CS 3330 Final Exam – Fall 2015

Name: ________________________________ Computing ID: __________

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

Multiple-select: are all clearly marked; put 0 or more letters in the box.
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: Given a write-through set-associative cache with

- 16-bit addresses
- 4 lines per set
- 8 bytes (64 bits) of data per block
- 2 bits stored per line to implement the replacement policy
- 64 sets

Answer:

how many bits are stored per line? Include data and all other bits stored in each line. Answer by writing a single base-ten integer in the box.

Question 2: Suppose an exception occurs during instruction x; after it is handled the CPU re-runs instruction x. The exception was a

A trap
B interrupt
C fault
D insufficient information to chose one of the above

Answer:

Question 3: Assume a set-associative and a direct-mapped cache both have the same address size, same data capacity, and same number of index bits. This means

A The direct-mapped cache has more tag bits
B The direct-mapped cache has fewer tag bits
C The direct-mapped cache has more block offset bits
D The direct-mapped cache has fewer block offset bits
E None of the above

Answer:

Question 4: If we were running a non-pipelined processor (like SEQ) with a real memory system, which of the following optimization strategies would not longer provide a speed benefit?

A adding local variables
B function call inlining
C using multiple accumulators
D loop unrolling
E improving cache locality
F all of the above would provide a speed benefit
G none of the above would provide a speed benefit

Answer:

Information for questions 5–6
The translation lookaside buffer is a cache. The following questions ask about how the TLB compares to more traditional caches like the L2.

Question 5: (see above) When splitting the bits of the input to the TLB, which part is missing (i.e., always 0 bits long)?

A the index
B the offset
C the tag
D none of the above

Answer:

Question 6: (see above) The input to most caches is a physical address; the input to the TLB is

A a physical page offset
B a virtual page number
C a physical address
D a virtual page offset
E a virtual address
F a physical page number

Answer:

Question 7: If virtual addresses are 28 bits and pages are 4K bytes, how many bytes is each page table entry?

A 2
B 8
C 4
D 16
E 1

Answer:

Question 8: Consider the following code:

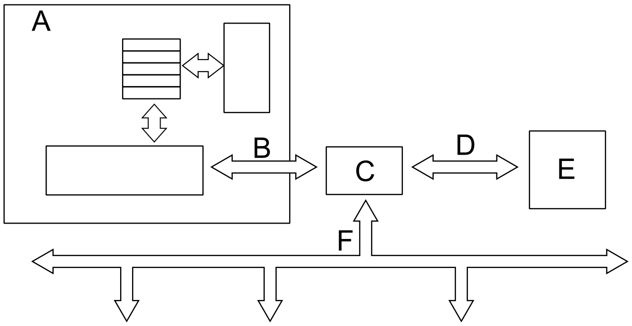
```
for(i=0; i<strlen(s); i+=1)
    if (strcmp(s+i, "now") > 0)
        cnt += 1;
```

This code would most likely benefit from which of the following optimization strategies?

A adding local variables
B improving cache locality
C loop unrolling
D function call inlining
E using multiple accumulators

Answer:

Information for questions 9–12
The chapters on storage and exceptions contained diagrams of the computers components, which looked much like this:



The following questions ask you to label this image. Answer with one of the letters on the above drawing (i.e., **A** through **F**), or with "**G** None of the labeled items in the drawing"

Question 9: (see above) The *Bus interface* is

Answer:

Question 10: (see above) The *I/O bus* is

Answer:

Question 11: (see above) The *System bus* is

Answer:

Question 12: (see above) The *memory bus* is

Answer:

Information for questions 13–15
Assume that each page table entry is 1 byte (8 bits); from most- to least-significant bit these mean 1 bit "on disk"; 4 bits "page number"; 1 bit "kernel mode"; 1 bit "executable"; and 1 bit "read-only". Suppose that, in the course of resolving an address using a two-level page table you encounter first a page table entry with value 0x3E, then one with value 0x7D.

Question 13: (see above) Code that may access this memory may also execute its contents as code

A False
B True

Answer:

Question 14: (see above) User-mode code can access this memory

A True
B False

Answer:

Question 15: (see above) Code that may access this memory may also updated its contents

A False
B True

Answer:

Question 16: 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 was an incorrect speculative execution and should not be allowed to continue, and that i_2 depends on some not-yet-available external resource and needs another cycle in the M stage of the pipeline.

Stage:		F		D		E		M		W
Contains:		i_5		i_4		i_3		i_2		i_1

Question 17: The exception table is an array containing

A exception handler code
B exception handler code numbers
C addresses of exception handler code
D exceptions
E exception numbers
F addresses of exceptions

Answer:

Information for questions 18–19
Assume that each page is 8 bytes long, each physical page number is 4 bits long, and each PTE is 1 byte long.

Question 18: (see above) A physical address is how many bits long?

