## **NATIONAL UNIVERSITY OF SINGAPORE**

## **Department of Electrical and Computer Engineering**

## **CG2028 Computer Organization**

**Tutorial 5 Solutions**

**Cache Memory Principles**

1. A computer system has 3 devices in the memory hierarchy – cache, RAM and Solid State Drive (SSD), having access times of 10 ns, 1 μs and 100 μs respectively. The access frequencies of two of the devices (not in any particular order) are 0.9 and 0.001. Compute the average access time for the memory system.

Ans:

Access frequencies should add up to 1 and should decrease at higher levels. Hence, access frequency of Cache = 0.9, SSD = 0.001. So the RAM should have an access frequency of 0.099.

Average access time = 10\*0.9+1000\*0.099+100000\*0.001 = 9+99+100 = 208 ns.

1. A computer system has main memory capacity of 8192 blocks whereas the cache can hold 256 blocks, organized into 64 sets. What is the maximum number of comparisons required before the cache controller can determine whether the data addressed by the CPU is present in the cache? How many bits will need to be stored as overhead for the tag field in each cache block?

Ans:

256 blocks are organized into 64 sets. The cache is 4-way set-associative, and hence we need 4 comparisons to check if a particular block is present within the set it is mapped to. The tag filed requires log2(8192/64) = 7 bits. Additional overhead bits such as valid bit and dirty bit (if using write-back policy) will be required for each block. Note that even more overhead bits may be required for the replacement algorithm to compute the ‘age’ of the block to make decisions regarding whether a particular block is a good candidate block to be replaced.

1. A cache can hold 3 blocks and is fully associative. Consider a request sequence as follows – 1, 2, 1, 2, 3, 3, 4, 1, 5, 2, 4, 2, 5. Draw a diagram that identifies the block to be replaced whenever a miss occurs, assuming an optimal replacement algorithm is used.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 |  | 1 |  | 1 |  | 1 |  | 1 |  | 1 |  | 1 |  | 1 |  | 5 |  | 5 |  | 5 |  | 5 |  | 5 |
|  |  | 2 |  | 2 |  | 2 |  | 2 |  | 2 |  | 2 |  | 2 |  | 2 |  | 2 |  | 2 |  | 2 |  | 2 |
|  |  |  |  |  |  |  |  | 3 |  | 3 |  | 4 |  | 4 |  | 4 |  | 4 |  | 4 |  | 4 |  | 4 |

Ans:

LRU or FIFO would have led to higher misses. For example, to accommodate block 4, FIFO and LRU would have replaced block 1, causing a miss at the next request.

1. Which mapping strategy is used for level 1 cache by the latest processors from Intel? (not examinable)

Ans:

Recent Intel processors (e.g., Coffee Lake microarchitecture) use two (per core, one for instruction and one for data) 8-way set associative caches of size 32KB, with block size of 64 bytes.

Zen 2 microarchitecture from AMD has the same configuration too.

<https://en.wikichip.org> is an excellent resource for information on various microprocessors.

Associative mapping is used only for very small caches (e.g., some [TLBs](https://en.wikipedia.org/wiki/Translation_lookaside_buffer)). Most caches are set associative; typically, 4-way, 8-way or 16-way.