Department of Computer Science
Computer Architecture Qualifying Exam
Spring, 2008

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:

- show your work whenever appropriate. There can be no partial credit unless you show how you derived your answers
- be succinct. You may lose points for facts that, while true, are not relevant to the question at hand

1. (30 points) Some computer system uses a snoopy cache protocol for memory consistency.

   The computer has a 32-bit (four byte) word size, and a 64-bit (eight byte) cache line. It has separate 4K byte direct-mapped instruction and data caches, both of which are initially empty.

   Processor P1 executes the following code:

   ```
   for (i=0; i < 100; i++)
       a[i] = a[i]+1;
   ```

   (a) Estimate the data cache hit rate. You may assume all address arithmetic is performed in registers, and that variable `i` is maintained in a register (so only reads and writes of array elements need to go to the memory system).

   (b) If the snoopy cache protocol is MSI (*i.e.* the cache block states are `Modified`, `Shared`, and `Invalid`), what proportion of the array element reads and writes require memory bus transactions (notice that some reads and writes that are not misses do require memory bus transactions)?

   (c) How would using an MESI protocol (*i.e.* having an additional `Exclusive` state) affect the hit rate, if at all? The number of bus transactions?

2. (30 points) Some computer system uses a two-level adaptive branch prediction scheme, as follows:

   There is a per-address branch history table (BHT), keeping track of the last eight executions of every branch instruction in the program, with a `1` representing a branch-taken and a `0` representing a branch-not-taken. All of the BHT entries are initialized to `11111111`.

   The current contents of an address's BHT are used to index a per-address Pattern History Table (PPHT).

   A two-bit saturating counter in each PPHT entry is used for the actual branch prediction: the counter is incremented on a branch-taken, and decremented on a branch-not-taken. A counter value of `10` or `11` predicts taken; a counter value of `00` or `01` predicts not-taken. All the counters are initialized to `10`.

   This is, of course, the scheme identified as PAp, combined with automaton A2, in the Yeh&Patt branch prediction paper.

   The following code is executed:

   ```
   for (i = 0; i < 100; i++)
       for (j = 0; j < 100; j++)
           a[i] = a[i+1];
   ```

   Compute the accuracy of this predictor for conditional branches for the inner loop, for the outer loop, and for all the conditional branches in the code (this will, of course, be a weighted average of the other two). You should assume that executing the code in a loop requires a branch-taken, while falling out of a loop requires a branch-not-taken. You may express your answer as a fraction.

3. (15 points) The following string has an eight-bit word, with four error correction bits and an overall parity bit:

   ```
   0111010001011
   ```

   The eight data bits come first in the order $D_7D_6D_5D_4D_3D_2D_1D_0$, then the four correcting bits in the order $C_8C_4C_2C_1$, then the overall parity bit.

   If we are using single error correction, double error detection, does this data have an error? If so, how many bits? If there is a correctable error, correct it.

4. (15 points) One of the differences between CISC architectures like the Intel x86 and RISC architectures is that the RISC architectures have many more architected registers. In spite of this, modern RISC machines typically use register renaming, just like the CISC machines, due to particular registers being "hot spots".

   (a) What is meant by a hot spot, in this context?

   (b) Give an example of a register that's likely to be a hot spot in a RISC architecture.

5. (10 points) In a network with a hypercube topology, a packet is to be routed from node `0110` to node `1010` using least-significant-bit-first routing. What is the sequence of nodes it goes through on the way?