A 9 or 10
B 14 or more
C 3, 4, or 5
D 6 or 7
E 11, 12, or 13
F 8
G 2 or less

Answer:

Question 19: (see above) If the PTBR contains page number 3, at what address will the top-level page table entry be found for 12-bit virtual address 0x93E? Answer as a hexidecimal number.

Answer:

Question 20: When an exception occurs, the hardware performs some actions before the exception handler is run. Which of the following is *not* part of that hardware-performed setup for handling an exception?

A change the PTBR to point to the handler's page table hierarchy
B change to kernel mode
C store the current state of the processor's registers
D all of the above are part of the handling setup

Answer:

Question 21: Given 6-bit numbers where 010001 + 001010 = 010110, which of the following describes the number representation used?

A floating-point with 3 exponent bits
B integer
C floating-point with 4 exponent bits
D floating-point with 2 exponent bits

Answer:

Question 22: Suppose you had a computer *N* which was much like the Intel chips we've studied this semester: 3 GHz clock, 10-stage pipeline, L1 through L3 caches with 100-cycle RAM lookups on cache miss. You also have a computer *M* with a 500 MHz clock, 1-stage pipeline, and a new memory technology where every memory access takes just a single cycle. For which of the following applications would you expect *M* to have the best speed relative to *N*?

A counting how many times each word appears in a very large string using a hashmap of word $\mapsto$ count
B sorting very large arrays
C searching for primes using a simple check-all-possible-factors approach
D a self-modifying-code program where most of the instructions it executes were written earlier by the program itself
E addition of large matrices

Answer:

Question 23: Which of the following is true?

A A direct-mapped cache is a set-associative cache with set size 1
B A direct-mapped cache is a fully-associative cache with only one set
C A direct-mapped cache is a fully-associative cache with set size 1
D A direct-mapped cache is a set-associative cache with only one set

Answer:

Information for questions 24–25
Recall the definitions

```
typedef struct node_t { TYPE payload; struct node_t *next; } node;
typedef struct range_t { unsigned int length; TYPE *ptr; } range;
```

as well as the sentinel-terminated `TYPE *` array.

Question 24: (see above) **Select all that apply**: Assume function `middle` returns the middle third of a provided list as a new list, but does not call `malloc` or other otherwise allocate memory. For which of the following signatures would it be possible for the old list not to have changed after invoking `middle`?

A `range middle(range)`
B `node *middle(node *)`
C `TYPE *middle(TYPE *)`

Answer:

Question 25: (see above) **Select all that apply**: Assume function `end` returns the last half of a provided list as a new list, but does not call `malloc` or other otherwise allocate memory. For which of the following signatures would it be possible for the old list not to have changed after invoking `end`?

A `node *end(node *)`
B `TYPE *end(TYPE *)`
C `range end(range)`

Answer:

Information for questions 26–31
Consider a ten-stage pipline, with stages named S_0 through S_9.

Suppose the `mul` instruction in this architecture requires its operands before beginning S_3 and produces its result by the end of S_5, writing it back to the register file in S_9.

Suppose both the `add` instruction and the `shift` instructions in this architecture require their operands before beginning S_3 and produces their result by the end of S_3, writing it back to the register file in S_9.

Assume there is data forwarding available.

In the C code in these questions, assume all variables are mapped to registers.

Question 26: (see above) Compare the runtime of `y = x * 5` and `y = (x<<2) + x`, measured by the time it takes `y`'s value to be entered into the register file

A they take the same number of cycles
B `y = (x<<2) + x` is faster
C `y = x * 5` is faster

Answer:

Question 27: (see above) Compare the runtime of `y = x * 5` and `y = (x<<2) + x`, measured by the time it takes for the resulting value of `y` to be available somewhere in the pipeline

A they take the same number of cycles
B `y = (x<<2) + 5` is faster
C `y = x * 5` is faster

Answer:

Question 28: (see above) What is the difference in cycles used to run `(x * y) * z * w` as compared to `x * y * (z * w)`?

A 2 cycles
B 0 cycles
C 7 or more cycles
D 3 cycles
E 4, 5, or 6 cycles
F 1 cycle

Answer:

Question 29: (see above) What is the *throughput* of `mul`?

A 1 / 13 cycles
B 0
C 1 / 10 cycles
D 1 / 3 cycles
E 1 / 1 cycle

Answer:

Question 30: (see above) What is the *latency* of `mul`?

A 0 cycles
B 10 cycles
C 1 cycle
D 13 cycles
E 3 cycles

Answer:

Question 31: (see above) Compare the runtime of `y = x * 5` and `y = (x<<2) + x`, measured by the time it takes before the following line of code may begin running

A `y = x * 5` is faster
B `y = (x<<2) + x` is faster
C it depends on what instructions follow this assignment
D they take the same number of cycles

Answer:

Information for questions 32–33
Which of the following code snippets is fastest? Assume `n` is very large (more than ten thousand).

