Lecture 12 Activity Solution

Model 1: Caches

1. Most convenient: backpack. Least convenient: parent's house.
2. Parent's house can hold the most books and notes.
3. You might pick the most important books and notes, or the books and notes related to the classes you plan to attend and the homework you plan to do.
4. You might move the outdated notes and textbooks for classes you know you are finished with to your parent's house.

Model 2: Lookup

1. 0x00 00 FF 50
2. The last (least-significant) byte.
3. You might use the lower bits, knowing that while related data tends to share the upper address bits, that means that related data is likely to hash to the same place, resulting in more conflicts when the data accesses are close together. Using the lower bits makes related data hash to more different buckets.
4. The high order bits often indicate whether the memory address lies in the stack, the heap, or the global variables, because the memory regions corresponding to the stack, heap, and global variables are separated in memory. The location of a memory access can be useful to recognize when debugging.

Model 3: Hardware

1. s bits are required for the set index bits, if there are $S = 2^s$ sets.
2. b bits are required for the block offset bits, if each block has $B = 2^b$ bytes.
3. m - s - b bits are in the tag.
4. (a) Fully associative, because a fully associative cache requires more hardware for additional tag comparisons.
   (b) Direct mapped, because a direct mapped cache does not have to check whether every line is the line being accessed.
   (c) Fully associative, because data can be stored in any cache block, not just the one corresponding to the set bits.

Model 4: Replacement

1. Answers may vary.
2. Answers may vary.
3. A(miss), B(miss), A(hit), C(miss), B(miss), C(hit), A(miss).
4. A policy that evicts the most recently used data could have done better: A(miss), B(miss), A(hit), C(miss), B(hit), C(hit), A(miss).
5. The cache should discard the evicted line.

Model 5: Writing to Cache Lines

1. When a processor writes to a cache line, the data comes from the cache and other memory systems.
2. The cache should not immediately update memory—there is temporal locality.
3. In a write-back cache, the cache must cache the new value. When the line is evicted, the cache must write the line to memory. The cache needs a dirty bit to indicate whether a cache line has been written to.
4. If the cache allocates a cache line for the new data upon a write miss (WBWA), the writes would be faster and multiple writes in a single block would require only one write back to main memory. If the write skips the cache and goes directly to main memory (WTWNA), it would be easier to implement and the data in main memory is more consistent, but writes would be slower and we would not be able to take advantage of spatial locality to reduce the number of writes.
5. See above.