CS 3330 Final Exam – Spring 2016

Name: <u>EXAM KEY</u>

Computing ID: <u>KEY</u>

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

Multiple-select: are all clearly marked; put 1 or more letters in the box.

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.

.....

Question 1 [2 pt]: Which of the following is an example of pipelining the creation of books?

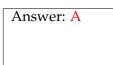
- **A** Person A is writing volume 2 while Person B is binding volume 1
- **B** Person A drafts the contents while Person B creates the paper to print it on
- **C** Person A is writing one book while Person B is writing another book
- **D** all of the above
- **E** none of the above

Information for questions 2–3

Optimization strategies are not restricted to code...

Question 2 [2 pt]: (see above) An online retail company buying a package delivery company is an example of

- A function inlining
- **B** eliminating loop inefficiencies
- **C** cache blocking
- **D** loop unrolling
- **E** multiple accumulators



Answer: A

Question 3 [2 pt]: (see above) This exam randomizes question order; grouping related questions (like this question and its pair) is intended to optimize your performance by utilizing

- A cache blocking
- **B** function inlining
- **C** eliminating loop inefficiencies
- **D** multiple accumulators
- **E** loop unrolling

Question 4 [2 pt]: Memory segments are defined by

- **A** data structures used by hardware only
- **B** data structures used by both hardware and software
- **C** data structures used by software only
- **D** none of the above; they are just an abstraction

Question 5 [2 pt]: An exception table is

- **A** an array
- **B** a hash table
- **C** a tree
- **D** none of the above

Information for questions 6–8

The following questions ask about how each of the three main types of exceptions differ from the other two.

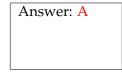
Question 6 [2 pt]: (see above) Faults are different from other exception types in that

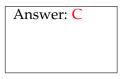
- **A** faults are not caused by running an assembly instruction
- **B** faults never cause Aborts or Signals
- **C** faults are handled by a different mechanism than other exceptions
- **D** faults are intentionally triggered by user code
- **E** faults always cause Aborts or Signals
- **F** faults are never intentionally triggered by user code

Question 7 [2 pt]: (see above) Traps are different from other exception types in that

A traps are handled by a different mechanism than other exceptions

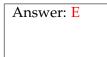
- B traps never cause Aborts or Signals
- **C** traps always cause Aborts or Signals
- **D** traps are not caused by running an assembly instruction
- E traps are always intentionally triggered by user code
- **F** traps are never intentionally triggered by user code





Answer: A	

[Answer: F



Question 8 [2 pt]: (see above) Interrupts are different from other exception types in that

- A interrupts are not caused by running an assembly instruction
- B interrupts never cause Aborts or Signals
- **C** interrupts always cause Aborts or Signals
- **D** interrupts are never intentionally triggered by user code
- **E** interrupts are always intentionally triggered by user code
- **F** interrupts are handled by a different mechanism than other exceptions

Information for questions 9–10

Consider the following C definitions:

```
typedef struct node_t { TYPE data; node *next; } node;
typedef struct range_t { size_t length; TYPE *array; } range;
```

Question 9 [2 pt]: (see above) Which list uses the least memory overall (including both heap and stack)? If multiple options are tied for smallest, select all that apply.

Α	TYPE	*list	
_			

- **B** range list
- **C** The answer is different if TYPE is char than if TYPE is int
- D node *list

Question 10 [2 pt]: (see above) Which list type puts the most data on the stack? If multiple options are tied for largest, select all that apply.

- **A** The answer is different if TYPE is char than if TYPE is int
- \boldsymbol{B} range list
- C TYPE *list
- D node *list

Question 11 [2 pt]: All of the following are enabled by virtual memory; which one would *not* be enabled without it?

