CSCI 6461: Computer System Architecture – Homework 3

Shubham Jadhav | G30570862

2.20.a)

To find: cycles to service an L2 cache miss

For critical word first & early restart strategy: total cycles to service an L2 cache miss would be **120 cycles**, as this strategy request the missed word first from memory and only wait for first block of L2 and send it to the processor as soon as it arrives.

For without critical word first & early restart strategy: it waits for the entire 64 byte block. So, the first block of 16 byte would requires 120 cycles and for each of the next 3 blocks would require 16 cycles. Therefore, total cycles: 120+(16*3) = **168 cycles**.

2.20.b)

Both critical word first and early start techniques only benefit when cache blocks are large enough. Their benefit is dependent on block size and the likelihood of another access to the portion of the block that is yet to be fetched. These techniques are implemented to reduce miss penalty; therefore, the impact will depend on level 1 and level 2 misses and how much they contribute to Average Memory Access Time. Since level 2 caches work at a better rate than level 1 due to its large block size, we can say that critical word first and early start techniques might have impact on performance of L2 caches. If both critical word first and early restart offer a similar percentage reduction in miss service durations for both level-1 and level-2 miss service, and level-1 misses contribute more to AMAT, critical word first would likely be more significant for level-1 misses.

2.22)

To calculate cycles for each case

Time spend = Σ (MPKI * Latency of next cache in hierarchy)

a) 32 KB L1; 8 MB L2; off-chip memory : (100*16) + (10*200) = **3600 cycles**

b) 32 KB L1; 512 KB L2; 8 MB L3; off-chip memory : (100*4) + (50*16) + (10*200) = **3200 cycles**

c) 32 KB L1; 128 KB L2; 2 MB L3; 8 MB L4; off-chip memory :
(100*2) + (80*8) + (40*16) + (10*200) = **3480 cycles**

We can observe that, a cache hierarchy that is too shallow or too deep need a greater number of cycles. Increasing hierarchy may result in unnecessary lookups. Amongst all the three design, option b seems to the better as it requires least cycles to access the cache hierarchy.