HW #1

1 a) Write-through policies take longer write times.
True

b) Larger cache block sizes reduce the miss penalty.
False, it increases miss penalty as it takes longer to replace larger blocks of memory in cache.

c) Higher associativity means higher conflict misses.
False, it means lower conflict misses, but higher hit times.

d) Miss penalties are always lesser than hit times.
False, they are higher than hit times or we wouldn't have the cache.

e) The choice among different types of mapping in memory hierarchy depends on the cost of miss vs cost of implementing associativity.
True

→

HW #1 (cont.)

2.20) a) With critical word first, it would take 120 cycles as it would get the data necessary as the first 16-byte word. Without critical word first, to guarantee the data in cache will take 120 cycles plus 3 more sets of 16 cycles, giving us 168 cycles.

b) Depending on how much the L1 or L2 caches misses add to the average memory access time, the one that has a bigger impact on AMAT is the one that CW first and early restart would matter more for, and therefore would be more important to implement in that level of cache.

2.21) a) Each write buffer entry size should match the L2 write data bus and be 16B wide.

HW #1 (cont.)

2.2.21) b) If the write buffer is 16B as decided on in part a, it would take a merging write buffer entry would take 2 cycles of storing to empty. If a non-merging write buffers used it would take 4 cycles since the buffer wouldn't have more than 8B per entry. Therefore the speedup is 2

{ write to cache address of 64-bit/8B stores takes 1 cycle.

c) For a non-blocking cache, writes can still happen during the processing of a miss so it could mean less entries are needed than in a blocking cache, which stops all execution progress, which means it doesn't matter how many entries are required in the buffer. If execution progress is stopped like in a blocking cache, nothing can be written or processed in the write buffer.

(B)

HW #1 (cont'd)

2 2.22) MPKI = 20 cache/latency/MPKI
32 KB/1/100 128 KB/2/80
512 KB/4/50 2 MB/8/40 8 MB/16/10
off-chip access time = 200 cycles

a) for 32 KB, 1 clock cycle hit time, 10% miss rate
for 8 MB, 16 clock cycle hit time, 1% miss rate
off chip memory, 200 cycle access time

$$AMAT = \overbrace{1}^{L_1 \text{ hit}} + \overbrace{(0.1(\overbrace{16}^{L_2 \text{ hit}} + \overbrace{(0.01(\overbrace{200}^{\text{mem}}))}^{L_2 \text{ miss}}))}^{L_1 \text{ miss}}$$

$$\boxed{AMAT = 2.8 \text{ clock cycles}}$$

b) 32 KB L_1, 512 KB L_2 (4 cycles, 5% miss), 8 MB L_3, off-chip

$$AMAT = \overbrace{1}^{L_1 \text{ hit}} + (0.1(\overbrace{4}^{L_2 \text{ hit}} + 0.05(\overbrace{16}^{L_3 \text{ hit}} + 0.01(\overbrace{200}^{\text{mem}}))))$$

L_1 miss; L_2 miss; L_3 miss

$$\Rightarrow \boxed{AMAT = 1.49 \text{ clock cycles}}$$

c) 32 KB L_1, 128 KB L_2 (2 cycles, 8% miss), 512 KB L_3, 8 MB L_4, off-chip

$$AMAT = 1 + (0.1(2 + 0.08(4 + 0.05(16 + 0.01(200)))))$$

$$\Rightarrow \boxed{AMAT = 1.2392 \text{ clock cycles}}$$

HW #1 (cont'd)

3 B.1) cache hit, 1 cycle cache miss, 110 cycles
main mem w/ cache disabled, 105 cycles

a) AMAT = 1 + 0.03(110) * only depends on cache miss & cache hit here

⇒ **AMAT = 4.3 clock cycles**

b) if truly random, miss rate is 99.993%

AMAT = 1 + 0.99993(110)

⇒ **AMAT = 110.993 clock cycles.**

c) The role of locality in choosing to use cache memory is extremely important as, if memory accesses are truly random, then a cache is virtually useless as its AMAT will be longer than a direct memory access.

d) 105 = 1 + x(110)

x = 0.94545 ∴ highest miss rate before a cache is disadvantageous is 94.545%

⇒

HW #1 (cont'd)

3 B.2) 512B cache w/ 8, 64B blocks
Main mem is 2KB w/ 32, 64B blocks

a)

Cache Block	Set	Way	Possible Mem Blocks
0	0	0	Any
1	0	1	Any
2	0	2	Any
3	0	3	Any
4	0	4	Any
5	0	5	Any
6	0	6	Any
7	0	7	Any

b)

Cache Block	Set	Way	Possible Mem Blocks
0	0	0	M0, M2, M4, M6, ..., M30
1	0	1	M0, M2, M4, M6, ..., M30
2	0	2	M0, M2, M4, M6, ..., M30
3	0	3	M0, M2, M4, M6, ..., M30
4	1	0	M1, M3, M5, M7, ..., M31
5	1	1	M1, M3, M5, M7, ..., M31
6	1	2	M1, M3, M5, M7, ..., M31
7	1	3	M1, M3, M5, M7, ..., M31

B.7) The best way to simplify this is to only check the address bits that would indicate which set in L2 that the write would go to, because it is faster than checking every slot to write to. Lower set associativity is good in this case, but a full write buffer would remove the issue in the case of higher set associativity.