

CS 573

Midterm Exam

February 27, 2002

The following exam is open book and open notes. You may feel free to use whatever additional reference material you wish, but **no calculators** 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. I will take off points for facts that, while true, are not relevant to the question at hand

You have until 12:30 to finish the exam. The questions are equally weighted.

1. A frequent problem in the development of new instruction set architectures is the need to avoid alienating an existing customer base. For each of the following architectures, very briefly discuss, the importance of backward compatibility, the approach taken to maintaining backward compatibility, and the degree to which the approach was successful: (a) VAX, (b) RISC 1, (c) Alpha, (d) IA-64.
2. Consider the following fragment of C code.

```
if (d != 3) {  
    a = *ptr + b;  
} else {  
    a = b;  
}
```

Hand-compile this code so it will run as quickly as possible on an IA-64. Assume the variables a, b, and ptr are in registers. You will want to use speculative loads and predication in your solution.

3. The classic algorithm for multiplying two square arrays is given by

```
for (i = 0; i < SIZE; i++)
  for (j = 0; j < SIZE; j++)
    for (k = 0; k < SIZE; k++)

      C[i][j] = C[i][j] + A[i][k]*B[k][j];
```

Explain why it is necessary to use a stride of greater than one to vectorize the inner-most loop for efficient execution on a Cray-1.

4. Rank the following three situations according to the importance of precise interrupts in each. Argue that you have selected the correct order: (a) interrupts caused by external devices, (b) interrupts by page faults, (c) interrupts caused by arithmetic exceptions (such as divide-by-zero).