Chihsuang Wang 101-64106

**Question 2:**

**Reference:**

http://courses.cs.washington.edu/courses/cse378/02sp/sections/section9-2.html

1. Compulsory misses

Increase Cache Size: No changes or Getting worse.

Increase Block Size: Cache miss rate improved.

Increase Associativity: No Changes.

Compulsory misses will go up as there are more blocks to miss on in a cold cache. The limit case is a cache with a single block - you will only get one compulsory miss. Increasing the block size means more adjacent words will be fetched on each miss, so references to these words will not cause compulsory misses - this exploits spatial locality. Associativity only affects how cache blocks are arranged, not how they are fetched from main memory, so will not affect compulsory misses.

1. Capacity misses

Increase Cache Size: Cache miss rate improved.

Increase Block Size: No Changes.

Increase Associativity: No Changes.

Capacity misses are not really affected by block size because the decrease in the number of blocks that can be held is offset by their increased size. Associativity has no effect on capacity misses as the total number of blocks remains the same no matter what the associativity.

c) Conflict misses

Increase Cache Size: No Changes.

Increase Block Size: No changes or Getting worse.

Increase Associativity: Cache miss rate improved.

Conflict misses are not affected by cache size since conflict misses arise from blocks from main memory mapping to the same position in the cache, which is mostly independent of the cache size. Increasing the block size may increase the number of conflict misses. There is a greater chance to displace a useful block from the cache.

**Question 3:**

**Reference:**

<http://web.cs.wpi.edu/~cs3013/a06/week4-paging.pdf>

https://www.cs.utah.edu/~mflatt/past-courses/cs5460/lecture8.pdf

Physical addresses refer to hardware addresses of physical memory. Virtual addresses refer to the virtual store viewed by the process. Virtual addresses might be the same as physical addresses or might be different, in which case virtual addresses must be mapped into physical addresses. Mapping is done by Memory Management Unit (MMU) from the OS. Virtual space is limited by size of virtual addresses (not physical addresses). Virtual space and physical memory space are independent.

The same virtual address in two different processes does not map to the same physical address. Some virtual addresses do not map to any physical address.

**Question 5:**

**Reference:**

<https://zh.scribd.com/doc/61748872/8/Merging-Write-Buffer-to-Reduce-Miss-Penalty#page=12>

Textbook

• Merging write buffers

Reduce Miss Penalty - Write buffer to allow processor to continue while waiting to write to memory. If buffer contains modified blocks, the addresses can be checked to see if address of new data matches the address of a valid write buffer entry. If so, new data are combined with that entry. Increases block size of write for write-through cache of writes to sequential words, bytes since multiword writes more efficient to memory.

• Victim buffer

A victim buffer is a type of write buffer that stores dirty evicted lines in [write-back](https://en.wikipedia.org/wiki/Write-back) caches so that they get written back to main memory. Besides reducing pipeline stall by not waiting for dirty lines to write back as a simple write buffer does, a victim buffer may also serves as a temporary backup storage when subsequent cache accesses exhibit [locality](https://en.wikipedia.org/wiki/Locality_of_reference), requesting those recently evicted lines, which are still in the victim buffer.

• Compiler-based prefetching

Reduce Miss Penalty or Miss Rate - insert prefetch instructions to request data before the processor needs it. There are two flavors of prefetch:

1. Register prefetch will load the value into a register.

2. Cache prefetch loads data only into the cache and not the register.

**Question 6:**

**Reference:**

<http://www.eecs.berkeley.edu/~kubitron/courses/cs252-S07/lectures/lec12-vector2.pdf>

<https://www.cs.umd.edu/class/fall2001/cmsc411/proj01/cache/vector.pdf>

• Explain what strip-mining is.

A technique to split the long vector into multiple vectors (of equal, or of maximum plus smaller lengths) with vector processors.

• Explain why it is used.

Vector lengths do not often correspond to the length of the vector registers. For shorter vectors, we can use a vector length register applied to each vector operation. For longer vectors we can use strip-mining technique. In short, vector registers have finite length, so we break loops into pieces that fit into vector registers.