

Nome: \_\_\_\_\_

Data: 05/07/2011

## 2ª AVALIAÇÃO

**Questão 1:** Considere o trecho de programa para o MIPS apresentado abaixo e que o ciclo de relógio seja **10ns** para o MIPS monociclo e **2ns** para o MIPS multiciclo e pipeline.

```

addi $8, $0, 2
sub $9, $9, $9

inicio: beq $8, $0, fim
        add $9, $8, $9
        addi $8, $8, -1
        j inicio
fim:     and $0, $0, $0
    
```

- Qual é o valor dos registradores **\$8** e **\$9** ao final da execução do programa? **(0,5 ponto)**
- Qual é o CPI deste programa para as **três** implementações do MIPS (monociclo, multiciclo e pipeline)? Mostre os cálculos. **(1,0 ponto)**
- Quanto tempo o trecho de programa demora para ser executado em cada uma das **três** versões de MIPS? **(1,0 ponto)**

**Questão 2:** Escreva instruções do MIPS para realizar as operações listadas abaixo (apenas uma instrução por item). Apenas as instruções abaixo são permitidas. **(1,0 ponto)**

- Copiar o conteúdo do registrador **\$8** no registrador **\$9**.
- Realizar nenhuma operação (**NOP**).
- Subtrair o conteúdo do registrador **\$8** do valor **135** e guardar o resultado no registrador **\$8**.
- Fazer o registrador **\$9** receber o complemento de dois do registrador **\$9**.
- Zerar o registrador **\$11**.

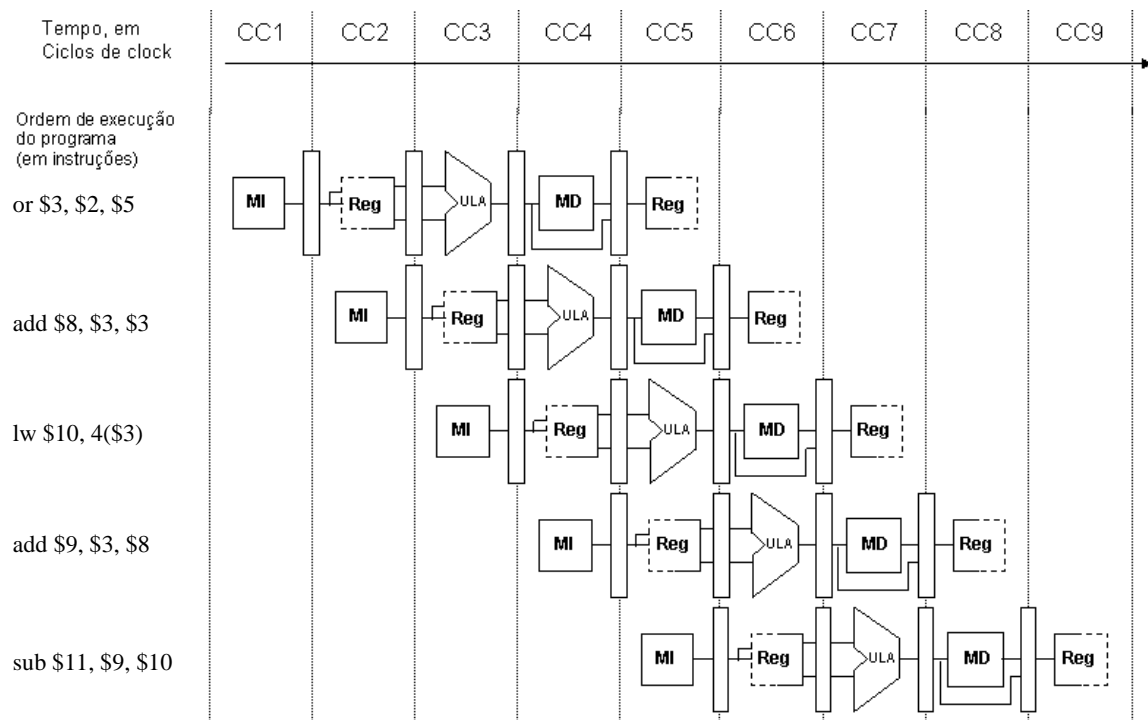
Instruções Permitidas
add
addi
sub
and
or
ori

**Questão 3:** Um projetista de compilador deseja decidir entre três possíveis seqüências de código para a resolução de um problema dados os tipos de instruções e o número de ciclos por instrução de cada tipo, qual seqüência é mais rápida? Apresente os cálculos. **(1,0 ponto)**

Tipo de Instrução	CPI
A	7
B	5
C	4
D	2

Código	Número de Instruções			
	Tipo A	Tipo B	Tipo C	Tipo D
1-9	2	3	2	2
2-9	1	2	2	4
3-8	2	1	2	3

**Questão 4:** Considerando a representação gráfica do pipeline do MIPS que está apresentada abaixo, faça as conexões corretas para utilizar a técnica de adiantamento e resolver os conflitos de dados (1,0 ponto).



