CS3330 Practice Exam 2 – Spring 2014

Name: _____

Directions: Put the letter of your selection or the short answer requested in the box. Write clearly: if we are unsure what you wrote you will get a zero on that problem.

. . . . . .

Question 1: Suppose that 20 years ago you wrote some that code that spends half its time accessing memory and half doing computations. If you ran that same code today, it would probably

A Still be well balanced between memory access and processing
B Spend most of its time computing
C Spend most of its time accessing memory

Answer:

Question 2: Your code never accesses the same address twice; it can benefit from

A temporal locality
B spatial locality
C both of the above
D none of the above

Answer:

Question 3: Pipelining generally

A reduces throughput and increases latency
B reduces throughput and reduces latency
C increases throughput and increases latency
D increases throughput and reduces latency

Answer:

Question 4: Consider a longer pipeline: F, D, E, A, M, W, where the A phase does some kind of address computation. Data needed in the E phase is always from a register, and new register values can can appear in the D, E, or M phases. If we do *not* have data forwarding (i.e., we stall any time data we need is not where we need it), what is the largest data dependency delay we could see?

A 2 cycles
B 6 cycles
C 3 cycles
D 4 cycles
E 1 cycle
F 5 cycles
G 0 cycles

Answer:

Question 5: Organize the following kinds of storage from cheapest to most expensive by byte. Do this by writing four letters in order in the box

A Solid-state disk
B SRAM
C Magnetic disk
D DRAM

Answer:

Question 6: The CPU is attached to which of the following busses? Select all that apply.

A CPU bus
B Memory bus
C School bus
D I/O bus
E System bus

Answer:

Question 7: What is the difference between a data dependency and a data hazard?

A a dependency requires special handling; it is only a hazard if it doesn't get handled correctly
B no difference, they mean the same thing
C a hazard is a property of a pipeline, a dependency is a property of code
D a hazard is a property of a code, a dependency is a property of pipeline

Answer:

Question 8: A 12-bit address $ABCDEFGHIJKL$ (where each letter represents a single bit) is sent to a cache with 16 sets of 1 line each; each block has 8 words of 4 byte each. What is the block offset? (if you think it is the first three bits, you would answer ABC)

Answer:

Question 9: Suppose A and B map to the same line of a direct-mapped cache but with different tags. Which of the following read sequences has exactly one conflict miss?

A Read A, then read B
B Read A
C Read A, then read B, then read A, then read B
D Read A, then read B, then read A

Answer:

Question 10: Suppose the memory phase takes a variable number of cycles. If it signals that it needs another cycle to finish its work, what should we do with the pipeline registers before it?

A different pipeline registers should get different signals
B bubble
C stall
D normal operation

Answer:

Question 11: Suppose your program accesses a large set of bytes of memory; each byte is far from others accessed but each is accessed many times. Which of the following characteristics of the cache will help the program run faster?

Answer:

A small blocks
B large blocks

Question 12: Consider choosing between two caches with the same capacity and access time: one fully-associative and one direct-mapped. If your code never accesses the same address twice, which will result in fewer cache misses?

Answer:

A direct-mapped caches
B they are the same
C fully-associative

Question 13: Suppose the decode phase takes a variable number of cycles. If it signals that it needs another cycle to finish its work, which pipeline registers should be bubbled?

Answer:

A all of those after decode
B just the one right after decode
C all those before decode
D just the one right before decode

Question 14: When given a stall signal, a pipeline register outputs

Answer:

A nop
B the same thing it outputted last cycle
C its input

Question 15: A 12-bit address is sent to a cache with 4 sets with 16 lines per set; each block has 4 words of 8 bytes each. How many bits long is the tag?

Answer:

Question 16: Imagine a 10-stage pipeline where an instruction could depend in the beginning of stage 5 on the results generated by the previous instruction as it completes stage 8. If we have data forwarding, how many cycles does this dependency waste?

A 6, 7, or 8
B 5
C 4
D 9 or more
E 3
F 2
G 0 or 1

Answer:

Question 17: How would a program decide if it queries the instruction cache or the data cache?

A it decides based on the pipeline stage that issues the request
B there's not a best choice; each chip designer picks one based on their use case.
C it decides based on the address used
D it tries both and sees which one has a hit

Answer:

Question 18: When given neither bubble nor stall inputs, a pipeline register outputs

A nop
B the same thing it outputted last cycle
C its input

Answer:

Question 19: Select all of the following that are true.

A A set-associative cache with one line per set is a fully-associative cache
B A set-associative cache with one set is a fully-associative cache
C A set-associative cache with one line per set is a direct-mapped cache
D A set-associative cache with one set is a direct-mapped cache

Answer:

Question 20: When given a bubble signal, a pipeline register outputs

A the same thing it outputted last cycle
B nop
C its input

Answer:

Question 21: Consider the following code:

```
void f(int** a, int**b, unsigned n, unsigned m) {
     for(i = n-1; i >= 0; i -= 1)
         for(j = 0; j < m; j += 1)
             a[j][i] = b[j][i] * 3330;
 }
```

Which of the following changes would make the largest positive impact on its cache locality?

A Change the for-j loop to count up from `0`, not down from `m-1`
B Change `int**` to `int*`
C Block the accesses by nesting three or four for loops instead of just two
D Put the for-i loop inside of the for-j loop instead of the other way around

Answer:

Question 22: Consider the following code:

```
void f(int** a, int**b, unsigned n, unsigned m) {
     for(i = n-1; i >= 0; i -= 1)
         for(j = 0; j < m; j += 1)
             a[i][j] = b[i][j] * 3330;
 }
```

Which of the following changes would make the largest positive impact on its cache locality?

A Block the accesses by nesting three or four for loops instead of just two
B Put the for-i loop inside of the for-j loop instead of the other way around
C Change `int**` to `int*`
D Change the for-i loop to count up from `0`, not down from `n-1`

Answer:

.....