**UNIVERSIDADE DO ESTADO DE MINAS GERAIS – UEMG**

**CAMPUS DE FRUTAL**

**CURSO DE SISTEMAS DE INFORMAÇÃO**

**PIPELINE**

**LEOPOLDO FERREIRA DE PAULA**

**Frutal (MG)**

**2015**

Sumário

**INTRODUÇÃO2**

1.1 Características2

1.2 Exemplos, Suas Vantagens e Desvantagens3

1.3 Problemas Relatados4

**Conclusão**

**Referências**

**INTRODUÇÃO**

Antigamente os processos eram feitos linearmente onde um processo era feito instrução por instrução como se fosse por uma única pessoa, O pipeline pode ser analogicamente pensado como uma linha de produção onde cada parte de um processo é feito por um único indivíduo, sendo assim, um processo completo não precisa ser finalizado para que outro seja iniciado, dando exemplo imagine pessoas cozinhando em conjunto, onde haja 3 pessoas realizando 3 operações, o pipeline é usado como se cada pessoa fosse responsável por um processo, sendo assim a primeira pessoa inicia o primeiro processo, assim que termina, recomeça o seu processo, enquanto a segunda pessoa inicia o seu processo, fazendo assim com que não precise um processo todo ser feito, para que se inicie outro. Em linguagem computacional, um processo é dividido em instruções pelo processador, quando uma porção especificada para realizar aquela etapa é liberada, uma nova é iniciada, liberando a antiga para que se inicie um novo processo e haja execuções simultâneas.

**CARACTERÍSTICAS**

Os processadores possuem seu clock, que é a velocidade do processados, ou seja, a frequência que o processador executa tarefas, quanto maior o clock, mais rápido um processo é terminado, o Pipeline aumenta o número de instruções a serem executadas simultaneamente, melhorando o desempenho do processo.

Até o processador 386, os processadores eram capazes de realizar apenas uma instrução por vez, inicialmente usado em processadores RISC, no processador 486 utilizou-se o pipeline que consistia em dividir o processador em vários estágios distintos.

Cada processador utiliza-se de um número de estágios de pipeline, a cada um deles, mais processos podem ser realizados simultaneamente, Pentium II possui 10 estágios, Pentium 4 possui 20, isso significa que cada instrução demora 20 ciclos a ser processada, porém, realizam 20 processos ao mesmo tempo.

**EXEMPLOS**

Superpipele: É fácil perceber que um grande número de divisões na execução de uma instrução (numero de estágios de Pipeline) resulta em um melhor desempenho de processamento, pois o numero de instruções executadas em série será maior. Isso se o Pipeline não tiver que ser esvaziado (todos os estágios devem esperar o encerramento de uma instrução especifica) com muita freqüência. O Superpipelining se baseia nestes princípios.

A desvantagem desta técnica, quando comparado a um Pipeline, é que este possui uma maior complexidade de desenvolvimento e uma maior dependência de detecção de desvios, visto que o esvaziamento do Pipeline resultaria num desperdício de mais estágios.

Pipeline Super Escalar: Este Pipeline é obtido aumentando-se o numero de Pipelines. Ou seja, mais de um Pipeline trabalhando de forma paralela no processador. Para ele tenha um bom desempenho as instruções não devem ter muita latência, para que diversos Pipelines trabalhem de forma sincronizada. Desta forma o processador deve possuir um bom algoritmo de tratamento de latências.

A grande vantagem do Pipeline Super Escalar é que o paralelismo na execução das instruções acontece de forma real, possibilitando um grande aumento de desempenho. Os pontos negativos estão no fato de que as micro-instruções devem ser cuidadosamente estudadas e a complexidade no tratamento de desvios aumenta significativamente.

Pipeline com Predição de Desvio: Predição de desvios são técnicas utilizadas para evitar o esvaziamento do Pipeline, deixando o fluxo de processamento das instruções contínuo e sem quebras. Estas técnicas podem ser implementadas por Hardware ou Software e são fundamentais para um uso eficiente do Pipeline.

