Universidade de São Paulo
Instituto de Ciências Matemáticas e de Computação
Departamento de Ciências de Computação e Estatística

Disciplina SCE541
Arquitetura de Computadores

Grupo nr.: 07

Nome: Matheus Odebrecht Oliveira Nº USP: 11832670
Nome: Uriel Ramiro Munerato Nº USP: 11816060
Nome: Lucas Carneiro de Souza Nº USP: 11914601
Nome: Victor Kenji Matsushita Nº USP: 11816021

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 Superrescalares
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.

Questão 1

De modo geral, as regras para cada classificação (SISD, SIMD, MISD e MIMD) estão relacionadas ao fluxo de instruções e ao fluxo de dados. O fluxo de dados faz referência à comunicação da unidade de processamento com a memória, na qual um fluxo é considerado Single (Único) e dois ou mais fluxos são considerados Multiple (Múltiplo). O fluxo de instruções define a multiplicidade de processadores, onde as tarefas podem ou não ser desempenhadas simultaneamente. Essa definição é conhecida como Taxonomia de Flynn.

SISD – Possui fluxo único de instruções e dados.
Exemplo: Arquitetura de von Neumann.

SIMD – Possui fluxo único de instruções e múltiplo de dados.
Exemplo: Gráficos 3D, processamento de imagem, vídeo e áudio.

MISD – Possui fluxo múltiplo de instruções (tarefas) e único de dados.
Exemplo: Algoritmos de criptografias e filtros de frequência (operando sobre um único fluxo de sinal).

MIMD – Possui fluxo múltiplo de instruções (tarefas) e dados.
Exemplo: Processadores de celulares e computadores modernos.

Questão 2

Arquiteturas com memória compartilhada são definidas pelo acesso concomitante de múltiplos processadores. A sincronização das tarefas é feita por meio da ordem de escrita e leitura na memória, tomando cuidado para não modificar um endereço de memória por uma tarefa enquanto outra o estiver acessando.
Por outro lado, as arquiteturas com memória distribuída impõem barreiras ao acesso dela mesma, uma vez que cada memória local só pode ser acessada pelo seu processador. A comunicação e sincronização é feita através da troca de mensagens.
Os sistemas comuns do nosso dia-a-dia (celulares e computadores) usam memória compartilhada, e servidores como os da Google ou Facebook apresentam memória distribuída. Um problema interessante que se observa na segunda ocasião é a sincronização de tempo: como a memória é distribuída, a ordenação das tarefas é fundamental, pois caso haja alguma diferença de tempo que inverta a ordem de duas requisições a um processador, dados que não deveriam ser alterados podem acabar sendo, antes da leitura necessária.

Questão 3

1. São processadores com capacidade de executar operações em sincronia, tentando evitar que as estruturas de processamento fiquem desocupadas na execução de um ciclo de instrução para obter máximo proveito da capacidade de processamento. Um exemplo é o IBM 370/168.

2. Faz o uso de várias pipelines diferentes podendo manipular várias instruções ao mesmo tempo. Ele procura eliminar dependências entre instruções e pode executá-las fora da ordem original. Para isso ele pode utilizar de registradores adicionais e renomeação de registradores. IBM RS/6000 e supercomputadores são exemplos desse tipo de arquitetura

3. Os processadores paralelos são 2 ou mais processadores com pipeline de operações em paralelismo que podem executar várias instruções ao mesmo tempo, funcionando efetivamente também como superescalar. Intel Core i9 é um exemplo, assim como o A14 Bionic da Apple de arquitetura ARM.

Questão 4

Dependência de dados: Uma execução não pode ser terminada antes do término da execução anterior, pois ela depende do seu dado. Como em:

ADD R1,R2
MOVE R3,R1

Para minimizar esse problema, pode-se realizar a troca de registrador para tentar manter o pipeline e evitar conflitos.

Dependência de desvio: Ocorre em if e for statements, por exemplo. Sempre quando há um jump, o pipeline deve ser esvaziado para começar novamente o processamento a partir desse ponto. Uma solução possível é a previsão de desvio tradicional que processadores superescalares utilizam.

Conflito de recurso (ULA): Quando duas ou mais instruções requerem o mesmo recurso na ULA. Para contornar esse problema é possível somente duplicar o recurso ou utilizar pipeline do mesmo.

Questão 5

A Renomeação de Registradores é uma técnica que consiste na renomeação de certos registradores, com o objetivo de aumentar o paralelismo disponível, isto é, permitir que instruções que anteriormente deveriam ser executadas em série agora possam ser executadas ao mesmo tempo e, consequentemente, resolvendo o problema de dependência de saída.

Questão 6

Delayed Branch é uma técnica de previsão de desvio, a qual tem como objetivo evitar o esvaziamento do pipeline colocando um opcode chamado NOOP, o qual não exerce nenhuma operação.

Otimização do Branch é uma técnica de otimização do pipeline a qual pega opcodes, que normalmente seriam executados antes da operação JUMP e que não interferem nela, e os colocam após, para que caso o “jump” aconteça, os ciclos sempre ficam trabalhando, e caso o “jump” não aconteça, não haverá a perda de processamento com os NOOPs, como acontece no Delayed Branch.

Exemplo:

End:	Normal:	Delayed Branch:	Otimização do Branch:
099	SUB A,1	SUB A,1	ADD B,3
100	ADD B,3	ADD B,3	JUMP B>3 104
101	JUMP B>3 104	JUMP B>3 105	SUB A,1
102	ADD A,B	NOOP	ADD A,B
103	SUB C,A	ADD A,B	SUB C,A
104	STORE end,B	SUB C,A	STORE end,B
105		STORE end,B	
106			