CS 473
Final Exam
December 13, 2002

The following exam is open book and open notes. You may feel free to use whatever additional reference material you wish, but **no electronic aids** are allowed. Please note the following instructions. There will be a ten point deduction for failure to comply with them:

- start each problem on a new sheet of paper

- write your social security number, but not your name, on each sheet of paper you turn in

- show your work whenever appropriate. There can be no partial credit unless I see how answers were arrived

- be succinct. You may lose points for facts that, while true, are not relevant to the question at hand

You have until 10:00 AM to finish the exam.

1. (20 points)

    (a) Multiply the following two IEEE floating point numbers together (using the floating point multiplication algorithm):
    $3fe00000_{16}$ * 40400000_{16}.

    (b) Convert your result into human-readable decimal format.

2. (15 points) Draw a Gantt chart showing how the following code will be executed on a MIPS processor, assuming all possible forwarding, stalling on `lw` when necessary, and a 1-cycle branch delay (no delayed loads or branches). Use arrows to show the forwarding between instructions.

```
            lw     $1, 100($2)
            addi   $3, $1, 1
            beq    $0, $0, label
            lui    $6, 12
            add    $1, $2, $3
    label:  lw     $2, 400($1)
            subi   $6, $3, 200
            lw     $12, 64($6)
            sw     $12, 20($5)
```

3. (10 points) Draw a Gantt chart showing how the following code will be executed on a CDC, using the assumptions from the notes.

```
    X1 <- X2 * X3
    X4 <- X5 * X6
    X1 <- X0 / X7
    X3 <- X4 * X1
    x4 <- X5 + X6
    X5 <- X1 + X3
```

4. (35 points) Suppose a computer system, using a 32 bit physical address, has a 16KB, 4-way set-associative cache with a 16 byte block size.

   (a) How would a physical address be broken down for cache lookups?

      Now suppose this same computer system also has a 32 bit virtual address, a virtual memory system just like Intel's, and a 64-entry, 4-way set-associative TLB.

   (b) How would a virtual address be broken down for TLB lookups?

   (c) Would it be possible to perform TLB and cache lookups simultaneously in this computer?

      Here are some relevant contents of the computer's TLB, cache, memory and the PDBR:

TLB

Index	Tag	Valid	Contents
7	22fa	1	000c2025
7	5767	0	5ca8b39e
7	696c	0	01a2fe7f
7	696c	1	0009a043
b	22fa	1	000d7003
b	5767	0	000c2025
b	5767	1	5ca8b39e
b	696c	0	0039707f
c	22fa	1	0039707f
c	22fa	1	000d7003
c	696c	0	000c2025
c	696c	1	00149025

Cache

Index	Tag	Valid	Contents (4 bytes at given offset)			
			c	**8**	**4**	**0**
11	00397	0	b55d3fd1	2cb265fc	729df029	79f0c9a5
11	0acab	0	a1112079	7e0f2618	d2225f7e	3cadc0d5
11	28fd3	0	6a7fd81e	e2fea616	eacf5cf6	71201139
11	77223	1	4eaee81d	06967fac	fd96b0a3	7ee0fbcb
47	00149	0	89ccea47	96d701e8	84ce6123	5839bff8
47	2bc3b	1	01457a16	046af216	1658f4ad	05691ab3
47	6d224	0	00f1a2cd	3cfb6a72	af09eedd	bd6f7564
47	7b483	0	8fe23130	fde64f24	2c6e3d14	5a92f7b0
c1	000d7	1	4ca424f7	a30ce94f	76f7b80c	ebc1bee0
c1	6c45e	0	0a87c67c	4adeb8a5	42b931ca	5acc9105
c1	759e2	0	c43123a1	0e3e097d	40a6a704	12d0d9bd
c1	78be5	1	34bbc2cf	ff113e65	ebc1bee0	e33fea7a

Memory

Address	Data
0004c9dc	0039707f
0009aeac	000d7003
000c2b30	00149025
000d7c14	76f7b80c
0012a22c	0009a043
0012a574	0004c067
0012a694	000c2025
0014947c	5ca8b39e
0039711c	01a2fe7f

PDBR = 0012a000

   (d) For each of the following attempted memory operations, what happens? Be sure to say whether TLB is a hit or a miss (and why, if a miss) cache is a hit or a miss (and why, if a miss), whether there are any protection violations, and whether there is a page miss. If the result is that data is returned to the program, give the value returned (assume a 4-byte read). If you have a page fault or a protection violation, there's no need to look in the cache. *Remember, the cache uses physical addresses.*

	Address	Mode	Operation
i.	696cc47c	User	Write
ii.	22fabc14	Kernel	Read
iii.	5767711c	Kernel	Write

5. (20 points) A particular computer system is capable of executing 2 billion instructions per second. It has a disk drive that has an average seek time of 10 msec (a bit slow by today's standards, but easy to work with), and turns at 6000 RPM (also a bit slow, but also easy to work with). The disk is capable of transferring 20 million bytes per second. This computer system is executing a program that requires it to repeatedly execute transactions requiring the following sequence of operations:

- Spend 1 million instructions calculating its next sequence of reads and writes.
- Read 5 blocks from random locations on the disk. Each block is 4000 bytes.
- Spend 10 millions instructions processing the data.
- Write one (4000 byte) block to a random location on disk.

(a) How many of these transactions can this system execute per second?

(b) You are given the option of doubling the performance of one of the following aspects of the system: the processor speed, the disk RPM, or the disk transfer rate. Which one should you pick? Why, in the practical world, is that the hardest one of the three to improve?