Universidade de Itaúna		**Curso:** Ciência da Computação	**Disciplina:** Arquitetura e Organização de Computadores II
Professor (a): Adriana Dornas			Ano: 2021
3º Período	Turno: Noite	**CIU:** 82148	Atividade relativa à aula de **12/04/2021**
Nome: Davi Ventura Cardoso Perdigão			
Paralelismo em Nível de Instruções e Processadores Superescalares e Operação da Unidade de Controle			

1 - As características essenciais da abordagem superescalar para projeto de processadores são:

- A técnica de pipeline permite que várias instruções sejam executadas ao mesmo tempo, mas elas têm que estar em estágios diferentes do pipeline em um dado momento.
- Superescalares incluem todas as características do pipeline, mas além disso, elas podem ter várias instruções executando simultaneamente no mesmo estágio de pipeline.
- Elas têm a capacidade de iniciar múltiplas instruções no mesmo ciclo de clock
- As arquiteturas superescalares permitem que várias instruções sejam despachadas e terminadas por ciclo de relógio
- Uma arquitetura superescalar consiste de um número de pipelines que estão trabalhando em paralelo.
- Dependendo do número e tipo das unidades paralelas disponíveis, um certo número de instruções pode ser executado em paralelo.

2 - A implementação superpipeline é capaz de executar duas fases da pipeline de cada vez. Assim, percebemos que as instruções executadas em cada fase podem ser divididas em duas partes, não sobrepostas, onde cada fase é executada em meio ciclo de clock. A implementação superescalar pode executar duas instruções em paralelo, devido ao fato de existirem duas fases homólogas. Ambas as implementações possuem o mesmo número de instruções executadas ao mesmo tempo, no mesmo estado.

3 - O paralelismo em nível de instrução é o recurso que os computadores possuem para executar várias instruções de forma paralela. Um exemplo dessa aplicação são as técnicas pipeline, arquitetura superescalar, superpipeline.

4 - A) Ocorre quando uma instrução depende do resultado de outra instrução que ainda está no pipeline. Ou seja, uma instrução pode ser obtida e decodificada, mas

não pode ser executada até que a instrução predecessora seja executada, pois ela necessita de dados gerados por ela.

B) Essa dependência ocorre quando se apresenta desvios em uma sequência de instruções, esse fator complica a operação do pipeline. As instruções que vêm depois de um desvio possuem uma dependência procedural com o desvio e não podem ser executadas até que o desvio seja executado.

C) É uma competição de duas ou mais instruções pelo mesmo recurso e ao mesmo tempo. Alguns exemplos de recursos são: memória, cache, entradas para banco de registradores e unidades funcionais.

D) É quando uma instrução escreve em um operando, ao mesmo tempo em que também é escrito por uma instrução sucessora.

E) Antidependência é quando uma instrução lê um operando que é escrito por uma instrução sucessora.

5 - O paralelismo em nível de instruções e de máquina são características importantes para melhorias de desempenho. Abordando primeiramente sobre o paralelismo em nível de instruções, ele é caracterizado pela execução em paralelo por sobreposição de instruções de uma sequência que são independentes. Já o paralelismo de máquina é definido como uma medida da habilidade do processador para obter vantagem do paralelismo em nível de instruções, isso é determinado pelo número de instruções que podem ser obtidas e executadas ao mesmo tempo e pela velocidade e sofisticação dos mecanismos que o processador utiliza para localizar instruções independentes. É importante relatar que um programa pode não ter paralelismo em nível de instruções suficiente para obter vantagem total do paralelismo de máquina, ao passo que, o paralelismo de máquina limitado irá limitar o desempenho, não se importando com qual a natureza do programa.

6 -

• Emissão em-ordem com conclusão em-ordem: Política mais simples, é a ordem exata que seria alcançada pela execução sequencial (emissão em-ordem) e pela escrita de resultados na mesma ordem (conclusão em-ordem).
• Emissão em-ordem com conclusão fora-de-ordem: A conclusão fora-de-ordem é usada em processadores RISC escalar para melhorar o desempenho das instruções que requerem múltiplos ciclos.
• Emissão fora-de-ordem com conclusão fora-de-ordem: As instruções são emitidas a partir da janela de instruções sem se preocupar com a ordem do programa original, tendo como única restrição que o programa se comporte corretamente.

7 - A Janela de Instruções armazena o resultado da decodificação das instruções e isola o estágio de busca e decodificação de instruções dos estágios de execução propriamente dita das instruções.

8 - Renomeação de registradores refere-se à uma técnica utilizada para evitar a desnecessária serialização das operações de um programa, imposta pelo reuso dos registradores por essas operações.