A assembly address size can differ from the amount of physical memory present

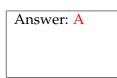
B multiple processes can share the same kernel memory

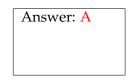
C code can be written using labels, letting the assembler generate their addresses

D code can be compiled without knowledge of what other processes will run concurrently with it

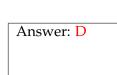
Question 12 [2 pt]: Computers typically have a special clock to generate exceptions for the purpose of allowing the kernel to create context switches. This clock should not generate exceptions too frequently for the same reason that

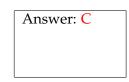
- A page tables should not have too many levels
- **B** cache sets should not have too many entries
- **C** pipelines should not be made too deep
- **D** loops should not be unrolled too many times





Answer: **B**





Information for questions 13–14

For each of the following, assume u and v are both declared as unsigned ints. Select all that could apply for some values of u and v; for example, given "u _____ v" you'd select <, =, and > I use \land as a carat and \sim as a tilde, both larger than usual for increased legibility.

Question 13 [2 pt]: (see above) (u<<16) & (u>>16) _____u

- A = B > C <
- Question 14 [2 pt]: (see above) $u + \sim v __u v$
- A = B >
- **C** <

Information for questions 15–16

Thus far, fast-and-expensive storage has always been volatile (like SRAM, DRAM, and registers) and slow-and-cheap storage always nonvolatile (like tape, disk, and flash).

Question 15 [1 pt]: (see above) Suppose someone invents a new storage technology: it is about as fast as magentic disk but costs a lot less and is volatile. What should we use it for?

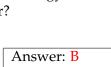
Select all that apply

- **A** it be good for existing file systems
- **B** it be good for existing virtual memory swapping
- **C** it be good for existing cache hierarchies
- **D** none of the above

Question 16 [1 pt]: (see above) Suppose someone invents a new storage technology: it costs similar to SRAM but is a little faster and nonvolatile. What should we use it for?

Select all that apply

- **A** it be good for existing file systems
- **B** it be good for existing cache hierarchies
- **C** it be good for existing virtual memory swapping
- **D** none of the above



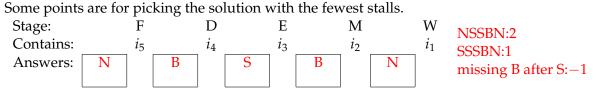
Answer: A C

Answer: **B**C



Question 17 [3 pt]: In the following diagram, indicate the control signals to give each pipeline register by putting a single letter in each box; use N for normal, B for bubble, and S for stall.

Assume that i_4 and i_5 resulted from incorrect speculative execution and should not be allowed to continue; that i_3 needs another cycle in the execute stage; and that all other instructions are OK and may continue to execute normally.



Question 18 [2 pt]: If we replace a set-associative cache with a different cache with half as many sets each containing twice as many lines (without changing block size),

- **A** the tag gets longer
- **B** the tag stays the same size
- **C** the tag gets shorter

Question 19 [2 pt]: pushq is a 10-byte instruction. We can replace pushq with other operations (math and register-memory moves); how does the storage requirements for push change if we use other operations instead of pushq?

- A increases by more than 3 bytes
- **B** decreases by more than 3 bytes
- **C** increases by 2 or 3 bytes
- **D** changes by no more than 1 byte
- **E** decreases by more than 2 or 3 bytes

Information for questions 20–23

Consider a floating-point format with 7 bits overall, 4 of which are exponent bits.

Question 20 [2 pt]: (see above) What exponent bits are used to represent $-\frac{5}{8}$? Answer as four bits, such as 0000

Question 21 [2 pt]: (see above) Which of the following is true using this format?

A 1.0 / 32.0 is 0.0

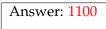
- **B** 6.0 + 1.0 is 6.0
- **C** (x x) == 0 is true for all x
- **D** None of the above

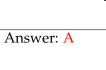
Question 22 [2 pt]: (see above) What number is represented by the bits 0101010? Answer as a base-2 number such as -101.11

Answer: C

Answer: D

Answer: 0110





KEY

KEY

Question 23 [2 pt]: (see above) What fraction bits are used to represent $-\frac{5}{8}$? Answer: 01 Answer as two bits, such as 00

Question 24 [0 pt]: Cognitive break. Write a joke or anecdote here, or doodle something interesting, or just smile at the blank space worth 0 points and move on.

