MO401 – Arquitetura de Computadores

Trabalho1

Título: "Balanced Cache: Reducing Conflict Misses of Direct-Mapped Caches through Programmable Decoders"

Referência: Zhang, C. "Balanced Cache: Reducing Conflict Misses of Direct-Mapped Caches through Programmable Decoders". International Symposium on Computer Architecture, June 2006, Boston. Pp. 155-166.

Aluna: Gizelle Sandrini Lemos – RA:066109

Resumo:

O aumento da diferença entre a latência para acesso à memória e a velocidade do processador, atualmente é considerado o gargalo mais crítico para que se alcance um sistema computacional de alto desempenho. Para contornar esse problema, arquiteturas que utilizam hierarquia de memória com vários níveis têm sido empregadas. Nesse artigo é analisado o acesso à memória cachê de primeiro nível e proposta uma nova arquitetura para o acesso à memória cachê, combinando algumas vantagens do acesso diretamente mapeado e do acesso associativo de conjunto.

Um cachê com acesso diretamente mapeado é mais rápido e consome menos energia do que um cachê associativo, devido à simplicidade de sua implementação em termos de hardware. Em contrapartida o cachê diretamente mapeado possui uma taxa de falha maior que o cachê associativo, dependendo do padrão de acesso à memória da aplicação que está sendo executada.

Um cachê associativo de conjunto tem duas vantagens em relação ao cachê diretamente mapeado: a redução de falhas por conflito, por possuir mais opções para escolha do bloco a ser substituído (vítima) e a adoção de uma política de troca que permite escolher o melhor bloco a ser substituído com base no histórico de acesso.

Idealmente, um modelo perfeito deveria ter o tempo de acesso de um cachê diretamente mapeado com a taxa de falhas inferior de um cachê associativo.

O artigo considera que, em geral, o acesso à memória executado por grande parte dos aplicativos é extremamente desbalanceado, sendo que algumas áreas da memória são muito mais acessadas que outras e propõe um mecanismo implementado inteiramente em hardware que provê o benefício do mecanismo de trocas, enquanto mantêm o tempo de acesso constante de um cachê mapeado diretamente. O mecanismo é chamado de Cachê Balanceado ou simplesmente B-Cachê. Em resumo, a proposta consiste em adicionar/alterar a memória diretamente mapeada em três pontos:

1. Aumentar o tamanho do endereço de índice de três bits para quatro bits onde os dois bits menos significativos são usados por um decodificador não programável e os dois bits mais significativos são usados por um decodificador programável. A saída dos dois decodificadores é usada para definir a posição do dado na memória. Através dos decodificadores é possível determinar os conjuntos mais acessados de maneira dinâmica. Por conseqüência, o espaço de endereçamento de memória é mapeado em conjuntos menores de cachê. Isso é chamado de mapeamento de memória limitado;
2. Implementar uma política de trocas capaz de detectar os conjuntos de memória menos utilizados;
3. Utilizar um decodificador programável permitindo que, na ocorrência de uma falha de leitura, o B-Cachê determine dinamicamente qual endereço de memória será mapeado para um conjunto de cachê.

A idéia fundamental do B-Cachê é reduzir a taxa de falhas (miss) balanceando o acesso às posições do cachê.

Em testes simulados, foi demonstrado que o modelo proposto é capaz de reduzir significativamente a taxa de falhas em relação aos modelos de cachê anteriores. Mesmo tendo um consumo de energia cerca de 10% maior que a memória mapeada diretamente, B-Cachê consegue no total uma economia de 2% devido à redução do número de falhas e tempo de execução das aplicações. Adicionalmente o modelo B-Cachê tem o mesmo tempo de acesso que um cachê mapeado diretamente tradicional.