

**CS 573**  
**Final Exam**  
**May 12, 2006**

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 to finish the exam. The questions are equally weighted.

1. One of the most striking features of the Cray-1 instruction set was its use of vector instructions, which would execute an operation on corresponding elements of arrays. So, for instance,

```
vmul V1, V2, V3
```

would effectively execute the loop

```
for (i = 0; i < 64; i++)  
    V1[i] = V2[i]*V3[i];
```

- (a) Why did these instructions give the Cray-1 a performance advantage?
  - (b) Would they still result in an advantage for a modern computer capable of out of order execution, such as a MIPS R10000 or a Pentium 4?
2. In an Alewife system variable  $x$  is homed on processor P1, and initially has a processor set  $P = \{1\}$ . Suppose the following sequence of memory operations takes place:

<b>P2:</b>	$w(x)_1$	
<b>P3:</b>	$r(x)_1$	$w(x)_2$

What are the messages that must be passed among the processors to accomplish this?

3. Suppose the following programs are executed on a multiprocessor using an MESI snoopy cache protocol, in the order shown (so P1 executes its program, then P2 executes its program). What will the states of the variables be in the two processors' caches after this? Assume the three variables map to different cache blocks, and initially the caches are all empty. I realize  $z$  is being used uninitialized; that's OK.

**P1:**  $y = 1;$   
**P2:**  $x = y + z;$

4. Here are some terms that appeared in the wormhole routing paper. What do they mean?
  - (a) Minimal (in the sense of a minimal routing algorithm)
  - (b) Deadlock

5. Consider the following three one-line programs:

**P1**            **P2**            **P3**  
 $x = 1;$      $y = 2;$      $z = x + y;$

- (a) Using the notation we've been using to show memory operations (as in Problem 2), show that it is possible, under lazy release consistency, for **P1** and **P2** to execute before **P3** but for  $z$ 's final result to be 2. Assume  $x$ ,  $y$ , and  $z$  are all in separate cache blocks.
- (b) Now add appropriate barrier operations to ensure that  $z$ 's final value is 1. Show both the modified programs and the new memory operations diagram.
6. The following 13 bits contain eight bits of data, 4 check bits, and one overall parity bit (so it's a single-error correcting, double-error detecting code). The table shows the bit numbering, the definition of the bit in each position (C for Check, D for Data, or P for Parity)

1	2	3	4	5	6	7	8	9	10	11	12	13
C1	C2	D1	C3	D2	D3	D4	C4	D5	D6	D7	D8	P
0	1	1	0	1	0	0	1	0	0	1	0	0

This data may be correct, may have a one-bit error, or may have a two-bit error. Which is it? If it's a 1-bit error, what is the correct data?