

### CS 3330 Exam 2 – Fall 2016

Name: EXAM KEY

Computing ID: KEY

Letters go in the boxes unless otherwise specified (e.g., for C 8 write “C” not “8”).

Write Letters clearly: if we are unsure of what you wrote you will get a zero on that problem.

Bubble and Pledge the exam or you will lose points.

Assume unless otherwise specified:

- little-endian 64-bit architecture
- %rsp points to the most recently pushed value, not to the next unused stack address.
- questions are single-selection unless identified as select-all

Variable Weight: point values per question are marked in square brackets.

Mark clarifications: If you need to clarify an answer, do so, and also add a \* to the top right corner of your answer box.

#### Information for questions 1–2

Suppose the value 0x9abcdef is written to address 0x12345678 in a 4MB, 4-way set-associative, write-back cache with 256-byte blocks.

Question 1 [2 pt]: (see above)

Which pieces of information that we provided are not needed to compute the tag?

Select all that apply by putting one or more letters in the box, or write “all” if all were needed.

- A the write policy 0.5 points
- B the value 0.5 points
- C the cache size -0.5 point
- D the block size 0.5 point
- E the address 0.5 points whether they answered this or not
- F the associativity -0.5 point

Answer: A B D
---------------

Question 2 [2 pt]: (see above)

What is the set index? Answer as a hexadecimal value.

[0001 0010 0011] [0100 0101 0110] [0111 1000]

Answer: 0x456
---------------

**Question 3 [1 pt]:** The dirty bit is associated with

- A write-back caches
- B write-through caches
- C direct-mapped caches
- D fully-associative caches
- E set-associative caches
- F all of the above
- G none of the above

Answer: **A**

**Question 4 [1 pt]:** The valid bit is associated with

- A write-through caches
- B set-associative caches
- C direct-mapped caches
- D write-back caches
- E fully-associative caches
- F all of the above
- G none of the above

Answer: **F**

**Information for questions 5–6**

Suppose I create a new functional unit that can do a multiply in a single cycle, but that doing this overheats it and it then needs three cycles to cool down before it can run again.

**Question 5 [1 pt]:** (see above)

What is this functional unit's issue rate, in cycles?

Answer: **4**  
(half credit for 3)

**Question 6 [1 pt]:** (see above)

What is this functional unit's latency, in cycles?

Answer: **1**

**Question 7 [2 pt]:** Answer each of the following by putting a T (for true) or an F (for false) in the box. If you put something ambiguous like F you will get 0 points.

- A  **F** dependencies can cause stalling without hazards
- B  **T** "Hazard" is more commonly applied to hardware than to instructions
- C  **F** "Dependency" is more commonly applied to hardware than to instructions
- D  **drop hazards** hazards can cause stalling without dependencies

**Information for questions 8–9**

Suppose you have pipelined a processor into six stages. Most of the work is divided evenly, but there are some operations in the fourth stage you couldn't split into pieces so the fourth stage needs 1300 picoseconds to run while the other five stages only need 1000 picoseconds each. Assume that those times include the pipeline register delays.

**Question 8 [2 pt]:** (see above) What is the the difference between the clock cycle duration needed for this pipeline and the clock cycle duration that would have been needed if stages could have been split evenly? Answer in picoseconds.

Answer: 250

**Question 9 [0 pt]:** (see above) The unequal division of work makes both the instruction latency and the instruction throughput worse than if the division had been equal. Consider the ratio by which each has been hurt by the unequal division. **question dropped**

- A Throughput has been hurt more than latency
- B Latency has been hurt more than throughput
- C They have both been hurt the same amount

Answer:

**Information for questions 10–13**

Caches can be described in terms of address size, tag size, block size, and lines per set. Assume we keep three of those four values the same and change one of the others.

**Question 10 [1 pt]:** (see above) Making the tag bigger results in

- A a bigger cache
- B a smaller cache
- C no change in cache size

Answer: B

**Question 11 [1 pt]:** (see above) Making the block bigger results in

- A a smaller cache
- B no change in cache size
- C a bigger cache

Answer: B

**Question 12 [1 pt]:** (see above) Increasing the number of lines per set results in

- A a smaller cache
- B a bigger cache
- C no change in cache size

Answer: B

**Question 13 [1 pt]:** (see above) Making the address bigger results in

- A no change in cache size
- B a smaller cache
- C a bigger cache

Answer: C

**Question 14 [2 pt]:** If a computer has 8GB memory, a single 1MB cache, and 4KB of registers, the total amount of data it can store is

- A more than 8GB but less than 8GB + 1MB
- B more than 8GB + 1MB but less than 8GB + 1MB + 1KB
- C more than 8GB + 1MB + 4KB
- D 8GB + 1MB + 4KB
- E 8GB + 1MB
- F less than 8GB
- G 8GB

Answer: A

**Information for questions 15–18**

Consider the two versions of following C code, which differ only in loop order. Assume that  $N$  is a large constant.