9 - Nos processadores superescalares incluem mais de uma unidade funcional em cada *pipeline*, as quais permitem a execução de mais de uma instrução simultaneamente (no mesmo ciclo de *clock*).

A arquitetura pipeline já permite que diversas instruções sejam executadas ao mesmo tempo, desde que estejam em **estágios diferentes do pipeline**. Porém, ainda que várias instruções sejam executadas concorrentemente (em cada estágio do pipeline), apenas uma instrução encontra-se na fase de execução, ou seja, o sistema continua sendo SISD.

10 - Uma arquitetura superescalar deve possuir uma série de componentes especiais para executar mais de uma instrução por ciclo:

- Unidade de busca de instruções: capaz de buscar mais de uma instrução por ciclo. Possui também um preditor de desvios, que deve ter alta taxa de acerto, para poder buscar as instruções sem ter que esperar pelos resultados dos desvios.
- Unidade de decodificação: capaz de ler vários operandos do banco de registradores a cada ciclo. Note que cada instrução sendo decodificada pode ler até dois operandos do banco de registradores.
- Unidades funcionais inteiras e de ponto flutuante: em número suficiente para executar as diversas instruções buscadas e decodificadas a cada ciclo.

11 - A operação de um computador, na execução de um programa, é composto por uma sequencia de ciclos de instrução por ciclo de máquina. Essa sequencia de ciclos de instrução se difere da sequencia de escrita de instruções que compõem o programa, por causa da ramificação de instruções. A execução real de instruções segue uma sequência de tempo de instruções.

12 - Micro-operações é a operação elementar da CPU, executada durante um pulso do clock. Uma instrução consiste na sequência de uma série de micro-operações.

13 - A unidade de controle de um processador executa duas principais funções: faz com que o processador execute micro-operações na sequência adequada,

determinada pelo programa a ser executado e também gera os sinais de controle, que fazem com que cada micro-operação seja executado.

14 - Caracterização da unidade de controle:

1) Define os elementos básicos do processador.

2) Descreve as micro-operações que o processador deve executar.

3) Determina as funções que a unidade de controle deve realizar para causar a execução das micro-operações.

15 - Basicamente, uma unidade de controle efetua duas tarefas:

1) Sequenciamento: A unidade de controle faz com que o processador percorra uma série de micro-operações na sequencia apropriada, com base no programa que está sendo executado.
2) Execução: A unidade de controle faz com que cada micro-operação a seja executada.

16 - Entradas:

• Clock: A unidade de controle faz com que micro-operação seja executado a cada pulso do clock, também chamado de ciclo de processador ou de ciclo de relógio.

• Registrador de instrução: o código de operação da instrução corrente é usado para determinar micro-operações que devem ser executadas durante o ciclo de execução.

• Códigos de condição: essa informação é requerida pela unidade de controle para determinar o estado do processador e a saída de operações previamente executadas pela ULA.

• Sinais de controle do barramento de controle: a parte de controle do barramento de sistema fornece sinais para a unidade de controle, tais como sinais de interrupção e de reconhecimento.

Saídas:

• Sinais de controle interno ao processador: Esses sinais são de dois tipos: os que causam movimentação de dados de um registrador para o outro e os que ativam funções específicas da ULA.

• Sinais de controle para o barramento de controle: existem também dois tipos de sinais: sinais de controle para a memória e sinais de controle para o módulo de E/S.

17 - Três tipos de sinais de controle:

1) Os que ativam uma função da ULA.
2) Os que ativam um caminho de dados.
3) Os sinais de barramento externo do sistema ou para alguma interface externa.

18 - Na implementação por hardware, a unidade de controle é essencialmente uma combinação de circuitos. Seus sinais lógicos de entrada são transformados em um conjunto de sinais lógicos de saída, que são os sinais de controle.

19 - Unidade de controle de hardware é uma combinação circuitos, na qual enviam sinais lógicos que são transformados em sinais de saída lógicos que funcionam como sinais de controle. E na unidade de controle microprogramada, a lógica é especificada pelo microprograma. O microprograma consiste de uma sequência de instruções na linguagem do microprograma, que são instruções muito simples que especificam micro-operações.

20 - Basicamente, ela contém configurações que definem a unidade de controle. Assim, ela define a sequência de micro-operações a serem efetuadas durante cada ciclo (busca, indireção, execução, interrupção) e especifica a ordem na qual esses ciclos ocorrem.

21 - Em uma microinstrução horizontal, cada bit de um campo de controle é ligado a uma linha de controle. Em uma microinstrução vertical, é usado um código para cada ação a ser efetuada e o decodificador traduz esse código em sinais de controle individuais. As vantagens de microinstruções verticais é que elas são mais compactas que microinstruções horizontais, ao custo de uma pequena lógica e tempo de atraso adicional.