Lista 02 - AOC 2018.2
Bruno Cesar da Silva Claudino - 1201424425

Respostas

1) **As vantagens de um processador multiciclo com a relação ao processador uniciclo ,são que diferentemente do uniciclo que nem todos os componentes são executados em uma execução de instrução,visto que a execução da instrução é feita em um único ciclo de clock . Na execução do multiciclo as instruçoes são divididas em 5 etapas: Busca de instrução, decodificação, busca de operandos, execução e armazenamento do resultado. Em um ciclo de clock é executado um componente para a estapa que esta sendo solicitada, logo cada ciclo faz menos coisas, isso quer dizer que os ciclos podem ser mais curtos. Então como o multiciclo é dividido em 5 etapas e cada etapa é realizada em um ciclo de clock, assim, o ciclo de clock do multicilo é 6 vezes mais curto que o do uniciclo, sua implementação é semelhante ao uniciclo, porem com uma quantidade menor de componentes no caminho de dados, mas sua implementação possui uma lógica de complexidade maior. O multiciclo ainda pode implementar pipeline que é uma técnica que permite a execução de instruções em paralelismo.**

2) **Para a implementação do pipeline no multiciclo são necessárias as seguintes modificações no multiciclo:**

   - **Registradores que funcionarão como buffers temporários, onde serão armazenados os resultados dos componentes e o estado em que a instrução se encontra.**
   - **Flags para a UC(Unidade de Controle) ter o controle dos registradores adicionais.**
   - **Tratamento de stalls evitando a parada do pipeline e das instruções, garantindo um funcionamento eficiente do pipeline.**

3) **Para um processador sem o pipeline funcionará da seguinte maneira: Serão um ciclo de clock para cada uma das etapas de uma instrução, então:**

Linhas	Instrução	Ciclos
1	subi $t2,$t2,4	4
2	lw $t1,0($t2)	5
3	add $t3,$t1,$t4	4
4	add $t4,$t3,$t3	4
5	sw $t4,0($t2)	4
6	beq $t2,$0,loop	3
-	Total	24

Para um processador com o pipeline funcionará da seguinte maneira:

Ciclos	Etapa(Linha)	Consequente
1	IF(1)	
2	ID(1), IF(2)	
3	EX(1), STALL(2)	Conflito de dados.Depende do resultado da linha 1
4	WB(1), STALL(2)	
5	ID(2), IF(3)	
6	EX(2), STALL(3)	Conflito de dados.Depende do resultado da linha 2.
7	MEM(2), STALL(2)	
8	WB(2), STALL(3)	
9	ID(3), IF(4)	
10	EX(3), STALL(4)	Conflito de dados.Depende do resultado da linha 3
11	WB(3), STALL(4)	
12	ID(4), IF(5)	
13	EX(4), ID(5)	
14	WB(4), EX(5)	
15	MEM(5),STALL(6)	Conflito estrutural.2 instruções acessando a memória juntamente.
16	IF(6)	
17	ID(6)	
18	EX(6)	

Logo o speedup é de $\frac{24}{18} \cong 1,33$.

4)

- **RAW:**
  (sub.d e div.d): para a sub.d ser calculada é necessário que a div.d calcule o valor para F1.
- **WAW:**
  (s.d e sub.d): para s.d é necessário que o valor de F4 seja calculado na linha 2.
- **WAR:**
  (add.d e sub.d): O valor de F5 é sobrescrito no linha 4.
  Na linha 5 possui WAW onde F4 é sobrescrito na linha 3, RAW, onde F5 precisa ser calculado na linha 4.
- **Na linha 5 possui WAW onde F4 é sobrescrito na linha 3, RAW, onde F5 precisa ser calculado na linha 4.**

5)

$$CPUexecutiontime = (CPUclockcycles + Memorystallcycles * Clockcycle)$$
$$CPUexecutiontime = (IC * CPI) * Clockcycle$$
$$CPUexecutiontime = IC * 1 * Clockcycle$$

$$Memorystallcycles = IC * (Memocyaccesses/Instruction) * Missrate * misspenalty$$
$$Memorystallcycles = IC * (1 + 0.5) * 0.02 * 25$$
$$Memorystallcycles = IC * 0.75$$

$$CPUexecutiontime(cache) = (IC * 1.0 + IC * 0.75) * Clockcycle$$
$$CPUexecutiontime(cache) = 1.75 * IC * Clockcycle$$

$$\frac{CPUexecutiontime(cache)}{CPUexecutiontime} = \frac{1.75*IC*Clockcycle}{1*IC*Clockcycle} = 1.75$$

6)
- **Write through: escrita ao mesmo tempo no cache e na memória.**
  - **Vantagens:**
    - **Fácil de implementar.**
    - **Um "cache-miss"nunca resulta em escritas na memória.**
    - **A memória tem sempre a informação mais recente.**
  - **Desvantagens:**
    - **A escrita é lenta;**
    - **Cada escrita necessita de um acesso à memória.**
    - **Consequentemente usa mais largura de banda da memória.**
- **Write back: escrita somente no cache e atualização na memória somente na substituição do bloco.**
  - **Vantagens:**
    - **A escrita ocorre à velocidade do cache;**
    - **Escritas múltiplas de um endereço requerem apenas uma escrita na memória;**
    - **Consome menos largura de banda.**
  - **Desvantagens:**
    - **Difícil de implementar;**
    - **Nem sempre existe consistência entre os dados existentes no cache e na memória;**
    - **Leituras de blocos de endereços no cache podem resultar em escritas de blocos de endereços "dirty"na memória.**
- **Localidade Temporal: Se um item é referenciado, ele tende a ser referenciado novamente dentro de um espaço de tempo curto. Se o estudante tiver trazido o livro recentemente para sua mesa, é provável que o faça em breve novamente.**
- **Localidade Espacial: Se um item é referenciado, itens cujos endereços sejam próximos dele tendem a ser logo referenciados.**