

Disciplina SCE541
Arquitetura de Computadores

Grupo nr.: 30

Nome: Ellian Carlos Oliveira Costa - N° USP: 11846324

Nome: Guilherme Cremasco Gulmini - N° USP: 11816077

Nome: Jonattan Willian da Silva - N° USP: 11821278

Nome: Lorenzo Paciello - N° USP: 10734182

1ª Questão: Explique o que são, compare e exemplifique as arquiteturas SISD, SIMD, MISD, MIMD.

2ª Questão: Explique o que são, compare e exemplifique arquiteturas com memória compartilhada e memória distribuída.

3ª Questão: Explique o que são, compare e exemplifique as seguintes máquinas:

1. Processador com pipeline de operações
2. Processadores Superescalares
3. Processadores Paralelos

4ª Questão: Explique as limitações intrínsecas do paralelismo: Dependência de dados, Dependência de desvio, Conflito de recurso (ULA) e o que pode ser feito para minimizar esses problemas.

5ª Questão: Explique renomeação de registradores.

6ª Questão: Explique o que são, compare e exemplifique as seguintes técnicas: Delayed Branch e Otimização do Branch.

Obs. Procure cobrir os pontos abordados em aula em suas respostas, mantendo o limite de UMA folha para cada questão.

1- Essas quatro nomeações se referem à Classificação de Flynn, que leva em consideração o fluxo de dados e de instruções e a maneira como é feita em uma determinada máquina. São elas:

- SISD: único fluxo de instruções e único fluxo de dados. Nessa arquitetura, as instruções são executadas serialmente sobre um único fluxo de dados e são usadas uma única unidade de controle e uma única unidade de processamento. Exemplo: uma máquina de Von Neumann simples, como um arduíno.
- SIMD: único fluxo de instruções, múltiplos fluxos de dados. Um mesmo programa é enviado a vários processadores em forma de fluxos de comandos iguais, que operam ao mesmo tempo sobre fluxos de dados diferentes. Possui uma única unidade de controle para todas as unidades de processamento. Exemplo: um pedal que processa os dados sonoros vindo de várias guitarras.
- MISD: múltiplos fluxos de instruções, único fluxo de dados. Um único fluxo de dados é operado por fluxos de comandos diferentes em várias unidades de processamento. Exemplo: vários filtros que processam o som de uma única guitarra, cada um aplicando um efeito diferente.
- MIMD: múltiplos fluxos de instruções, múltiplos fluxos de dados. Processos diferentes operam sobre fluxos de dados diferentes, podendo ser de forma síncrona ou não. Possui várias unidades de controle, uma para cada unidade de processamento. Exemplo: supercomputador moderno.

2- Na memória compartilhada, cada processador pode endereçar toda a memória do sistema, sendo a comunicação entre os processos bastante eficiente, pois os dados não precisam se movimentar fisicamente. No entanto, esse tipo de arquitetura não é muito escalável, devido ao limite de memória. Na memória distribuída, cada processador endereça somente a própria memória local e, ao contrário da memória compartilhada, é altamente escalável, permitindo, assim, a construção de computadores massivamente paralelos. No entanto, é necessária a distribuição da carga entre os processadores.

3 - Processadores com pipeline de operações são máquinas que dividem o processamento de instruções em etapas e assim conseguem executar diversas instruções ao mesmo tempo, desde que estejam em etapas diferentes. Um exemplo desse processo é separar as instruções nas etapas fetch de instrução, decodificação, cálculo de operandos, execução, escrita dos operandos, enquanto uma instrução está sendo decodificada, outra pode estar sendo recebida da memória (fetch), e outra já executada pode estar tendo seus resultados escritos.

Processadores Superescalares são máquinas capazes de receber e executar diversas instruções ao mesmo tempo, logo, as mesmas possuem diversos pipelines. Esse tipo de processador também utiliza de diversas técnicas para maximizar o número de instruções executadas a cada ciclo de clock, além dos pipelines, algumas dessas técnicas são inversão de ordem de instruções, busca de instruções independentes, previsão de desvio, registradores adicionais e renomeação de registradores. Um exemplo são os computadores POWER da IBM que implementaram um sistema de instruções e pipelines em computadores superescalares através do padrão POWER.

