Department of Computer Science
Qualifying Examination
Computer Architecture
Fall, 2001

The following exam is open book and open notes. You may feel free to use whatever additional reference material you wish. Please note the following instructions.

- show your work whenever appropriate. There can be no partial credit unless I see how answers were obtained.
- be succinct. I will take points for facts that, while true, are not relevant to the question at hand.
- The questions are equally weighted (the total number of points is 90).

1. (30 points) Many architectures draw a distinction between external interrupts (interrupts caused by peripheral devices needing service) and exceptions (interrupts caused by events occurring within a program). An example of an external interrupt would be a disk drive with data ready to transfer; an example of an exception would be a segmentation fault. Ordinarily, any interrupt (be it an external interrupt or an exception) causes the execution of an interrupt handler in kernel mode.

   Another possibility would be to let some exceptions be handled in user mode by the process that caused them, rather than going to kernel mode to handle them. What advantages would such a scheme have? Can you think of some specific exceptions which could not be handled in user mode?

2. (30 points) Consider the following fragment of C code:

   ```
   if (a < b)
       d = *c;
   else
       *c = 7;
   ```

   Explain how speculation and predication can be used to make the compiled version of this code run more quickly. Assume all variables are in registers.

3. (30 points) Consider a bus-based multiprocessor computer system with an MSI (modified-shared-invalid) cache coherence scheme and four processors.

   (a) Could a standard writeback uniprocessor cache controller be used in implementing this scheme? If so, how would each of the three multiprocessor cache states map to uniprocessor cache states?

   (b) Suppose the processors in the system execute the sequence of reads and writes given in the following table. After each operation, what is the state of each of the variables in all four caches, and is the copy of the variable in memory valid or invalid? Assume that each variable is in a different cache block, and that the initial conditions are as shown in the table. The operations refer to CPU operations, so (for instance) a read may or may not cause a read from memory, depending on whether there is a cache hit or a miss. The "flush" operation means that the processor has decided to remove the specified data from its own cache (which may or may not have impacts on the other caches and memory). You may use the following table for your answers.

Operation	P1			P2			P3			P4			Memory		
	X	Y	Z	X	Y	Z	X	Y	Z	X	Y	Z	X	Y	Z
(initial state)	I	I	I	I	I	I	I	I	I	I	I	I	V	V	V
P1: read X															
P2: read Y															
P1: write X															
P3: write Z															
P4: write Y															
P2: read X															
P2: flush X															
P4: flush Y															