Question 25 [2 pt]: Suppose we add a new ifun for OPq, mulq that requires four consecutive cycles in the Execute stage. That means execute may stall for a single operation, but does it also impact pipeline hazards?

Select all that apply

A Two consecutive OPqs will become a new kind of hazard.

B The branch misprediction hazard may now result in more instructions being removed from the pipeline via bubbling.

- **C** The load-use hazard can now need more than a single cycle of stalling.
- **D** The return hazard can now need extra cycles of stalling.
- **E** None of the above

Information for questions 26–29

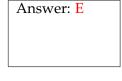
Various topics discussed during our exploration of exceptions enabled communication between elements of a computer system. The following questions ask about these communications

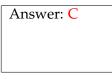
Question 26 [1 pt]: (see above) Communication from kernel to user is enabled by

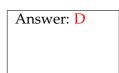
- A fault
- **B** interrupt
- **C** signal
- **D** trap
- **E** none of the above

Question 27 [1 pt]: (see above) Communication from hardware to kernel is enabled by

- A signal
- **B** trap
- **C** fault
- **D** interrupt
- **E** none of the above







Question 28 [1 pt]: (see above) Communication from user to kernel is ena

- A trap
- **B** interrupt
- **C** signal
- **D** fault
- **E** none of the above

Question 29 [1 pt]: (see above) Communication from kernel to hardware is enabled by

- A signal
- **B** fault
- **C** interrupt
- **D** trap
- **E** none of the above

Question 30 [2 pt]: An exception handler is

- **A** both user- and kernel-mode software
- **B** both kernel-mode software and hardware
- **C** primarily kernel-mode software
- **D** primarily hardware
- **E** primarily user-mode software

Information for questions 31–32

A binary tree can be stored in an array; entry *i*'s left child is 2i and its right child is 2i + 1:

0	1	2	3	4	5	6	7	8	
(unused)	root	root.l	root.r	root.l.l	root.l.r	root.r.l	root.r.r	root.l.l.l	

Consider such an array used with a direct-mapped cache with 128 lines, each large enough to hold 4 array entries. Suppose the array is aligned so that entries 0, 1, 2, and 3 are in the same cache line.

Question 31 [2 pt]: (see above) If code accesses root, then root.l, then root.l, then root.l.l.l, etc.; how many entries can we accesses before we have to evict $\begin{bmatrix} 0 & 1 & 2 & 4 & 8 & 16 \end{bmatrix}$ one of the other entry's cache lines? Answer as a base-10 number.

Question 32 [2 pt]: (see above) Which method of tree traversal would have 1 locality?

- A pre-order depth first
- **B** in-order depth-first
- **C** breadth-first
- **D** post-order depth first
- **E** all of the above have the same locality

Information for questions 33–36

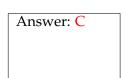
In a multi-level page table,

Answer: E

Answer: C	

32, 64, 128=0)	
ve the best spatia	L

Answer: 10 (0,



Answer: A

Question 33 [2 pt]: (see above) If part-way through following the page table the MMU hardware finds the read-only bit set and the CPU is attempting to write to memory,

A stop; the next page table will be in RAM but will also be marked read-only

- **B** stop; the next page table might not even be in RAM half-credit
- **C** keep going, only stopping if the last page table is marked read-only
- **D** keep going; even if the last page table is marked read-only it is the OS,

not the MMU hardware, that enforces read-only

Question 34 [2 pt]: (see above) Which of the following tells the location of the first page table?