**Version 1.**

```
for (int i = 0; i < N; ++i)
  for (int j = 0; j < N; ++j)
    foo[j * N + i] -= bar[i * N + i] + bar[j * N + i];
```

**Version 2.**

```
for (int j = 0; j < N; ++j)
  for (int i = 0; i < N; ++i)
    foo[j * N + i] -= bar[i * N + i] + bar[j * N + i];
```

**Question 15 [2 pt]:** (see above) Which has *temporal* locality for bar?

- A neither
- B version 2
- C version 1
- D both

Answer: C

**Question 16 [2 pt]:** (see above) Which has *spatial* locality for foo?

- A version 1
- B both
- C version 2
- D neither

Answer: C

**Question 17 [2 pt]:** (see above) Which has *spatial* locality for bar?

- A version 1
- B version 2
- C both
- D neither

Answer: B

**Question 18 [2 pt]:** (see above) Which has *temporal* locality for foo?

- A version 1
- B both both read and write, giving weak locality
- C version 2
- D neither neither has strong locality

Answer: B D

**Question 19 [2 pt]:** Suppose you are choosing between stalling and forwarding to resolve a particular hazard in a five-stage pipeline.

Suppose that extra logic needed to implement forwarding would slow down your clock cycle by 5%, or you could stall for a single cycle instead without slowing the clock. You can only implement one of the two options. You should chose stalling if you expect the hazard to be triggered by

- A more than 75% of instructions
- B fewer than 1% of instructions **half credit**
- C more than 95% of instructions
- D more than 99% of instructions
- E fewer than 25% of instructions
- F fewer than 5% of instructions

Answer: **F**

**Question 20 [2 pt]:** Consider a 15-stage pipeline. If

- instruction X produces a value at the end of its eleventh stage
- instruction Z needs that value at the beginning of its fifth stage
- you are running the instruction sequence X; Y; Z

Answer: **5 half credit for 4 or 6**

how many cycles of stalling will occur?

**Information for questions 21–23**

For each of the following, answer by putting an N (for normal), B (for bubble), or S (for stall) in each pipeline register’s box.

**Question 21 [2 pt]:** (see above) Suppose the instruction currently in the memory stage emits a signal saying it is not ready to continue. What pipeline control signals should you provide?

F     D     E     M     W

**Question 22 [2 pt]:** (see above) Suppose the instruction currently in the execute stage needs to stay there for another cycle and the instruction currently in the decode stage needs to be squashed. What is the set of pipeline control signals that implements these two constraints with the fewest stalls?

F     D     E     M     W

**half credit for SBSBN**

**Question 23 [2 pt]:** (see above) Suppose the instruction currently in the execute stage determines all instructions that have been fetched since it was fetched were incorrect. What pipeline control signals should you provide?

F     D     E     M     W

**half credit for NBBBN**

**Question 24 [2 pt]:** If a sequential processor currently has a clock cycle duration of 500 picoseconds and if pipeline registers take 10 picoseconds to stabilize, what throughput would we expect if we made the processor into a two-stage pipelined processor?

- A  $\frac{1}{250}$  instructions per picosecond
- B more than  $\frac{1}{250}$  instructions per picosecond
- C between  $\frac{1}{250}$  and  $\frac{1}{260}$  instructions per picosecond
- D  $\frac{1}{260}$  instructions per picosecond
- E less than  $\frac{1}{260}$  instructions per picosecond

Answer: **D E**

**Question 25 [2 pt]:** If we expanded our five-stage pipeline to seven stages by having two cycles of fetch and two cycles of memory, which of the following code snippets would require more cycles of stalling than they do in the five-stage pipeline?

Select all that apply by putting one or more letters in the box

- A `ret`
- B `rmmovq %rax, (%rax)`  
`rmmovq %rax, (%rax)`
- C `addq %rax, %rax`  
`addq %rax, %rax`
- D `mrmovq (%rax), %rax`  
`mrmovq (%rax), %rax`

Answer: **A D**

.....  
**Pledge:**

On my honor as a student, I have neither given nor received aid on this exam.

---

Your signature here