**Questão 5:** Considerando que o código abaixo será executado no MIPS pipeline **sem adiantamento**.

```

64    addi $1, $0, 1
68    add $2, $0, $1
72    lw $3, 0x4($1)
76    and $4, $3, $2
80    add $5, $1, $2
84    lw $6, 0x8($5)
88    add $7, $5, $4

```

- a) Indique quais são os conflitos de dados presentes neste código. Identifique o conflito com um número e indique as linhas onde estão as instruções, o registrador envolvido e o tipo de conflito (1A, 1B, 2A, 2B) conforme exemplo abaixo. (1,0 ponto)

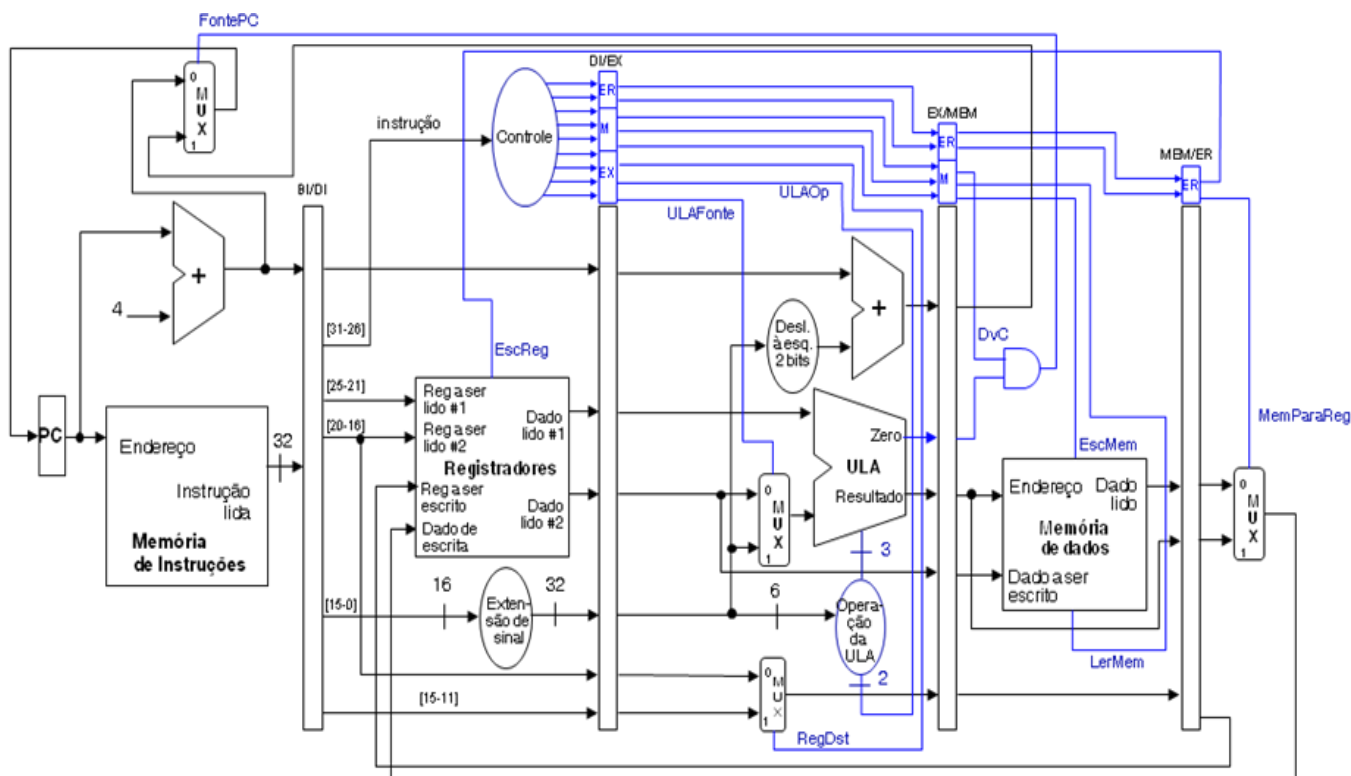
1) Linhas 94 e 98 – registrador \$24 – tipo 1A

- b) Indique quais conflitos identificados na questão acima podem ser resolvidos por adiantamento (apenas cite os números dos conflitos). (0,5 ponto)
- c) Considerando que os conflitos resolvíveis por adiantamento tenham sido eliminados, reescreva o código eliminando TODOS os conflitos SEM usar instruções adicionais e sem modificar o resultado final do programa. (0,5 ponto)
- d) A solução da questão c) criou conflitos de dados diferentes daqueles apresentados na solução da questão a). Indique os conflitos diferentes usando a mesma metodologia da questão a), mas agora usando o código gerado na questão c) como referência. (1,0 ponto)

**Questão 6:** Considere o bloco operativo do MIPS pipeline apresentado a seguir. Deseja-se acrescentar a este processador a instrução **ori**, cujo formato é descrito abaixo. Esta instrução realiza a operação lógica “OR” entre o conteúdo do registrador especificado no campo Rd com um operando imediato (valor absoluto). O operando imediato é um número SEM sinal de 16 bits e deve ser estendido para 32 bits. O resultado é armazenado no registrador Rd.

opcode (6 bits)	não usado (5 bits)	Rd (5 bits)	operando imediato (16 bits)
-----------------	--------------------	-------------	-----------------------------

- a) Adicione, na figura, todas as modificações necessárias no bloco operativo do MIPS pipeline para contemplar a instrução **ori** (não é necessário incluir as conexões de novos sinais de controle, se existirem). **(1,0 ponto)**
- b) O bloco de controle do MIPS pipeline teria que ser modificado a fim de acomodar a instrução **ori**. Explique por que esta modificação é necessária. **(0,5 ponto)**



**Questão 7:** Considerando o diagrama do MIPS pipeline apresentado acima, explique como é possível que o Banco de Registradores seja utilizado tanto no estágio “DI” quanto no estágio “ER” sem gerar paradas no pipeline. **(0,5 ponto)**

**Questão 8:** Considerando o MUX controlado pelo sinal “**RegDst**” do MIPS pipeline apresentado acima, explique porque este MUX foi alocado no estágio “EX” ao invés de estar no estágio “DI”, “MEM” ou “ER”. **(0,5 ponto)**