Computer Science Comprehensive Examination
Computer Architecture
[60 points]

Please do all of your work on these sheets. Do not do your work in a blue book.

Problem 1: Short Answer [2 points each, 18 points total] (Keep your answer to one line or less)

A. Compared to an 8K direct-mapped cache, what type of misses will an 8K two-way set associative cache will have fewer of? Circle all that apply.

(a) compulsory
(b) conflict
(c) capacity

B. Which instruction set will give better code density: (a) a RISC instruction set with fixed-format 32-bit instructions, or (b) a CISC instruction set with variable-length instructions?

(b) CISC

C. Which mean should be used to combine normalized execution times? Circle all that apply.

(a) Arithmetic mean
(b) Geometric mean
(c) Harmonic mean

D. Adding a cache memory to a system changes which of the following memory parameters? Circle all that apply and denote the direction of change with an up arrow or a down arrow.

(a) Memory latency *(decreases)*
(b) Memory bandwidth *(usually increases)*
(c) Memory address space
(d) Memory reliability

E. With very accurate (>95%) branch prediction, adding an additional decode stage to a pipeline causes very little (<5%) increase in CPI. (True or False?)

True

F. In the steady state, what will be the prediction accuracy of a two-bit branch predictor on the sequence TTNTTNTTN (T=taken, N=not taken)? 6/9 = 67%

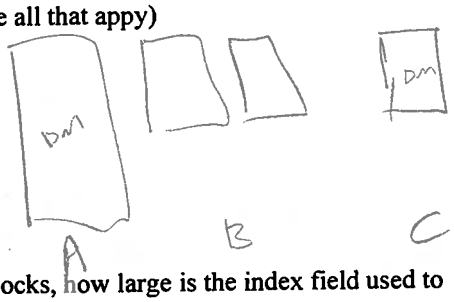
The 2-bit predictor always predicts T and thus is right 2/3 of the time.

G. An instruction for protected subsystem entry must change what two things atomically? (5 words or less).

Control flow and privilege level

H. Suppose that cache A, an 8K-byte direct-mapped cache, cache B, an 8K-byte two-way set associative cache, and cache C, a 4K-byte direct-mapped cache are all referenced with an identical address sequence. Which of the following statements are true: (circle all that appy)

(a) A will contain a superset of the data in B
(b) B will contain a superset of the data in A
(c) A will contain a superset of the data in C
(d) B will contain a superset of the data in C



I. In an 8K-byte two-way set-associative cache with 32-byte blocks, how large is the index field used to address the cache array? (write down the number of bits)

2^{13} bytes, 2^{5} blocks, 2 ways => 2^{-} blocks per way, so 7-bits are required

Problem 2: Cache Architecture [4 points each, 16 points total]

Consider a computer system with a 32-bit virtual addresses space, byte addressing, and a 4K-byte page size. Each virtual address is divided into a 12-bit offset field that identifies a byte within a page and a 20-bit page field that identifies a page in virtual memory. An address translation unit translates the 20-bit page number into a 20-bit physical page frame number.

A. With this machine organization how large a direct-mapped cache can be realized as a virtually-indexed, physically-tagged cache without any homonym or synonym problems?

4K Bytes. Any larger an the index field of the cache would need to include translated bits from the VA

B. How large a four-way set-associative cache can be realized as a virtually-indexed, physically-tagged cache without any homonym or synonym problems?

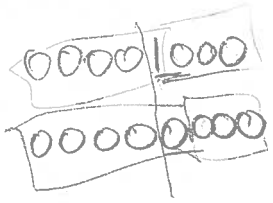
16K Bytes. Each of the four "ways" can be up to 4K Bytes in size.

C. Suppose I build a direct-mapped cache ~~eight~~ times the size of your answer to part (A). What can go wrong? Give a short example.

Two synonyms may map to different locations in the cache even though they refer to the same word of main memory.
Suppose virtual addresses 00000000_{16} and 00010000_{16} both map to physical address 00000000. On cache hits, however they will refer to different sets.

D. Can the problem you identified in part (C) be fixed by restricting the mapping of pages to page frames? (Yes or No?). If your answer is "Yes", explain how.

