

LISTA DE EXERCÍCIO 02

1) Quais as diferenças de um processador multiciclo em relação a um uniciclo?

R: O processador monociclo executa cada instrução em ciclos de relógio e o multiciclo executa cada uma das instruções em um ciclo próprio de clock.

2) Quais as modificações necessárias em um processador multiciclo simples para que se introduza a função de pipeline?

R: Dividimos os caminhos de dados e adicionamos na etapa de Busca e decodificação de duas instruções por ciclo, Leitura de 4 registros e escrita de 2 registros por ciclo, ALU dedicada ao cálculo de endereço de memória e apenas a 2ª instrução do DP tem capacidade de aceder à memória.

3) No programa abaixo, relacione as dependências (dados, WAR, WAW e outros) e conflitos existentes para execução em um processador MIPS usando pipeline.

div.d F1, F2, F3 sub.d F4, F5, F1 s.d F4, 4(F10) add.d F5, F6, F7 div.d F4, F5, F6

4) Descreva os seguintes conceitos:

a) Write through:

R: Quando o sistema escreve para uma zona de memória, que está contida no cache, escreve a informação, tanto na linha específica do cache como na zona de memória ao mesmo tempo. Este tipo de caching providencia pior desempenho do que Write-Back Cache, mas é mais simples de implementar e tem a vantagem da consistência interna, porque o cache nunca está dessincronizada com a memória como acontece com a técnica Write-Back Cache. Algumas de suas vantagens: fácil de implementar; um "cache-miss" nunca resulta em escritas na memória; a memória tem sempre a informação mais recente.

b) Write back:

R: É uma técnica de atualização da memória cache, na qual as escritas são feitas apenas nessa memória, e a memória principal só é atualizada se o bit de atualização do bloco substituído tiver o valor 1. Exemplo: A escrita de um endereço é feita inicialmente numa linha do cache, e somente no cache. Quando mais tarde algum novo endereço precisar desta linha do cache, estando esta já ocupada, então o endereço inicial é guardado na memória e o novo endereço ocupa-lhe o lugar na respectiva linha do cache. Para reduzir a frequência de escrita de blocos de endereços na memória aquando da substituição é usado um "dirty bit", este é um bit de estado, ou seja, quando o endereço é instanciado

inicialmente numa linha do cache, estando essa linha vazia, o valor inicial é implicitamente '0', quando o bloco do endereço é modificado (quando ocorre uma substituição) o valor inicial passa a '1' e diz-se que o bloco do endereço está "dirty". Algumas de suas vantagens: A escrita ocorre à velocidade do cache; escritas múltiplas de um endereço requerem apenas uma escrita na memória; consome menos largura de banda.

c) Localidade Temporal

R: O Princípio da Localidade Temporal diz que um dado acessado recentemente tem mais chances de ser usado novamente, do que um dado usado há mais tempo. Isso é verdade porque as variáveis de um programa tendem a ser acessadas várias vezes durante a execução de um programa, e as instruções usam bastante comandos de repetição, o que faz instruções serem acessadas repetidamente. Sendo assim, o Sistema de Memória tende a manter os dados e instruções recentemente acessados no topo da Hierarquia de Memória.

d) Localidade Espacial

R: O Princípio da Localidade Espacial diz que há uma probabilidade de acesso maior para dados e instruções em endereços próximos àqueles acessados recentemente. Isso também é verdade porque os programas são sequenciais e usam de repetições. Sendo assim, quando uma instrução é acessada, a instrução com maior probabilidade de ser executada em seguida, é a instrução logo a seguir dela. Para as variáveis o princípio é semelhante. Variáveis de um mesmo programa são armazenadas próximas uma às outras, e vetores e matrizes são armazenados em sequência de acordo com seus índices. Baseado neste princípio, o Sistema de Memória tende a manter dados e instruções próximos aos que estão sendo executados no topo da Hierarquia de Memória.