

## Segunda Prova de Arquitetura de Computador

**1ª Questão) (0.5 ponto)** Relacione a coluna da esquerda com a coluna da direita.

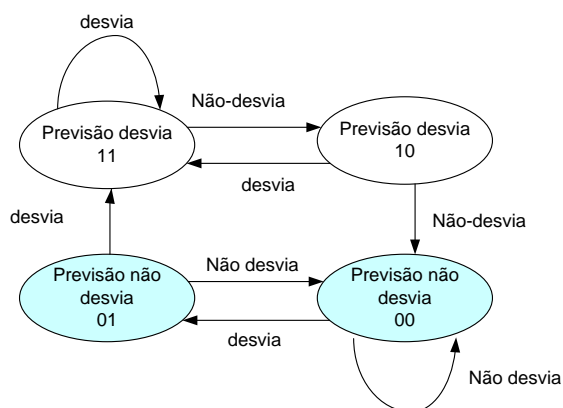
- |                        |  |
|------------------------|--|
| (I) Multicore          | (A) Múltiplos pipelines que operam em paralelo                     |
| (II) Superpipeline     | (B) Execução de instruções fora de ordem em um pipeline.           |
| (III) Superescalar     | (C) Pipelines com grande número de estágios.                       |
| (IV) Pipeline dinâmico | (D) Múltiplos processadores compartilhando um espaço de endereços. |
| (V) Multiprocessadores | (E) Múltiplos processadores em um único encapsulamento             |

Assinale a alternativa que contém a associação correta.

- a) I-B, II-A, III-C, IV-E, V-D.
- b) I-C, II-A, III-B, IV-D, V-E.
- c) I-D, II-E, III-B, IV-A, V-C.
- d) I-E, II-C, III-A, IV-B, V-D.
- e) I-E, II-C, III-A, IV-D, V-B.

**2ª Questão) (1.5 ponto)** Usando o sistema de previsão local de desvio de 2 bits, mostrado na Figura 1, um certo loop é executado duas vezes:

- a) considerando-se que o estado inicial seja 00, calcular a porcentagem de acertos e erros de previsão, considerando-se que o loop termina com 10 iterações;
- b) comparar com o caso em que não use esse sistema de previsão, e apenas considere que a previsão seja sempre de desvio;
- c) comparar com o sistema de previsão de um bit.



**3ª Questão) (1.0 ponto)** Identificar as situações de dependência (WAW, WAR, RAW) na seguinte sequência de código, do MIPS64:

```
DIVD  F1, F3, F5
ADDD  F4, F1, F9
SD     F4, 0(R1)
SUBD   F1, F10, F14
MULD   F9, F10, F8
```

**5ª. Questão) (2.0 ponto)** Assuma que o seguinte código é executado sobre um processador pipeline com 5 estágios, com adiantamento e um preditor de desvio (o qual assume que todo desvio é tomado)

Label1: lw \$1, 40 (\$6) beq \$2, \$3, Label2 //tomado add \$1, \$6, \$4 Label2: beq \$1, \$2, Label1// não tomado sw \$2, 20 (\$4) and \$1, \$1, \$4
--

- Desenhe o diagrama de execução para este código, assumindo que não há slots de atraso e o que desvio executa no estágio EX
- Qual é o speed-up alcançado ao mover a execução de desvio para o estágio ID. Assuma que a comparação no estágio ID não afeta o tempo de ciclo de clock.

**6ª Questão) (1.0 ponto)** Um projetista de compilador deve decidir entre duas sequências de código assembly para uma máquina particular. Existem três diferentes classes de instruções (A, B, C) que compõem cada uma das sequências de código assembly. Cada Classe de instruções apresenta diferentes números de ciclos de relógio por instrução (CPI).

Classe	CPI	Sequência de código	Nº de instruções por classe		
			A	B	C
A	1	1	2	1	2
B	2	2	4	1	1
C	3				

- Qual a sequência de código que executa mais instruções?
- Qual a sequência mais rápida? Quanto?
- Qual o CPI de cada sequência?