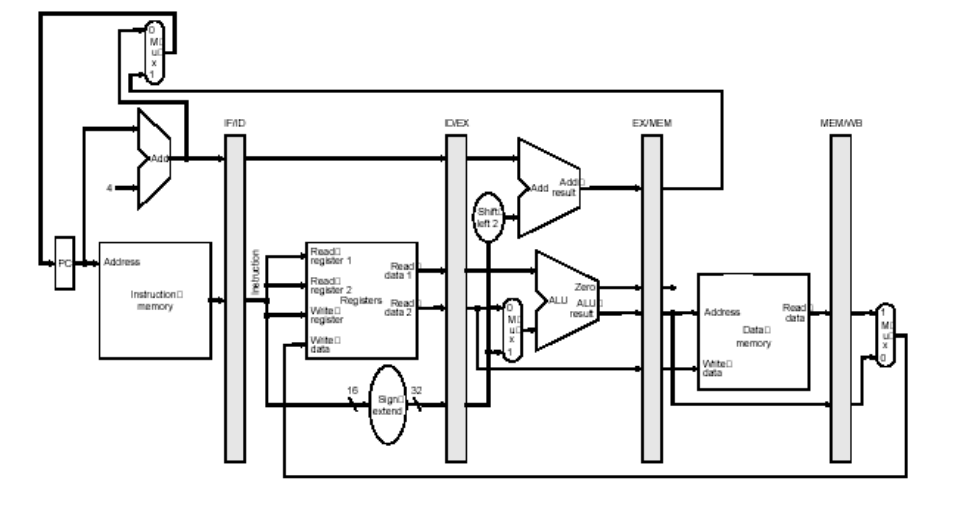
A predição de desvio pode ser estática ou dinâmica. Na predição estática a unidade de controle assume que o desvio sempre ocorrerá, resultando numa taxa de 50% de acerto. Na predição dinâmica a unidade de controle utiliza informações coletadas em tempo de execução para decidir-se sobre o desvio.

Pipeline com Execução Fora de Ordem: A Execução Fora de ordem faz o processador reordenar as instruções para melhor aproveitar o Pipeline e evitar a seu esvaziamento. Esta reordenação é complexa devido ao fato de que instruções que dependem uma das outras devem ser tratadas de forma especial.

O uso da execução fora de ordem necessita de um controle para reordenação das instruções após executá-las.

Estes “tipos de Pipeline” são meramente teóricos, na prática eles podem possuir mais de uma dessas características a fim de obter um melhor de desempenho.

Arquitetura de processador com Pipeline



Disponível em: <http://www.dcc.ufrj.br/~gabriel/arqcomp2/Pipeline.pdf> Página 11.

**PROBLEMAS RELATADOS**

Os estágios podem ter tempos de execução diferente, sendo assim duas soluções são cabíveis ao problema, a primeira solução seria: Implementar esses estágios como um pipeline onde cada sub-estágio possui tempo de execução semelhante aos demais estágios do pipeline principal., e a segunda seria: Replicar esse estágio, colocando réplicas em paralelo no estágio principal. O número de réplicas é dado pela razão entre o tempo do estágio mais lento e os demais.

Outro problema também citado por Gabriel P. é que o sistema e memória é incapaz de manter o fluxo de instruções: O uso de memória cache com alta taxa de acerto e tempo de acesso compatível com o tempo de ciclo do pipeline.

Dependências ou Conflitos (“Hazards”) – Conflitos Estruturais: Pode haver acessos simultâneos à memória feitos por 2 ou mais estágios. Dependências de Dados: As instruções dependem de resultados de instruções anteriores, ainda não completadas. Dependências de Controle: A próxima instrução não está no endereço subsequente ao da instrução anterior.

**CONCLUSÃO**

De forma singular o pipeline não reduz o tempo para se completar a execução de um processo, porém é eficaz quando se trata de executar diferentes processos simultaneamente, cada tipo de pipeline é mais eficiente em certos aspectos, quanto mais estágios o processador possui, mais processos simultâneos ele pode executar, e quanto mais rápido o seu clock, mais rápido o ciclo de processos será executado.

**REFERÊNCIAS**

Cin UFPE – RESUMO pipeline, Disponível em: <www.cin.ufpe.br/~jvwr/RESUMO%20pipeline.doc> Acesso em: 18 de jun 21:00.

Hardware – Pipeline, Disponível em: <http://www.hardware.com.br/termos/pipeline> Acesso em: 18 de jun 19:00.

Cleberopo – Pipeline, Disponível em: <http://pt.slideshare.net/cleberopo/pipeline-18503784> Acesso em: 18 de jun 19:30.

TecMundo – Processador desvendando o mistério do clock e velocidade real, Disponível em: <http://www.tecmundo.com.br/4415-processador-desvendando-o-misterio-do-clock-e-da-velocidade-real.html> Acesso em: 18 de jun 20:00.

UFRJ – Pipeline, Disponível em: <http://www.dcc.ufrj.br/~gabriel/arqcomp2/Pipeline.pdf> Acesso em: 18 de jun 20:30.