Yes, by restricting each page to be mapped to a page frame so that bits 12-13 of the physical address match bits 12-13 of the virtual address.



Problem 3: Instruction Issue [16 points]

Consider the following instruction sequence:

```
1      LD     X,R1
2      LD     Y,R2
3      ADD    R1,R2,R3
4      ADD    R3,1,R8
5      LD     Z,R4
6      ADD    R2,R4,R5
7      MUL    R5,3,R6
8      ADD    R6,1,R7
```

You may assume that all operations have two-cycle latency. That is, the result of an operation is available two cycles after that operation enters the first execution stage of the machine. Also assume that there is full bypassing, and that all loads hit in the cache. Hint: you need only consider the execution and memory stages of the pipeline to answer this question.

A. [3 points] How many cycles does this sequence take to execute on a single-issue in-order machine? Measure time from the cycle that the first instruction issues to the cycle in which the result of the last instruction is available for use. Show your work.

```
   1  2  3  4  5  6  7  8  9 10 11 12 13 14
1  X  X
2     X  X
3           X  X
4                 X  X
5                    X  X
6                          X  X
7                             X  X
8                                   X  X
```

14 cycles

B. [3 points] How many cycles does this sequence take to execute on an in-order machine with multiple issue (assume an issue width as wide as you need)? Show your work

```
   1  2  3  4  5  6  7  8  9 10 11 12
1  X  X
2  X  X
3        X  X
4              X  X
5              X  X
6                    X  X
7                          X  X
8                             X  X
```

12 cycles

C. [4 points] How many cycles does this sequence take to execute on a single-issue out-of-order machine? Again, show your work.

```
   1  2  3  4  5  6  7  8  9 10
1  X  X
2     X  X
3           X  X
4                 X  X
5        X  X
6              X  X
7                    X  X
8                          X  X
```

10 Cycles

D. [3 points] How many cycles does this machine take to execute on a multiple-issue out-of-order machine? Show your work.

```
   1  2  3  4  5  6  7  8
1  X  X
2  X  X
3        X  X
4              X  X
5  X  X
6        X  X
7              X  X
8                    X  X
```

8 Cycles

E. [3 points] Can you statically reorder the code to give the in-order machines the same performance as the out-of-order machines? If so, show the new ordering.

Yes! One possible ordering is 1,2,5,3,6,4,7,8 or:

```
1      LD     X,R1
2      LD     Y,R2
5      LD     Z,R4
3      ADD    R1,R2,R3
6      ADD    R2,R4,R5
4      ADD    R3,1,R8
7      MUL    R5,3,R6
8      ADD    R6,1,R7
```

Problem 4: Pipeline Microarchitecture and Performance [10 points]

Suppose you have a CPU with a 5 stage pipeline containing the following stages:

F: instruction fetch
R: register read (and instruction decode)
A: execute ALU operations (including determination of branch address and condition)
M: memory load and store
W: write back to register file

On this architecture, conditional branches account for 20% of all instructions, and all other instructions complete in a single cycle. That is, there are no pipeline stalls except for branches.

A. [2 points] With no speculation or prediction, what is the latency (in cycles) of a conditional branch operation on this CPU?

3 cycles

B. [3 points] What is the average CPI (cycles per instruction) for this CPU with no speculation or prediction of branches?

$CPI = (0.8)(1) + (0.2)(3) = 1.4$

C. [2 points] Suppose conditional branches are taken 60% of the time. What is the average CPI for this CPU if we speculatively predict all conditional branches as **not** taken?

$CPI = 0.8(1) + 0.2(0.4(1) + 0.6(3)) = 1.24$

D. [3 Points] Suppose we predict all conditional branches as taken and correctly predict the branch target address in the F stage 50% of the time. What is the average CPI for this case?

