Timothy Holmes (username: THOLME15)

Attempt 3

Written: Sep 15, 2020 7:21 PM - Sep 15, 2020 7:42 PM

Submission View

Released: Sep 7, 2020 10:15 PM

Question 1 — 1 / 1 point

Which statement is *incorrect?*

A. SRAM is much more expansive than DRAM per bit.

B. SRAM is about 10 times faster than DRAM.

C. SRAM requires more transistors per bit than DRAM.

D. DRAM requires constant rewrite (*refresh*) of the data, but SRAM does not.

E. None of the above, they are all correct.

- ( ) Answer A.
- ( ) Answer B.
- ( ) Answer C.
- ( ) Answer D.
- ✓ ( ) Answer E.

View Feedback

Question 2 — 1 / 1 point

Rank these storage types by read speed, from fastest to slowest.

A. DRAM, SRAM, SSD, rotating disk.

B. SRAM, DRAM, rotating disk, SSD.

C. SRAM, SSD, DRAM, rotating disk.

D. SRAM, DRAM, SSD, rotating disk.

E. DRAM, SSD, SRAM, rotating disk.

- Answer A.
- Answer B.
- Answer C.
- ✓ Answer D.
- Answer E.

View Feedback

Question 3

0 / 1 point

ROM are used for (or in):

A. Solid state disks.

B. Firmware programs that are unlikely to change.

C. Disk caches.

D. All of the above.

E. None of the above.

- Answer A.
- ✗ Answer B.
- Answer C.
- ➡ Answer D.

Answer E.

View Feedback

Question 4 1 / 1 point

What do RAM and ROM stand for?

A. Random Access Memory and Random Order Memory.

B. Read Allowed Memory and Random Only Memory.

C. Random Access Memory and Read Only Memory.

D. Read Access Memory and Read Order Memory.

Answer A.

Answer B.

✓ Answer C.

Answer D.

View Feedback

Question 5 1 / 1 point

Reading "$>$" as "consists of", which inclusion sequence is correct for common disks?

A. Disk $>$ surfaces $>$ platters $>$ tracks $>$ sectors.

B. Disk $>$ platters $>$ surfaces $>$ tracks $>$ sectors.

C. Disk $>$ platters $>$ tracks $>$ surfaces $>$ sectors.

D. Disk $>$ platters $>$ sectors $>$ surfaces $>$ tracks.

E. Disk $>$ surfaces $>$ platters $>$ sectors $>$ tracks.

Answer A.

✓ Answer B.

Answer C.

Answer D.

Answer E.

View Feedback

Question 6 1 / 1 point

What is *not* part of a common rotating disk drive?

A. A spindle.

B. A processor.

C. A mechanical read/write head.

D. Some DRAM.

E. Some ROM.

F. None of the above, they are all common on hard drives.

Answer A.

Answer B.

Answer C.

Answer D.

Answer E.

✓ Answer F.

View Feedback

Question 7 1 / 1 point

The access time in a disk is the sum of:

A. seek time, execution time, transfer time.

B. seek time, loading time, execution time.

C. wait time, rotational latency, transfer time.

D. wait time, loading time, execution time.

E. seek time, rotational latency, tranfer time.

- ◯ Answer A.
- ◯ Answer B.
- ◯ Answer C.
- ◯ Answer D.
- ✓◯ Answer E.

▷ View Feedback

Question 8 1 / 1 point

When the CPU requests information from the disk, how does it wait for the information to arrive?

A. It probes the disk every N cycles to check if the information is ready.

B. It does not wait, it will receive a message when the information is ready in memory, and can do something else in the meantime.

C. It does not have to wait, since the disk is fast enough.

D. It stops executing instructions until the information is available.

- ◯ Answer A.
- ✓◯ Answer B.
- ◯ Answer C.

Answer D.

View Feedback

Question 9 1 / 1 point

How does the CPU know in which surface/track/sector is the data it is searching for?

A. It precomputes a transaction table that is stored in the cache.

B. It does not; the disk provides an abstract view as a 1-dimensional array of sectors.

C. It asks an extra component on the CPU chip that is designed to find that information.

D. It does not; requests to the disk are made using file names.

Answer A.

✔ Answer B.

Answer C.

Answer D.

View Feedback

Question 10 1 / 1 point

Locality is the idea that programs tend to use data and instructions with addresses near or equal to those they have used recently. What are the definitions of *temporal* and *spatial* locality?

A. Temporal: items that have been recently referenced tend to be re-referenced. Spatial: items with nearby addresses tend to be referenced close together in time.

B. Temporal: the less local, the more time a program needs. Spatial: the less local, the more memory a program needs.

C. Temporal: small chunks of program execute faster. Spatial: programs that use a small amount of space run faster.

D. Temporal: within small time windows, only few items are read from memory. Spatial: conversely, within a small memory window, only few instructions read that content.

✔ Answer A.

Answer B.

Answer C.

Answer D.

View Feedback

Question 11 1 / 1 point

SSDs are made up of a set of EEPROMS. When writing a byte to a (functioning) SSD, what is the worst case situation?

A. A whole *page*, i.e., a *part* of an EEPROM, has to be temporarily erased.

B. A whole *block*, i.e., a set of pages corresponding to a whole EEPROM, has to be temporarily erased.

C. Multiple blocks have to be temporarily erased.

D. The whole SSD has to be temporarily erased.

Answer A.

✔ Answer B.

Answer C.

Answer D.

View Feedback

Question 12 **1 / 1 point**

Consider a `for` loop that accesses completely random positions in a large array.

A. The loop still exhibits *some* temporal locality, since it accesses the same instructions at each iteration.

B. The loop still exhibits *some* spatial locality, since the random accesses are all done on the same array.

C. Both A and B.

D. The loop has neither temporal nor spatial locality.

✔ Answer A.