**7ª Questão) (3.0 pontos)** Considere a seguinte sequência de instruções, e assuma que estas sejam executadas em um pipeline com 5 estágios

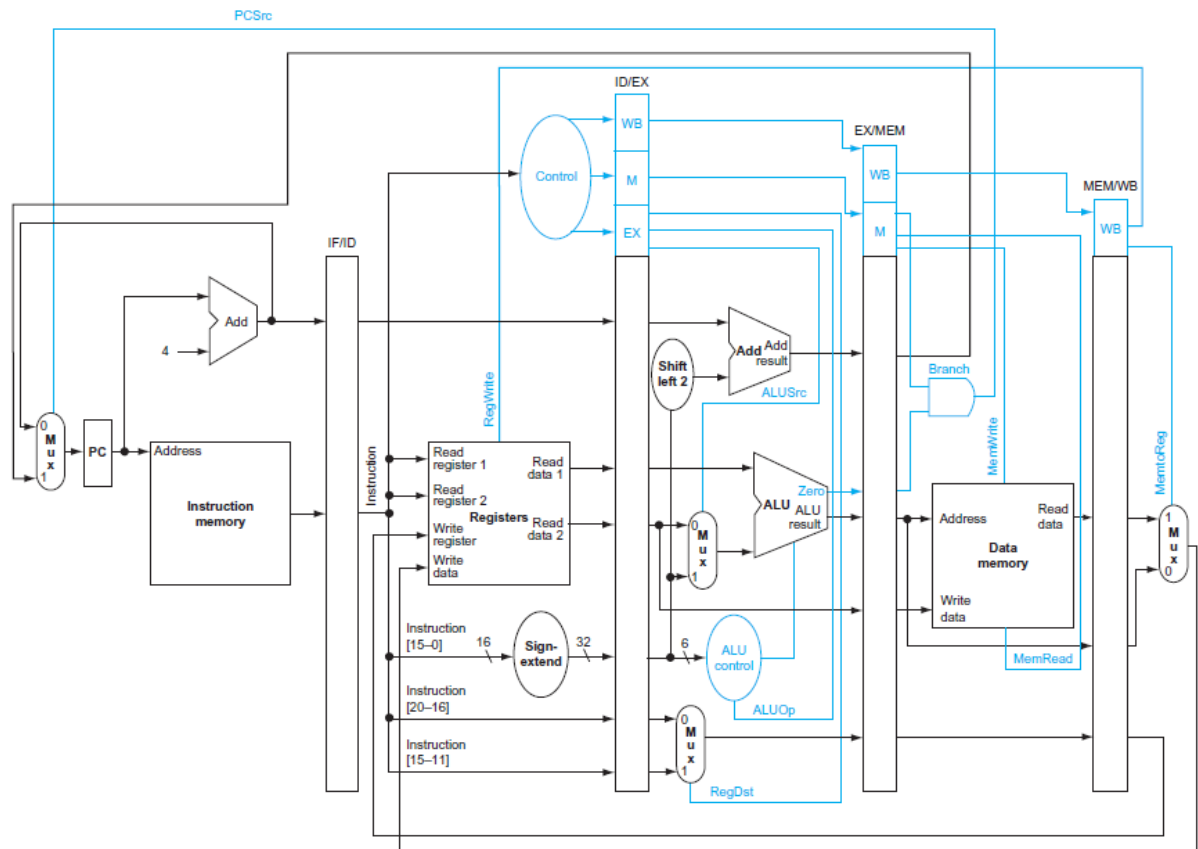
Sequencia Instruções
lw \$1, 40 (\$6)
add \$2, \$3, \$1
add \$1, \$6, \$4
sw \$2, 20(\$4)
and \$1, \$1, \$4

- Quais dependências são conflitos (hazards) que podem ser resolvidos com adiantamento? Quais dependências que são conflitos e irão provocar a parada (bolhas) na execução?
- Se não há adiantamento ou detecção de conflito, insira nops para assegurar a execução correta e desenhe o diagrama de execução do pipeline para este código
- Repita o item anterior, mas adicione nops somente quando um conflito não pode ser evitado por mudando ou rearranjando estas instruções. Você pode assumir o registrador R7 para guardar valores temporários em seu código modificado.
- Um conflito estrutural (duas instruções tentando acessar a memória) pode ser resolvido pelo compilador inserindo uma instrução nops.
- Suponha as instruções abaixo. Qual o procedimento a ser adotado pela unidade de detecção de conflito

load \$1,(10) \$2

add \$2, \$1, \$3

- f) Dado o caminho de dados abaixo, desenhe a unidade adiantamento, mostre os seus sinais de entrada e controle.



**8ª. Questão (1.0 ponto - opcional)** Mostrar o resultado (décimo ciclo) do uso do placar(scoreboard) para a sequência de instruções, considerando-se que a instrução LD leva 1 ciclo para execução; MUL, 6 ciclos. ADD e SUB levam 3 ciclos; e DIV, 20 ciclos.

LD F2, 34(R2)

LD F6, 45(R3)

MUL F0, F2, F4

SUB F8, F6, F2

DIV F10, F0, F6

ADD F6, F8, F2