$CPI = 0.8(1)+0.2(0.4(3) + 0.6(0.5(1)+0.5(3))) = 0.8+ 0.48 = 1.28$

ANSWER SHEET
Comprehensive Examination in LOGIC
October 1999

MAGIC NUMBER: _____

	(A)	(B)	(C)	(D)
1.	Y	Y	N	Y
2.	N	Y	N	Y
3.	N	N	Y	Y
4.	Y	Y	Y	Y
5.	N	N	Y	N
6.	Y	N	N	N
7.	Y	N	N	N

	(A)	(B)	(C)	(D)
8.	N	Y	Y	Y
9.	N	Y	N	Y
10.	N	N	Y	Y
11.	N	N	N	Y
12.	Y	N	N	Y
13.	N	Y	Y	N

THE STANFORD UNIVERSITY HONOR CODE

A. The Honor Code is an undertaking of the students, individually and collectively:

(1) that they will not give or receive aid in examinations; that they will not give or receive unpermitted aid in class work, in the preparation of reports, or in any other work that is to be used by the instructor as the basis of grading;

(2) that they will do their share and take an active part in seeing to it that others as well as themselves uphold the spirit and letter of the Honor Code.

B. The faculty on its part manifests its confidence in the honor of its students by refraining from proctoring examinations and from taking unusual and unreasonable precautions to prevent the forms of dishonesty mentioned above. The faculty will also avoid, as far as practicable, academic procedures that create temptations to violate the Honor Code.

C. While the faculty alone has the right and obligation to set academic requirements, the students and faculty will work together to establish optimal conditions for honorable academic work.

I acknowledge and accept the Honor Code. (Signed) _____

Computer Science Department
Stanford University
Comprehensive Examination in Networks
99

November 4, 1999

READ THIS FIRST!

1. You should write your answers for this part of the Comprehensive Examination in a BLUE BOOK. Please use a separate blue book for each problem. Be sure to write your MAGIC NUMBER on the cover of every blue book that you use.

2. This exam is CLOSED BOOK.

Comprehensive Exam: Networks (60 points) **Autumn 1999**

1. (15 points total) *End to end.*

   (a) (5 points) Describe the "end-to-end" argument in networking, given one good example illustrating it.

   (b) (5 Points) Describe how it has influenced the design of the original Internet protocols, with one specific example.

   (c) (5 points) Describe how the IP security standard (IPsec) fits into the overall end-to-end story.

2. (15 points total) *Congestion*

   (a) (6 points) Al Gore, inventor of the Internet, lectured Monica Lewinsky that *congestion collapse* occurs from routers just getting tired from forwarding some many packets. However, even Monica didn't buy that line, or may be she slept through it. Describe what causes congestion collapse in reality, or is it just a fantasy of the liberal-biased media?

   (b) (6 points) Describe the characteristics and/or limitations that make the Aloha protocol suitable for some radio networks and CSMA-CD suitable for the original Ethernet, and not vice versa, with suitable justification.

   (c) (3 points) Bill Clinton decides to extend his managed healthcare initiative to cover the "health" of networks by legislating a new link-level flow control (LLFC) mechanism that is supposed to prevent congestion collapse. What are some concerns that the Republicans might legitimately raise about this approach.

3. (15 Points total) *Bandwidth and Delay*

   (a) (5 points) Suppose I need to move M megabytes of data from San Francisco to New York. One option is leasing a 1.5 Mbps Internet connection; Another is to write the data to 1.5 Megabyte floppy disks and hop on the plane for the 5 hour trip to New York. How big does M have to be for it to be faster to take the plane (stating whatever additional assumptions you need/make)?

   (b) (4 points) What factors could favor using the lease line over the plane trip, and vice versa, in reality.

   (c) (6 Points) Some routing experts advocate providing a bandwidth-delay product worth of buffering per port in the router, with delay referring to roundtrip time delay. For example, with 1.5 Mbps links and 25 millisecond round-trip time, the router should have roughly 4.7 kilobytes per port. Are these guys just trying to sell memory or what? Describe the legitimate argument for this amount of memory, if any, and how that relates to the memory required at the end points.

4. (15 Points Total) *Routing*

   (a) (6 points) Describe how distance vectoring (DV) routing works.

X
60

(b) (5 points) Describe the bad behavior that is inherent with DV routing with a specific example, and one solution that have been proposed/used to deal with it.

