BLG 322E COMPUTER ARCHITECTURE FINAL EXAM SOLUTIONS

QUESTION #1:

A. Note that both caches have same capacity. While accepting other reasonable, meaningful answers, I was mainly looking for the following:

Option #1 (smaller block size)
- Less conflict misses
- Benefits less from spatial locality, more cold misses
- Beneficial for irregular (scattered) accesses

Option #2 (larger block size)
- Benefits more form spatial locality, expected less cold misses
- More conflict misses
- Beneficial for sequential (regular) accesses

B. Following are given:
- Direct mapped (We can think of a direct mapped cache, an set associative cache with 1 way)
- 12 bits for tag
- 24 bits address
- Word size 2 Bytes and word accessible, this means accesses must be word-aligned (to the even byte numbers)
- Block size 16 Bytes (that is 8 words)

Since we have 16 Bytes in the block 4 bits in the address identify the Byte in the block, but 3 of these 4 bits are used as Word-offset.
Total number of bits for set and offset = 24 – 12 = 12
Total number of block bits = 12 – 4 = 8
In this cache, we have 2^4 = 256 blocks in Total

But, I accepted the answer of 512 blocks, too.

C. Following are given:
- 16-way set associative
- Words are 32 Bits (4 Byte), half-word accessible
- 4 Mbytes (= 2^{22}) of capacity
- Block size 32 (2^5) Words
- Addresses are 32 Bits

Block Size $2^5 * 2^2 = 2^7$ Bytes in total

$2^{22} / 2^7 = 2^{15}$ Blocks in the cache
Since, we have 16 Blocks in each set;
2^{15} divided by 2^4 = 2^{11} Sets that we have, And we need 11 Bits for Sets

We need 5 Bits for Word offset, 1 Bit for Half Word, total, we need 6 Bits offset to access any half word aligned address.

Even though, we do not have Byte access, 1 Bit is used for Byte address in the Block.

Then, Total 7 Bits are used for addressing any Byte in a Block.

Tag Bits = 32 – (11 + 7) = 14 Bits

Tag Bits = 14
Set Bits = 11
Offset bits = 6

QUESTION #2:

a) P0's Frame 0's coherency state changes from E to M.
b) Frame 1 on P0 does not get invalidated, since it has a data block with a different address.
c) It is cached in S state.

QUESTION #3:

a) What is the average time to serve one I/O request for this system?

10000 Rotations -> 60 sec

1 Rotation -> 60/10000 = 6 ms

Rotational Latency for a sector = 6/2 = 3ms

1 rotation covers 1 track, 1 track has 600 sectors, therefore;

Transfer rate for a sector = 6/600 = 1/100 ms

Seek time = 6ms

Every sector requires a seek time!

Total time needed for an I/O request = 100 (number of sectors) * (6 (seek time) + 3 (ave rotational latency of sector) + 1/100 (transfer time of sector))

Total time needed for an I/O request = 901 ms => it is around 0.9 s

b) What is the maximum number of I/O operations per second for the system?

I/O per second for a single disk ~= 1/0.9
Since we have 20 disks, total IOPS = 20 * 1/0.9 ~= 22.2