**Lista 2: Rodrigo de Andrade Rolim Bem >>> 2017009480**

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

**R:**

As principais vantagens da implementação multiciclo são: A possibilidade de fazer com que instruções sejam executadas em quantidades diferentes de períodos do clock devido a divisão de etapas, e a capacidade de compartilhar unidades funcionais dentro do espaço de tempo necessário à execução de uma única instrução ao contrário do uniciclo que duplica alguns componentes.

1. **Quais as modificações necessárias em um processador multiciclo simples para que se introduza a função de pipeline?**

**R:**

É necessária a separação da memória em duas partes independentes. Uma para instrução e outra para dados, e se faz necessária a adição de registradores temporários entre cada etapa.

1. **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?**

**R:**

**Sem Pipeline**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Ciclo | 1 | 2 | 3 | 4 | 5 | 6 |  |
| subi $t2, $t2, 4 | IF | ID | EX | WB |  |  |  |
| lw $t1, 0($t2) |  |  |  |  | IF | ID |  |
|  | | | | | | | |
| Ciclo | 7 | 8 | 9 | 10 | 11 | 12 |  |
| lw $t1, 0($t2) | EX | MEM | WB |  |  |  |  |
| add $t3, $t1, $t4 |  |  |  | IF | ID | EX |  |
|  | | | | | | | |
| Ciclo | 13 | 14 | 15 | 16 | 17 | 18 |  |
| add $t3, $t1, $t4 | WB |  |  |  |  |  |  |
| add $t4, $t3, $t3 |  | IF | ID | EX | WB |  |  |
| sw $t4, 0($t2) |  |  |  |  |  | IF |  |
|  | | | | | | |  |
| Ciclo | 19 | 20 | 21 | 22 | 23 | 24 |  |
| sw $t4, 0($t2) | ID | EX | MEM |  |  |  |  |
| beq $t2, $0, loop |  |  |  | IF | ID | EX |  |

**Com pipeline**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Ciclo | 1 | 2 | 3 | 4 | 5 | 6 |  |  |
| subi $t2, $t2, 4 | IF | ID | EX | MEM | WB |  |  |  |
| lw $t1, 0($t2) |  | IF | ID | EX | MEM | WB |  |  |
| Conflito de dado |  |  | stall |  |  |  |  |  |
| Conflito estrutural |  |  |  | stall |  |  |  |  |
| Conflito estrutural |  |  |  |  | stall |  |  |  |
|  | | | | | | | |  |
| Ciclo | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
| add $t3, $t1, $t4 | MEM | REG | EX | MEM | WB |  |  |  |
| add $t4, $t3, $t3 |  | MEM | REG | EX | MEM | WB |  |  |
| sw $t4, 0($t2) |  |  | MEM | REG | EX | MEM | WB |  |
| beq $t2, $0, loop |  |  |  | MEM | REG | EX | MEM | WB |

**24/13 = 1,846**

1. **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**

**R:**

Conflito RAW: 2ª linha >>> dependência de sub.d com o dado F1 de div.d;

Conflito RAW: 3ª linha >>> dependência de s.d com o dado F4 de sub.d;

Conflito RAW: 5ª linha >>> dependência de div.d com o dado F5 de add.d;

Conflito WAW: sub.d e segundo div.d com o registrador F4;

Conflito WAR: antidependência se sub.d com o add.d com o registrador F5.

1. **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?**

**R:**

1. CPU tempo ideal = (CPU ciclos de clock + Ciclos de Clock de Stall de Memória) x Período de ciclo de Clock =>

CPU tempo ideal = (IC x CPI + 0) x Período de ciclo de Clock =>

CPU tempo ideal = IC x 1 x Período de ciclo de Clock

Ciclos de clock de stall de memória = IC x (1 + 0,5) x 0.02 x 25 = IC x 0,75

CPU tempo real = (IC x 1 + IC x 0,75) x Período de ciclo de Clock =>

CPU tempo real = IC x 1,75 x Período de ciclo de Clock

Comparando os desempenhos com stalls pelo ideal:

1. **Descreva os seguintes conceitos:**

**R:**

a) Escrita somente no cache e atualização na memória somente na substituição do bloco.

b) Escreve ao mesmo tempo em ambas as memórias cache e ram.

c) Se um programa usa um trecho de código, ele tende usar o mesmo trecho de código em um curto espaço de tempo.

d) Um programa tende usar trechos de código próximos ao que foi usado recentemente.