



d0 d1 / d2 d3 d4 d5 d6

Nome: GABARITO

Matrícula:   /

## Prova 2

1) (6.0) Na implementação do processador MIPS em pipeline simplificada de 5 estágios 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) jalr \$rd,\$rs # PC=R[\$rs] ; R[\$rd]=PC+4 ; Tipo-R ; rt=00<sub>hexa</sub>; Opcode/Funct=00/09<sub>hexa</sub>  
b)(0.5) jr \$rs # PC=R[\$rs]  
c)(0.5) bneal \$rs,LABEL # If (R[\$rs] != 0) then {PC=LABEL; R[\$ra]=PC+4;} else {PC=PC+4;} ;  
considere: Tipo-I ; Opcode=01<sub>hexa</sub>; rt=1F<sub>hexa</sub> ; Previsto como não-tomado  
d)(0.5) subi \$rt,\$rs,IMM # R[\$rt]=R[\$rs]-{16{IMM[15]},IMM} ; considere: Tipo-I; Opcode=10<sub>hexa</sub>  
e)(0.5) ori \$rt,\$rs,IMM # R[\$rt]=R[\$rs] or {16b'0,IMM}

1.2)(2.5) Especifique adequadamente os valores de todos os sinais de saída da unidade de controle para a CPU completa (ISA com as instruções antigas e as novas acima), desconsiderando apenas o tratamento de exceções.

1.3)(1.0) Especifique o tamanho necessário em bits de cada um dos registradores de pipeline, considerando o controle, a detecção de hazards e forwards. Porém sem o tratamento de exceções;

2) (5.0) Hoje em dia, com a implementação de memórias caches (e seus controladores) cada vez mais eficientes, o tempo de acesso à memória tornou-se bastante reduzido, impactando cada vez menos no projeto do processador, no entanto a inclusão de operações de maior complexidade ainda é um problema a ser resolvido.

Considerando apenas os seguintes tempos de atraso do bloco operativo de uma CPU MIPS:

Operação com a ULA de Inteiros: 100ps

Operação com a ULA de Ponto Flutuante: 240ps (que é dividido em 4 ciclos de clock)

Leitura dos Bancos de Registradores: 80ps

Escrita nos Bancos de Registradores: 90ps

Leitura da memória: 110ps

Escrita na memória: 120ps

Com relação ao trecho de programa em assembly MIPS ao lado, onde o registrador \$a0 = d4d5d6. Responda as questões abaixo especificando como as instruções não implementadas em aula 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? Considere que os registradores podem ser escritos e lidos no mesmo ciclo, que o branch não previsto é avaliado no 3º ciclo, 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 forward sugeridos. Considere que os registradores podem ser escritos e lidos no mesmo ciclo, que o branch é avaliado no 2º ciclo e previsto pelo processador de acordo com a última decisão tomada (decisão inicial como não-tomado).

f)(0.5) Ao final da execução do trecho de programa, qual o valor armazenado no endereço (\$gp+1000) da memória?

g)(1.0) Para maximizar o desempenho da implementação em pipeline deste processador, de 1 a 10, em quantos ciclos de clock você dividiria a operação com a ULA PF? Por quê?

```
99      ...
100     add $t0,$zero,$zero
101 LOOP: lwc1 $f0,0($gp)
102     swc1 $f0,1000($gp)
103     add.s $f0,$f0,$f0
104     swc1 $f0,2000($gp)
105     addi $gp,$gp,4
106     addi $t0,$t0,1
107     slt $t1,$t0,$a0
108     beq $t1,$zero,OUT
109     j LOOP
110 OUT:  sw $t0,1000($gp)
111     ...
```

Muita hora nessa calma! Boa Sorte!!!







OAC - Turma A  
2ª Prova

2008/2

Gabarito

1)

1.1) no verso da Folha de Prova

1.2) CONTROLE

	ID					EX				MEM		WB		
	Branch	Branch	JR	IF.Flush	RegDst	OpAU	OpALU	OpI	OpI	Mem	Mem	ExcReg	MemReg	Link
Format	0	0	0	0	1	10	0	0	0	0	0	1	0	0
LW	0	0	0	0	0	00	1	0	0	0	0	1	1	0
SW	0	0	0	0	X	00	1	0	0	0	1	0	X	0
BEQ	1	0	0	0	X	XX	X	X	0	0	0	0	X	0
JALR	0	0	1	1	1	XX	X	X	0	0	0	1	X	1
JR	0	0	1	1	X	XX	X	X	0	0	0	0	X	0
BranchL	0	1	0	0	0	XX	X	X	0	0	0	0	X	1
SUBJ	0	0	0	0	0	10	1	0	0	0	0	1	0	0
ORI	0	0	0	0	0	11	X	1	0	0	0	1	0	0

1.3)

32

32

$$ID/ID = (PC+4) + (DataMem) = 64 //$$

5

2

4

32

32

32

32

$$ID/EX = (EX) + (M) + (WB) + (PC+4) + (Data1) + (Data2) + (EXT) +$$

5

5

5

5

$$+ I(25-21) + I(20-16) + I(20-16) + I(15-11) = 159 //$$

2

4

32

32

32

5

$$EX/MEM = (M) + (WB) + (PC+4) + (Data1) + (Data2) + (EXT) = 107 //$$

4

32

32

32

5

$$MEM/WB = (WB) + (PC+4) + (DataMem) + (Data1) + (Data2) = 105 //$$