- **A** the PO
- **B** the VPN from high-order bits of the address
- **C** the TLB
- **D** the VPN from low-order bits of the address
- **E** the PTBR

Question 35 [2 pt]: (see above) The last VPN used is

- A an index into a page containing data (not a page table)
- **B** it depends on if there is a page fault or not
- **C** an index into a page table

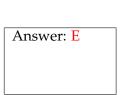
Question 36 [2 pt]: (see above) In the common case where there are 3 or 4 levels of page table and several thousand pages are allocated in a few contiguous regions of virtual memory, table storage \div data storage is

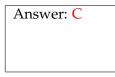
- **A** less than $\frac{1}{100}$
- **B** between $\frac{1}{2}$ and 2
- **C** between 2 and 100
- **D** more than 100
- **E** between $\frac{1}{2}$ and $\frac{1}{100}$

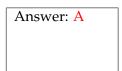
Information for questions 37–39

Consider 38-bit virtual addresses and 4-byte page-table entries, where each PTE stores 8 bits of metadata (executable, protected, etc).

Question 37 [2 pt]: (see above) If you have 256-byte pages, then the largest possible physical address space is how many bytes? Answer as a power of two, such as 16B or 128GB.







Answer: 1MB (12+8=20)



F sometimes one of the above, sometimes another, depending on if we have a hit or not

Question 38 [2 pt]: (see above) If you want to have a single-level page table and to fit the entire page table in one page of memory, what is the smallest page size (in bytes) you could use? Answer as a power of two, such as 16B or 128GB.

Question 39 [2 pt]: (see above) If you want to have a three-level page table and to fit each page table in one page of memory, what is the smallest page size (in bytes) you could use? Answer as a power of two, such as 16B or 128GB.

Question 40 [2 pt]: Suppose we wanted to add a conditional call instruction to Y86-64. Conditional call (i.e., callg, callge, etc.) would require

Variant O page 9 of 10

Select all that apply

- A new branch prediction logic (beyond that already present for jXX)
- B more than the 9 bytes needed to encode call
- **C** more register read- or write-ports than call
- **D** none of the above

Information for questions 41-42

The translation lookaside buffer is a cache that

Question 41 [2 pt]: (see above) Produces as output

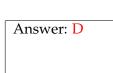
- A the physical page number from an address
- B entire physical addresses half-credit
- **C** all the virtual page numbers from an address
- **D** entire virtual addresses
- **E** a single virtual page number from an address
- **F** none of the above

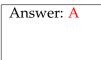
Question 42 [2 pt]: (see above) Takes as input

- A all the virtual page numbers from an address
- **B** a single virtual page number from an address half-credit
- **C** entire virtual addresses half-credit
- **D** entire physical addresses
- **E** the physical page number from an address
- **F** none of the above

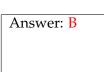
Question 43 [2 pt]: Which of the following assembly snippets is removed by the assembler and not present in the binary?

- A nop
- B loop:
- C addq %rax, %rcx
- **D** jg loop





Answer: A





Answer: 2KB

Question 44 [2 pt]: After profiling your code you find that it is spending 20% of its time evaluating the comparison statement in the innermost for loop; and 50% of its time accessing elements of an array.

Which of the following optimizations would give the biggest speedup? Assume the optimizations add no overhead not explicitly mentioned.

A add an extra 30% to the runtime to set up, then split the entire program to run in parallel on two processors saves 20%

B reorder loops to reduce memory access times by 30% saves 15%

C blocking to reduce memory access times by 50%, but increase loop

comparison time by $1.5 \times$ saves 15%

D $10 \times \text{loop unrolling saves 18\%}$

Question 45 [2 pt]: Suppose physical memory is larger than the virtual address space. Which of the following can benefit from swapping?

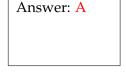
Select all that apply

- **A** a single user process by itself
- **B** multiple concurrent user processes
- **C** a single user process and the kernel
- **D** none of the above

Pledge:

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

Your signature here



Answer: B

Email ID: KEY