

Organização de Computadores I

DCC006

Professor: Omar Paranaíba Vilela Neto

Lista de Exercício 4

1 – Um arquiteto de computadores precisa projetar o pipeline de um novo microprocessador. Ele tem um núcleo de um programa de exemplo com 106 instruções. Cada instrução exige 100ps para terminar.

a) quanto tempo será necessário para executar esse núcleo de programa em um processador sem pipeline?

b) O microprocessador mais moderno tem cerca de 20 estágios de pipeline. Suponha que ele tenha um pipeline perfeito. Quanto é o ganho de velocidade conseguido em comparação com o processador sem pipeline?

c) O pipeline real não é perfeito, pois sua implementação introduz algum *overhead* por estágio do pipeline. Esse *overhead* afetará a latência das instruções, a vazão das instruções ou ambos?

2 – Observe o código abaixo:

```
add x3, x4, x6
sub x5, x3, x2
lw  x7, 100(x5)
add x8, x7, x2
```

a) quais as dependências encontradas?

b) Mostre o pipeline, indicando os caminhos de adiantamento necessários.

3 – Identifique todas as dependências de dados do código a seguir. Quais dependências são *harzards* de dados que serão resolvidos por meio de encaminhamento? Quais dependências são *harzards* de dados que causarão *stall*? Mostre o diagrama do pipeline.

```
add x3, x4, x2
sub x5, x3, x1
lw  x6, 200(x3)
add x7, x3, x6
```

4 – Considere a execução do código abaixo no caminho de dados em pipeline do RISC-V.

```
add x2, x3, x1
sub x4, x3, x5
add x5, x3, x7
add x7, x6, x1
add x8, x2, x6
```

Ao final do quinto ciclo de execução, quais registradores estão sendo lidos e qual registrador será escrito?

5 – No exemplos visto em aula, vê-se que a vantagem no desempenho dos projetos de multiciclo e pipeline é limitada pelo maior tempo exigido para o acesso à memória em relação ao uso da ALU. Suponha que os acessos à memória passem a levar 2 ciclos de *clock*. Desenhe o pipeline modificado. Liste todas as novas situações de adiantamento possíveis e todos os *hazards* possíveis com sua extensão.