Answer B.

Answer C.

Answer D.

View Feedback

Question 13 **1 / 1 point**

A *hit* happens when the data requested is in a given cache. Otherwise, a *miss* occurs. What are the different types of miss?

A. block, line, set.

B. front, back, sideways.

C. register, cache, main memory.

D. cold, conflict, capacity.

- ◯ Answer A.
- ◯ Answer B.
- ◯ Answer C.
- ✔◯ Answer D.

▷ View Feedback

Question 14 **1 / 1 point**

The main idea behind the memory hierarchy is:

A. Exploit multi-threading to access different data in a given level in parallel.

B. Exploit fast Internet access to backup data instantly to the cloud.

C. Exploit locality to have storage that costs as much as the cheap storage, but performs as well as the expensive storage.

D. Exploit low-level parallelism to access different levels of the hierarchy at the same time.

- ◯ Answer A.
- ◯ Answer B.
- ✔◯ Answer C.
- ◯ Answer D.

View Feedback

Question 15 1 / 1 point

A line in a cache contains at least these info:

A. A dirty bit, the set number, and a part of the lower level memory.

B. A map between some cached data and their actual address in memory.

C. The cache number, its type, and the data stored.

D. A valid bit, the tag, and the set of bytes cached.

- Answer A.
- Answer B.
- Answer C.
- ✓ Answer D.

View Feedback

Question 16 1 / 1 point

The typical implementation of caches depends on three parameters S, E, B. What do these letters correspond to, respectively?

A. Number of lines, number of elements, binary size.

B. Number of sets, number of edges, number of blocks.

C. Number of sets, number of lines, block size.

D. Number of caches, number of subcaches, byte size.

- Answer A.
- Answer B.
- ✓ Answer C.

Answer D.

View Feedback

Question 17 1 / 1 point

A cache where each set has only one line is called:

A. A 0-way set associative cache.

B. A linear cache.

C. A direct mapped cache.

D. Such a cache cannot exist.

Answer A.

Answer B.

✓ Answer C.

Answer D.

View Feedback

Question 18 1 / 1 point

How is an address in memory translated to a line in a cache?

A. The cache stores a map between the memory addresses that it has seen, and the lines in the cache.

B. The cache is always as big as the whole memory, so memory addresses of the memory are also valid in the cache.

C. Some bits of the address are interpreted as an index in a list of *sets*, then some other bits correspond to a *tag* that is searched among the lines of that set.

D. By hashing the (say) 64-bit address to a (say) 8-bit address, and using this as the cache address.

- Answer A.
- Answer B.
- ✓ Answer C.
- Answer D.

View Feedback

Question 19 **1 / 1 point**

Since memory is organized as a hierarchy that duplicates information, multiple copies of data exist (in fact, if some DRAM data is in the L1 cache, it is also in all the caches in between). In the event the CPU writes at some address that *is not* in a given cache (miss):

A. The data is written straight to memory, without using the cache.

B. The data is loaded into the cache, then modified in the cache.

C. A mix of A and B, depending on policy.

D. None of these options: caches are read-only.

- Answer A.
- Answer B.
- ✓ Answer C.
- Answer D.

View Feedback

Question 20 **1 / 1 point**

Since memory is organized as a hierarchy that duplicates information, multiple copies of data exist (in fact, if some DRAM data is in the L1 cache, it is also in all the caches in between). In the event the CPU writes at some address that *is* in a given cache (hit):

A. The data is immediately written to the main memory.

B. The data is overwritten in the cache, and when the line is flushed, it is written back to memory.

C. A mix of A and B, depending on policy.

D. None of these options: caches are read-only.

- ◯ Answer A.
- ◯ Answer B.
- ✔ ◯ Answer C.
- ◯ Answer D.

▷ View Feedback

Question 21 **1 / 1 point**

In a typical multi-core CPU, how are on-chip caches shared?

A. All caches are shared across all cores.

B. No caches are shared across cores, cores use the main memory to share data.

C. The L1 and L2 caches are shared, but there is one unshared L3 cache per core.

D. There are L1 and L2 caches for each core, and a bigger L3 cache shared across cores.

- ◯ Answer A.
- ◯ Answer B.
- ◯ Answer C.

✔ Answer D.

View Feedback

Question 22 1 / 1 point

The three important metrics to evaluate cache performance are:

A. read throughput, write throughput, latency.

B. refresh rate, hit rate, miss rate.

C. miss rate, miss time, hit penalty.

D. miss rate, hit time, miss penalty.

Answer A.

Answer B.

Answer C.

✔ Answer D.

View Feedback

Question 23 1 / 1 point

Blocking is a technique that can improve temporal locality of code. But what is blocking?

A. Reworking a computation so that instead of working on scalars (e.g., ints), it works on sets of scalars that can fit in the cache.

B. Stopping the computation after a certain number of cycles, and restarting it at a different position in memory.

C. Stopping the computation for a certain number of cycle, so that the cache is passively flushed.

D. Carefully putting limits on the memory addresses accessible, and then relaxing these limits one by one.

✔ Answer A.

Answer B.

Answer C.

Answer D.

View Feedback

Question 24 1 / 1 point

Consider a piece of code that has some nested loops, and some code in the innermost loop. This code matches closely a given simple specification in English. Then rearranging the order of the loops:

A. Can drastically improve spatial locality.

B. Has no impact on spatial locality.

C. Probably cannot improve spatial locality: since the English description was simple, the matching code has the best spatial locality.

✔ Answer A.

Answer B.

Answer C.

View Feedback

Attempt Score:23 / 24

Overall Grade (highest attempt)**:**23 / 24

Done