

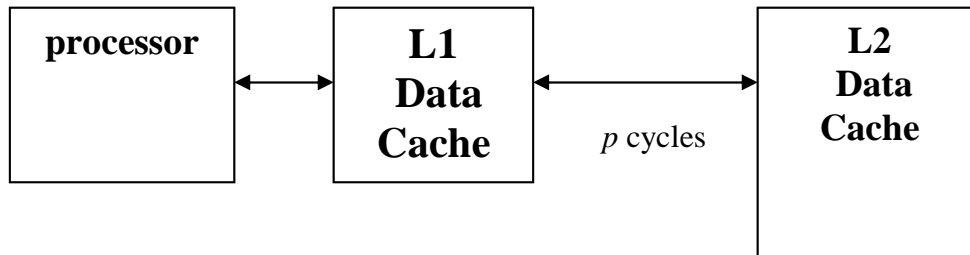
**University of Illinois at Urbana-Champaign
Department of Electrical and Computer Engineering**

**Quiz 1
ECE 511, Fall 2004**

You have until 6pm Friday, October 8th to complete this quiz. By turning in this quiz, you will have attested that you have neither received nor given inappropriate aid on this quiz from any person except the instructor. You may use class notes, textbooks, web resources, computer simulations, and published materials.

NAME:

1. (10 pts) The following figure describes the data cache system for a particular microprocessor. We know that the processor consists of a simple in-order pipeline that blocks on L1 cache misses. We know very little about the caches of this system except that the L1 cache, if it is associative, uses the Least Recently Used (LRU) replacement policy. We also know that the latency between the L1 and L2 cache is p cycles.

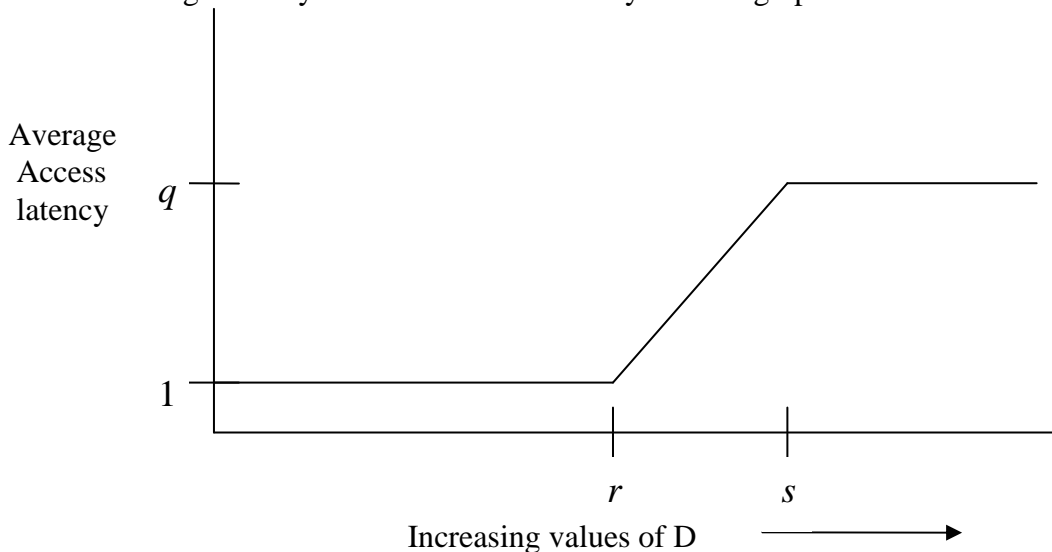


In order to discover the size and organization of the L1 data cache, we have decided to run a small code kernel on it, which is described in pseudo code below. The code sequentially accesses D elements of an array called A repeatedly 100 times.

```

do 100 times {
    for i = 0 to D-1 {
        access A[i];
    }
}
  
```

By running this small loop for various values of D , we are able to obtain a graph of the average latency for an access to the array A . The graph is shown below.



Using the values q , r , and s derived from this graph, along with p (L2 cache latency), determine the following properties of the L1 cache: (a) hit latency, (b) size, (c) line size, (d) associativity.

2. (10 pts.) In class on September 13 I said that if the BTB has a miss but the branch predictor predicts taken, you have no choice but to fetch PC+4, but that the branch predictor was probably correct and the BTB incorrect. Luckily you all ignored my suggestion when you did Homework 2. Explain why I was wrong.

3. (20 pts.) You are designing the address generation unit for a new processor, and your lab partner, Ed Davidson, suggests that rather than updating the BTB's LRU information each time a branch is found in the BTB we should only mark a branch entry as Least Recently Used in the BTB when the branch retires and is seen to be taken. What are the relative advantages and disadvantages of this idea?

4.(20 pts.) You are working on a design with a single cycle direct mapped instruction cache, but where the instruction cache miss penalty averages 25 cycles. You have discovered that your direct mapped instruction cache has a pretty bad miss rate. Your lab partner, Maurice Wilkes, suggests that you could replace the direct mapped cache with a *pseudo*-associative cache: instead of hashing the PC to find a cache line and then comparing tags, you could store a particular block of instructions in *any* of the cache lines (as with a fully associative cache), but then in the BTB you would store a prediction of the corresponding cache line number along with the target address for each branch. What are the relative advantages and disadvantages of this idea?

5. (20 pts.) You are the chief architect at Illin Corp, the makers of a processor (called the Illin E511) that issues instructions out-of-order, but does no register renaming. The scoreboard enforces true- (RAW), anti- (WAR) and output- (WAW) dependences by using the following three rules: An instruction is ready to issue only if (1) All previous instructions that write its source registers have completed, (2) All previous instructions that write its destination register have completed, (3) All previous instructions that read its destination register have issued.

A young summer intern in your group named Seymour Cray points out that rule (2) is overly conservative. The instruction in question doesn't really need to wait until the previous writer of its destination completes, it just needs to wait long enough to ensure that the writes happen in the correct order. For example, if multiplies take 10 cycles and adds take 5 cycles then in the instruction sequence:

ADD R13 <= R10 + R2

MUL R13 <= R1 * R5

It would really be okay to issue the add and then issue the multiply in the very next cycle. The add will write its result at the end of the sixth cycle, the multiply will write its result at the end of the twelfth cycle.

Would it be a good idea to use Seymour's new rule or a bad idea? What extra information would you need to keep in the scoreboard? Are output- dependences a cause of poor performance on your machine (explain)?

6. (20 pts) Consider all the places that a physical register tag could be stored in the MIPS R1000 like pipeline we have been considering in class. In particular, a physical register tag could be listed (1) in the register free-list (FL), (2) in the physical destination register field of one entry in the reorder buffer (ROB), (3) in the register alias table (RAT) or (4) in the retirement register alias table (RRAT). What are the possible legal combinations of places that a physical register tag could be stored, and under what conditions will the register tag transition from one set of places to another. (You may want to start out by considering the following questions: Is it possible that a physical register tag could be stored in *none* of these four places? Is it possible that a physical register tag could be stored in both the RRAT and the ROB? Is it possible that a physical register tag could be stored in both the RAT and the RRAT?)