

SSC0902 – Organização e Arquitetura de Computadores

Lista de Exercícios sobre Hierarquia de Memória

1. Qual a principal razão de se ter o sistema de memória do computador organizado de maneira hierárquica?
2. Uma cache pode ser organizada de tal modo que se tenha:
 - Mais palavras por bloco, mas poucos blocos
 - Menos palavras por bloco, mas muitos blocos

Considere que nas duas opções a capacidade da cache (em bytes) é a mesma. Quais as vantagens e desvantagens de cada uma das abordagens? Dê um exemplo de sequência de acesso que tem uma taxa de acerto melhor em cada uma das opções.

3. Para cada uma das caches definidas abaixo, faça a divisão do endereço nos campos necessários para os mapeamentos direto, associativo e associativo por conjunto ($k = 4$). Considere que a memória principal é endereçada a Byte e que endereça 2^{32} Bytes, portanto, são necessários 32 bits para endereçar qualquer posição da memória. Considere também que uma *word* (palavra) tem 4 Bytes.
 - a) Cache com 16 words, 1 word por bloco (portanto, 16 blocos)
 - b) Cache com 32 words, 1 word por bloco (portanto, 32 blocos)
 - c) Cache com 64 words, 1 word por bloco (portanto, 64 blocos)
 - d) Cache com 128 words, 2 words por bloco (portanto, 64 blocos)
 - e) Cache com 128 words, 4 words por bloco (portanto, 32 blocos)
 - f) Cache com 1024 blocos, 8 words por bloco (portanto, 8192 words)
 - g) Cache com 256 blocos, 2 words por bloco (portanto, 512 words)

4. Após fazer a divisão do endereço, considere a cache (a) definida no exercício anterior e calcule em que posição da cache será mapeada cada uma das seguintes palavras de memória (não se esqueça que a memória é endereçada a byte, então tem que converter cada endereço de palavra em um endereço de byte).

- 2, 3, 11, 16, 21, 13, 64, 48, 19, 11, 3, 22, 4, 27, 6 e 11

5. Realize os acessos abaixo na hierarquia de memória, exatamente na ordem em que eles aparecem. Os **endereços para acesso estão a Byte**. O computador trabalha com **palavras de 32 bits** (4 Bytes). Suponha que o conteúdo da palavra de memória é o endereço do 1º byte daquela palavra multiplicado por 10 (Ex: a palavra começando no byte 8 tem o conteúdo 80 e a palavra começando no byte 20 tem o conteúdo 200). Há dois níveis de cache (L1 e L2) e a memória RAM. A **cache L1 é totalmente associativa e tem 2 blocos. Cada bloco tem 2 palavras**. A **cache L2 tem 2 conjuntos, é associativa por conjunto com 2 blocos por conjunto e cada bloco tem 4 palavras**. A memória **RAM tem 512 Bytes** (não é necessário endereçar mais do que essa quantidade de memória). A política de substituição nas caches

L1 e L2 é LRU (use 1 para o mais recentemente utilizado e 0 para o menos). As políticas de atualização são *write-back* e *write allocate* para as duas caches (use 1 para modificado e 0 para não modificado). Os acessos abaixo podem ser Leitura ou *Escrita*. Quando o acesso for *Escrita*, o conteúdo a ser escrito na palavra é o endereço do 1º byte. Quando o grau de associatividade da cache for maior do que um, insira primeiro nos últimos blocos do conjunto.

- Represente graficamente as caches L1, L2 e a memória RAM
- Indique a divisão do endereço da memória principal para cada cache
- Faça a simulação e indique a situação final dos conteúdos na L1, L2 e RAM.
- Indique para cada cache: quantidade de acessos, acertos e faltas.

São acessadas palavras que começam nos seguintes endereços, **já em bytes**:

Acesso	Tipo	End Dec	End Hexa	Endereço em Binário – Cache L1	Endereço em Binário – Cache L2
1	Escrita	04	04		
2	Escrita	12	0C		
3	Leitura	24	18		
4	Leitura	16	10		