

Universidade Federal de Roraima
Departamento de Ciência da Computação
Arquitetura e Organização de Computadores

DISCIPLINA: ARQUITETURA E ORGANIZAÇÃO DE COMPUTADORES

ALUNO: TALLES BEZERRA DE ASSUNÇÃO

LISTA DE EXERCÍCIO 02

1) Quais as vantagens de um processador multiciclo em relação a um uniciclo?

No processador uniciclo a quantidade de ciclos de clock para todas as instruções é definida de acordo com a instrução com maior quantidade de ciclos de clock, já no multiciclo, o ciclo de clock de uma instrução é definido com o seu tipo, então instruções de tipos diferentes podem usar quantidades de ciclos de clock diferentes e dependendo das instruções, o multiciclo pode economizar ciclos de clock em relação ao uniciclo. Também os processadores multiciclo suportam a técnica de pipeline.

2) Quais as modificações necessárias em um processador multiciclo simples para que se introduza a função de pipeline?

Deve-se utilizar 4 registradores entre os 5 estágios do pipeline, um registrador é colocado entre a transição de um estágio para o outro, devendo ser capaz de armazenar todos os dados que passam por ele, dados que poderão ser utilizados por outra instrução no próximo ciclo.

3) Considerando o pipeline do MIPS (simples com MEM compartilhada para instrução e dados) e uma iteração de loop conforme o trecho de programa abaixo, relacione os conflitos que podem ocorrer e seus consequentes stalls. Qual o speedup (por iteração) para o programa em relação à versão sem pipeline?

Loop:

subi \$t2, \$t2, 4

lw \$t1, 0(\$t2) -> conflito de dados, \$t2 ainda não foi calculado

add \$t3, \$t1, \$t4 -> conflito de dados, \$t1 ainda não foi carregado

add \$t4, \$t3, \$t3 -> conflito de dados, \$t3 ainda não foi calculado

sw \$t4, 0(\$t2) -> conflito de dados, \$t4 ainda não foi calculado

beq \$t2, \$0, loop

Cada conflito de dados gera stalls após o IF, atrasando 3 ciclos de clock.

Sem pipeline com ciclos de 1 ns e instruções de salto utilizando 3 ciclos, tipo R e SW, 4 ciclos, e tipo LW, 5 ciclos.

$$4+5+4+4+4+3 = 24\text{ns}$$

Com pipeline com todas as instruções utilizando 5 ciclos.

$$5+4+4+4+4+1 = 22 \text{ ns}$$

$$\text{Speedup} = 24/22 = 1,09 \text{ ns}$$

- 4) Na questão anterior, assuma que a memória de instruções e dados podem ser segmentadas e que o processador aplica a técnica de bypassing. Como ficarão os conflitos e seus stalls, e o speedup?**

Utilizando bypassing, os conflitos de dados são corrigidos e com as memórias segmentadas não ocorrem possíveis conflitos estruturais.

Tempo de execução

$$5+1+1+1+1+1 = 10 \text{ ns}$$

$$\text{Speedup} = 24/10 = 2,4 \text{ ns}$$

- 5) No programa abaixo, relacione as dependências (dados, WAR, WAW e outros) existentes.**

div.d F1, F2, F3

sub.d F4, F5, F1

s.d F4, 4(F10)

add.d F5, F6, F7

div.d F4, F5, F6

RAW:

Sub.d depende de F1 em div.d.

s.d depende de F4 em sub.d.

div.d depende de F5 em add.d.

WAR:

Add.d escreve em F5 que sub.d lê.

WAW

Sub.d usa F4 e pode terminar depois de div.d

- 6) Apresente 5 processadores diferentes que trabalham paralelismo. Adicionalmente, descreva o funcionamento e o tipo de paralelismo de cada um dos processadores, bem como suas vantagens e desvantagens.**

7) Em relação a memória cache. Um computador tem CPI 1 quando todos os acessos à memória acertam no cache. Loads e Stores totalizam 50% das instruções. Se a penalidade por miss é de 25 ciclos e o miss rate é 2%, qual o desempenho relativo se o computador acertar todos os acessos?

Desempenho sem stalls:

$$\begin{aligned}\text{CPU exec. time} &= (\text{CPU clock cycles} + \text{memory stall cycles}) \times \text{clock cycle} \\ &= (\text{IC} \times \text{CPI} + 0) \times \text{clock cycle} \\ &= 1 \times \text{IC} \times \text{clock cycle}\end{aligned}$$

Desempenho com stalls:

memory stall cycles = IC x (CPI + memory access/instructions) x miss rate x miss penalty

$$\begin{aligned}&= \text{IC} \times (1 + 0,5) \times 0,02 \times 25 \\ &= \text{IC} \times 1,5 \times 0,5 \\ &= 0,75 \text{ IC}\end{aligned}$$

$$\begin{aligned}\text{CPU exec. time} &= (\text{CPU clock cycles} + \text{memory stall cycles}) \times \text{clock cycle} \\ &= (1 \times \text{IC} + 0,75 \times \text{IC}) \times \text{clock cycle} \\ &= 1,75 \times \text{IC} \times \text{clock cycle}\end{aligned}$$

Resultado:

$$\frac{\text{CPU exec. time com stalls}}{\text{CPU exec. time sem stalls}} = \frac{1,75 \times \text{IC} \times \text{clock cycle}}{1 \times \text{IC} \times \text{clock cycle}} = \frac{1,75}{1} = 1,75$$

8) Descreva os seguintes conceitos:

a) Write through

Escreve-se o dado no nível da hierarquia atual e no inferior.

b) Write back

Escreve-se o dado apenas no nível de hierarquia atual e, quando o bloco ou página for substituído, ele é atualizado no nível inferior.

c) Localidade Temporal

Posições de memória, uma vez referenciadas, tendem a ser referenciadas novamente dentro de um curto espaço de tempo.

d) Localidade Espacial

Se uma posição de memória é referenciada, posições de memória cujos endereços sejam próximos a ela tendem a ser logo referenciadas.