(c) (4 points) Describe how link-state routing works, in essence, how it avoids the problems of DV routing, and at what costs.

5. (1 Point) Name 3 top networking experts whose first name is David.

The End — I'm clearly out of questions.

Computer Science Department
Stanford University
Comprehensive Examination in Numerical Analysis
Autumn 1999

November 3, 1999

READ THIS FIRST!

1. You should write your answers for this part of the Comprehensive Examination in a BLUE BOOK. Be sure to write your MAGIC NUMBER on the cover of every blue book that you use.

2. This exam is CLOSED BOOK. You may not use notes, articles, or books.

1. (15 pts.) Give a direct proof that Euler's method is a convergent algorithm for the initial vaue problem $y' = ay$, $y(0)$ given, for $0<=t<=T$. By direct it is meant that no theorems are to be used.

2. (15 pts.) Use the error formula for polynomial interpolation to construct an error formula for the trapezoidal rule approximating the integral of $f(x)$ from $x=a$ to $x=b$. This is for the basic trapezoidal rule, NOT the composite trapezoidal rule.

Computer Science Department
Stanford University
Comprehensive Examination in Software Systems
Autumn 1999

November 2, 1999

READ THIS FIRST!

1. You should write your answers for this part of the Comprehensive Examination in a BLUE BOOK. Be sure to write your MAGIC NUMBER on the cover of every blue book that you use.

2. Be sure you have all the pages of this exam. There are 2 pages.

5. This exam is OPEN BOOK.

Comprehensive Exam — Systems software

Fall 1999

Short-answer (32 points)

Answer each of the following questions, and give a sentence or two justification for your answer (4 points each).

1. Without "free" (deallocate) it is easy to write a "malloc" implementation that never fragments memory. (True/false)

2. You profile a web server and notice that over the course of a day the virtual memory it allocates and uses is 1000 larger (measured in bytes) than the system's page cache. The cache cannot provide much benefit for this type of workload. (True/false)

3. Even without locks, a system can deadlock. (True/false)

4. Why do page-based systems use the high bits in a virtual address to specify the virtual page number?

5. Give two examples of when an OS intentionally causes internal fragmentation and say why.

6. You implement a distributed file system that allows clients to cache file blocks. You notice that decreasing the block size results in less network traffic when you revoke a cached block. What's a likely reason for this?

7. Ignoring the overhead of context switching, will I/O utilization on a round-robin scheduling system increase or decrease as the time-slice length is increased?

8. Ignoring context switching overhead, can increasing the time-slice length on a round-robin scheduling system dramatically improve a process's execution time?

Problem 2 — File system atomicity (18 points)

The Happy-go-lucky file system (HFS) moves a file from one directory to another by:

1. finding a free slot in the destination directory and inserting the file-name-to-inode mapping in this slot;

2. deleting the file-name-to-inode mapping in the source directory;

3. returning to the user;

4. at some later point HFS writes the cached copies of the modified directory blocks and the inode to disk (in no particular order).

Recall that disks can only atomically write a sector at a time.

1. (9 points) The system crashes. Give two possible file system inconsistencies that you might see.

2. (9 points) In a few sentences, describe how to use synchronous disk writes and file system reconstruction to eliminate these two inconsistencies. (Recall, "synchronous disk write" means when the OS issues a disk write, it waits until the write completes before performing another disk operation.)

Problem 3 — Synchronization (10 points)

The following sequential code adds and removes characters from an infinite buffer ("buf"). Rewrite the code *without using locks* to work correctly when there are exactly two threads: one producer thread that adds characters, and one consumer thread that removes them. State what assumptions your code makes and provide a short, intuitive correctness argument for your modifications.

```
char buf[]; /* infinite buffer */
int head = 0,   /* producer's position in buf */
    tail = 0,  /* consumer's position in buf */
    n = 0; /* number of characters in buf */

char get(void) {
        /* no characters = infinite loop */
        assert(n > 0);
        /* take character */
        c = buf[tail];
        tail++;
        n--;
        return c;
}

void put(char c) {
        buf[head] = c;
        head++;
        n++;
}
```