Question 32: (see above)

A `for(i=0;i<n;i+=1) for(j=0;j<n;j+=1) a[i][j] = b[j][i];`

B `for(j=0;j<n;j+=1) for(i=0;i<n;i+=1) a[i][j] = b[j][i];`

C
```
for(k=0; k<n; k+=16) for(l=0; l<n; l+=16)
 for(j=0;j<16;j+=1) for(i=0;i<16;i+=1)
   a[i+l][j+k] = b[j+k][i+l];
```

D two or more of the above are equivalently the fastest

Answer:

Question 33: (see above)

A
```
for(k=0; k<n; k+=16) for(l=0; l<n; l+=16)
 for(j=0;j<16;j+=1) for(i=0;i<16;i+=1)
   a[i+l][j+k] = b[i+l][j+k];
```

B `for(i=0;i<n;i+=1) for(j=0;j<n;j+=1) a[i][j] = b[i][j];`

C `for(j=0;j<n;j+=1) for(i=0;i<n;i+=1) a[i][j] = b[i][j];`

D two or more of the above are equivalently the fastest

Answer:

Information for questions 34–37
In binary, the number $\frac{1}{3}$ is 0.01010101 . . .. In the following, assume that `x` is an `int` and `y` is a `short`.

Question 34: (see above) `(short)((0x5555 * (int)y)>>16)` and `y/3` could differ because of overflow

A True
B False

Answer:

Question 35: (see above) Assume the throughput of `>>` and `-` is 1 per cycle, of `*` is 1 per 3 cycles, and of `/` is 1 per 9 cycles. Assume also an additional 1-cycle latency for each of these operations. Which would be faster: `((0x55555555>>w)*x)>>(32-w)` or `x/3`? Assume the compiler optimizes operation ordering within the constraints of the parentheses.

A `((0x55555555>>w)*x)>>(32-w)` would be faster
B `x/3` would be faster

Answer:

Question 36: (see above) For each `x` there exists some integer `w` (which may be different for different `x`s) such that `abs(x/3 - q) < 10`, where `q = (((0x55555555>>w)*x)>>(32-w))`.

A False
B True

Answer:

Question 37: (see above) `(short)((0x5555 * (int)y)>>16)` and `y/3` could differ even when neither computation overflows

Answer:

A True
B False

Information for questions 38–40
In Y86, two instruction families checked condition codes: `cmovXX` and `jXX`. In ARM architectures, all instructions can check condition codes. The following questions ask about adding conditions to all instructions in Y86, so we'd have e.g. `rmmovqge` and `addql` and so on.

Question 38: (see above) Which of the following would need more bytes than it currently uses in Y86 to encode its new conditional versions?

A `OPq`
B `pushq`
C `rmmovq`
D `ret`
E all of the above
F none of the above

Answer:

Question 39: (see above) We had to implement prediction to efficiently handle `jXX` but not `cmovXX`. Which of the following instructions would also need prediction to efficiently handle its new conditional versions?

A `mrmovq`
B `call`
C `OPq`
D `popq`
E all of the above
F none of the above

Answer:

Question 40: (see above) **Select all that apply**: For which instructions would a conditional version have identical semantics to the standard, non-conditional version?

A `popq`
B `call`
C `nop`
D `pushq`
E `halt`
F `rmmovq`
G `OPq`
H `irmovq`
I `ret`
J `mrmovq`

Answer:

Question 41: Assuming the standard IEEE-style bias, what is the smallest number of bits a floating-point number could have and represent the number $\frac{-29}{4}$ exactly?

A 4
B 6
C 8
D 5
E 7

Answer:

Information for questions 42–44
Suppose addresses A and B have different tags but the same index (for those caches that have tags and indices, that is). Assume the cache is empty prior to the accesses listed in the questions.

Question 42: (see above) In the access pattern "read A, then read B, then read A, then read B" the second read of A will be a miss for

A a set-associative cache
B a fully-associative cache
C a direct-mapped cache
D all of the above
E none of the above

Answer:

Question 43: (see above) In the access pattern "write A, then write B, then write A", to minimize writes to the higher-level cache we should pick which of the following?

A a write-through fully-associative cache
B a write-back fully-associative cache
C a write-through direct-mapped cache
D a write-back direct-mapped cache
E two or more of the above are equivalently the best option

Answer:

Question 44: (see above) In the access pattern "read A, then read B" the read of B will be a miss for

A a fully-associative cache
B a direct-mapped cache
C a set-associative cache
D all of the above
E none of the above

Answer:

Question 45: Given 6-bit numbers where 010101 + 000001 = 010110, which of the following describes the number representation used?

A floating-point with 2 exponent bits
B integer
C floating-point with 3 exponent bits
D floating-point with 4 exponent bits

Answer:

Pledge:

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

____

Your signature here