Processadores Paralelos são máquinas de um tipo de arquitetura de computadores que consiste em utilizar diversos processadores para resolver um mesmo problema (computação distribuída). Estes podem ser classificados a partir da quantidade de fluxos de dados e fluxos de instruções. Um exemplo de computadores paralelos são os PCs modernos, qualquer chip processador, seja Intel ou AMD, hoje conta com múltiplos processadores, além de múltiplos fluxos de dados e múltiplos fluxos de instruções, estes são MIMD de acordo com a classificação de Flynn.

Qualquer processador ou microprocessador moderno hoje conta com algum tipo de pipeline de operações, ambos processadores superescalares e processadores paralelos utilizam de técnicas de pipeline a fim de agilizar o processamento de instruções. Processadores superescalares conseguem agilizar muito o processamento de instruções e são compatíveis com processamento paralelo, tendo como desvantagem apenas cenários extremos onde a previsão de desvio ou de dependência falha, algo raro nos computadores modernos. O processamento paralelo também consegue agilizar muito a computação de uma tarefa, porém este é limitado pela parte paralelizável de um programa, isto é, existem partes em programas que são necessariamente síncronas, seja por dependência de dados ou até mesmo erro do programador, assim há um limite para o quanto o paralelismo consegue acelerar o processamento, este potencial é definido pela lei de Amdahl.

4- A dependência de dados pode ser classificada em três tipos:

- Dependência de dados verdadeira: ocorre quando uma instrução depende do resultado de outra instrução que ainda está em execução (Leitura após Escrita) e, portanto, sendo evitada por meio da execução da última instrução após o término da primeira.
- Dependência de saída: ocorre quando uma instrução escreve em um registrador que é escrito por uma instrução sucessora (Escrita após Escrita). Pode ser minimizada pelo uso da técnica de renomeação de registradores.
- Antidependência: ocorre quando uma instrução lê um registrador que é escrito por uma instrução sucessora (Escrita após Leitura). Pode ser minimizada pelo uso da técnica de renomeação de registradores.

A dependência de desvio pode ser vista quando uma instrução, situada após uma instrução de desvio, é executada apenas se o desvio não acontecer. Caso contrário, essa instrução não pode ser executada, pois será “desviada” pela instrução de desvio. Pode ser minimizada com o uso de técnicas como a redução dos atrasos no desvio ou da previsão de desvios tradicional, por exemplo.

A dependência de recurso na ULA ocorre quando há o uso simultâneo de uma mesma unidade funcional por duas ou mais instruções, podendo ser solucionada pela duplicação da unidade funcional ou pela implementação de um pipeline do recurso.

5- Renomeação de registradores é uma técnica que consiste em trocar o registrador em uso a fim de permitir que duas ou mais instruções, que antes deveriam ser executadas sequencialmente por estarem utilizando o mesmo e único registrador, possam ser executadas paralelamente, pois as instruções estarão utilizando registradores diferentes (trocados pela técnica).

6- Na técnica Delayed Branch, a instrução após o desvio é sempre executada, sendo a próxima instrução chamada de delay slot. Por fim há o reordenamento das instruções.

Otimização de Branch é uma técnica para minimizar o recurso utilizado para a execução de um programa através de manipulações de instruções e previsões de desvios. Boa parte da otimização de branch se deve pelos circuitos que preveem quando uma branch irá ou não desviar antes de calcular a resposta, podendo assim tomar ações caso o valor previsto seja o desvio ou o não desvio. A previsão de branch é útil porque caso um desvio vá ocorrer é possível saber antes que ele ocorra e assim saber qual instrução fazer fetch no pipeline, um erro de previsão obviamente acarreta em alguns ciclos de clock perdidos.