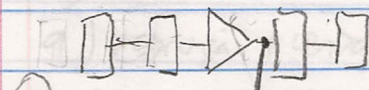
e) melhor caso

$$n_{\text{ciclos}} = 40 \times 1 - 19$$

$$T_{\text{premed}} = (40 \times 1 - 19) \times 150 p = //$$

f) Vão estar escritos os números de 0 a 9999999 sem imprimir o 06

add \$t1

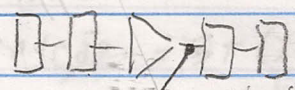


② Forward EXMEM - UCA

addi \$t1

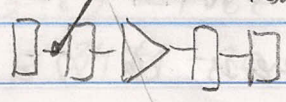


add \$t2



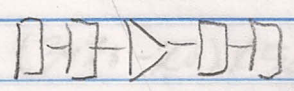
② Forward EXMEM - REG + 1 b0/ha

LAB: 1 beg



②

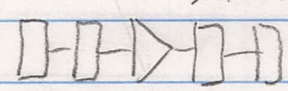
CALL LAB 3



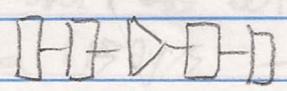
①

1 b0/ha

addi \$t0



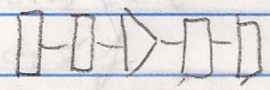
LAB 1



①

1 b0/ha

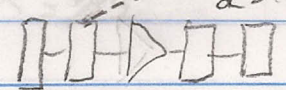
addi \$t0



②

2 b0/ha

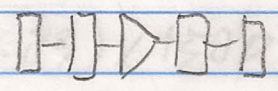
syscall



①

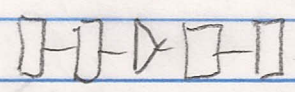
1 b0/ha

one \$t0



②

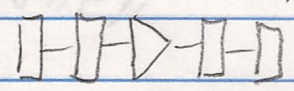
jr \$ra



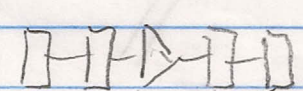
①

1 b0/ha

addi \$t0



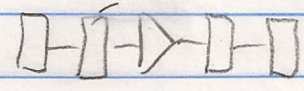
add \$t0



②

2 b0/ha

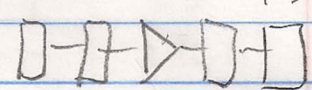
syscall



①

1 b0/ha

jr \$ra



①

1 b0/ha

LAB
b0/ha