Lista de Exercício – Memória Cache – Infra-Estrutura de Hardware

Grupo:

lab7

sgfl

lccao

vlma

1. A matriz é linearizada em função das linhas, representando-as de forma contígua. Por isso a leitura de linhas completas aproveita melhor da localidade espacial.

1. a) 2 bits de deslocamento de byte. 1 bit de deslocamento de blocos. 4 bits de índice.   
   e 25 bits de tag.  
     
   b) O sistema tem que buscar o bloco na memória principal que contém o endereço procurado e o colocar na cache.  
     
   c) A cache fornece os dados para o processador.  
     
   d) Identifica se a entrada indicada pelo índice é correspondente ao bloco desejado.
2. a) Tamanho da cache = número de entradas \* tamanho do slot, representa o tamanho de dados que pode ser armazenadas em cache, quanto maior o tamanho menor a chance de ter cache miss, porém mais cara será.   
   Tamanho de Bloco da Cache representa a quantidade de words que um bloco possui. Se ela for maior, mais a localidade espacial será aproveitada. Porém aumenta a penalidade.

Número de conjuntos em cache representa a associatividade da cache, que mostra quantos slots de memórias estão em um dado conjunto. Isso diminui a necessidade de substituição constante, por diminuir os conflitos. Melhorando o índice de cache miss, mas aumenta o custo.

Políticas de Escrita: podem ser write-through ou write-back. Na write-through todas as alterações realizadas na cache devem ser imediatamente guardada na memória o que faz com que o processador perca tempo, para minimizar isso utilizamos um write buffer. Na write-back existe a memória apenas é alterada quando a cache é substituída, sendo necessário a adição de um bit indicando se o valor foi modificado, o que aumenta o tamanho do cache e consequentemente seu custo.

Política de Substituição: podem ser randômico, LRU ou MRU, ela representa a forma de substituição de cache que faz com que se faça menos substituições de slots.  
  
b) Apenas na primeira iteração haverá falha na cache. Uma vez que todos os blocos utilizadas no loop estiverem na cache, não existirá mais falhas durante o loop. O que é bom, porque o loop é repetido muitas vezes, e se a cada iteração fosse necessário fazer uma busca na memória principal o tempo de execução seria muito maior.

c) Deve apresentar muita localidade espacial, já que por carregar grandes blocos, instruções e dados próximos vão ser carregados juntos. Então o código deve ir lendo os blocos de uma linha de forma consecutiva, e uma linha após a outra, dada a representação da matriz que é utilizada na memória.

4) A matriz analisada no código é armazenada na memória seguindo a ordem de suas linhas, ou seja, elementos que estão adjacentes na mesma linha da matriz também estarão adjacentes na memória. Elementos de colunas diferentes estarão, em geral, mais distantes.

SumByColRow (Soma por colunas): A ordem de acesso não é em posições contíguas na memória, já que ocorre seguindo a sequência de colunas da matriz.

SumByRowCol (Soma por linhas): A ordem de acesso é em posições contíguas da memória, já que ocorre seguindo a sequência de linhas da matriz.

Sim, existe diferença entre os acertos de cada cache. Essa diferença é justificada pela exploração da localidade espacial. A função SumByRowCol explora bem essa propriedade, o que diminui a taxa de falta de dados, pois os elementos próximos, que serão acessados pela função, já estarão, na maioria das vezes, presentes na cache.