Noel Ball
HW3

Question 1:
Ai.) With 1 GHz processor operations (1 ns cycle time) and 40% of operations access memory with 100 ns latency, the additional cycles due to memory stage latency $= .40 \times 100 = 40$ cycles. Slowdown $= \frac{NewCPI}{BaseCPI} = \frac{41}{1} = 41$x.

Aii.) With 1 Ghz processor operations (1 ns cycle time) and all operations fetch from memory with 100 ns latency, the additional cycles due to fetch stage latency $= 100$ cycles.
Slowdown $= \frac{101}{1} = 101$x.

Bi.) Number of blocks $= \frac{Cache\ size}{Block\ size} = \frac{32 \times 1024}{64} = 512$ blocks.
Number of sets $= \frac{Blocks}{Associativity} = \frac{512}{4} = 128$ sets.
Offset bits $= \log_2(Block\ size) = \log_2(64) = 6$ bits.
Index bits $= \log_2(Sets) = \log_2(128) = 7$ bits.
Tag bits $= Address\ size - (Index + Offset) = 32 - (7 + 6) = 19$ bits.
0xcafebebe → 11001010111111101011111010111110.

Bii.) Number of blocks $= \frac{Cache\ size}{Block\ size} = \frac{128 \times 1024}{8} = 16384$ blocks.
Number of sets $= 1$ set, blocks can be placed anywhere since fully associative.
Offset bits $= \log_2(Block\ size) = \log_2(8) = 3$ bits.
Index bits $= \log_2(Sets) = \log_2(1) = 0$ bits, not necessary since fully associative.
Tag bits $= Address\ size - (Index + Offset) = 32 - (0 + 3) = 29$ bits.
0x28288282 → 00101000001010001000001010000010.

Question 2:
A.) Hit rate access #1:
The first time an element in a block is accessed it results in a compulsory miss because the block must be fetched, however subsequent accesses to the other integers hit since they are within the same 16-byte block that is loaded into cache from the row they are on. Since each block can hold 4 integers, ¾ accesses hit with an approximate 75% hit rate.

B.) Hit rate access #2:
The first time an element in a block is accessed it results in a compulsory miss because the block must be fetched, however subsequent accesses to the other integers are also going to miss since they are an entire row apart from the previously cached data, with an approximate 0% hit rate.

C.) Access #1 has strong spatial locality since it accesses matrix elements that are adjacent in memory, significantly contributing to the effectiveness of the cache.

Question 3:
A.) Average memory access time $= Hit\ Time + (Miss\ Rate \times Miss\ Penalty) =$ $4 + (0.25 \times 28) = 4 + 7 = 11$ cycles

B.) Average memory access time $= L1\ Hit\ Time + (L1\ Miss\ Rate \times (L2\ Access\ Time$ $+ (L2\ Miss\ Rate \times Memory\ Access))) = 2 + (0.25 \times (8 + (0.5 \times 80)))$ $= 2 + (0.25 \times (8 + (0.5 \times 80))) = 14$ cycles.

C.) Without a dirty bit, the cache management system would lack crucial information about which cache blocks have been modified. Trying to create a workaround without a dirty bit would cause several problems with inefficiency, performance penalties, and increased wear on the physical memory hardware.

D.) Workloads with high temporal and spatial locality in their write patterns benefit more from write allocate, minimizing the cost of subsequent cache misses. However, workloads with little access to recent writes can benefit from a no write allocate policy to keep cache available for other data.

Question 4:
Ai.) Hit rate typically starts increasing as block size increases, reaches an optimal point, and then decreases if the block size becomes too large. The optimal point is where the cache best balances the spatial locality against the costs of reduced cache capacity.

Aii.) AMAT decreases as the cache capacity grows, and flattens out after the capacity supports more frequently accessed data being accommodated within the cache.

$Aiii$.) The miss rate rapidly decreases as associativity increases from a low level, since higher associativity reduces conflict misses, then gradually plateaus since the most significant reductions in conflict misses are achieved with initial increases in associativity.

B.) The miss rate continues to increase as STRIDE increases, very sharply once the STRIDE value surpasses the number of integers that can be stored in a single cache block, since each subsequent access will lack cached data due to spatial locality.