Computer Architecture and Engineering

CS152 Quiz #2

March 3rd, 2008

Professor Krste Asanovic

Name:_____

This is a closed book, closed notes exam.

80 Minutes

10 Pages

Notes:

- **Not all questions are of equal difficulty, so look over the entire exam and budget your time carefully.**
- **Please carefully state any assumptions you make.**
- **Please write your name on every page in the quiz.**
- **You must not discuss a quiz's contents with students who have not yet taken the quiz. If you have inadvertently been exposed to the quiz prior to taking it, you must tell the instructor or TA.**
- **You will get no credit for selecting multiple choice answers without giving explanations if the instructions ask you to explain your choice.**

Writing name on each sheet	**_____**	**1 Point**
Question 1	**_____**	**29 Points**
Question 2	**_____**	**18 Points**
Question 3	**_____**	**15 Points**
Question 4	**_____**	**15 Points**
TOTAL	**_____**	**80 Points**

NAME:_____

Problem Q.2.1: Victim Cache Evaluation [29 Points]

Although direct-mapped caches have an advantage of smaller access time than set-associative caches, they have more conflict misses due to their lack of associativity. In order to reduce these conflict misses, N. Jouppi proposed *victim caching* where a small fully-associative back up cache, called a victim cache, is added to a direct-mapped L1 cache to hold recently evicted cache lines.

The following diagram shows how a victim cache can be added to a direct-mapped L1 data cache. Upon a data access, the following chain of events takes place:

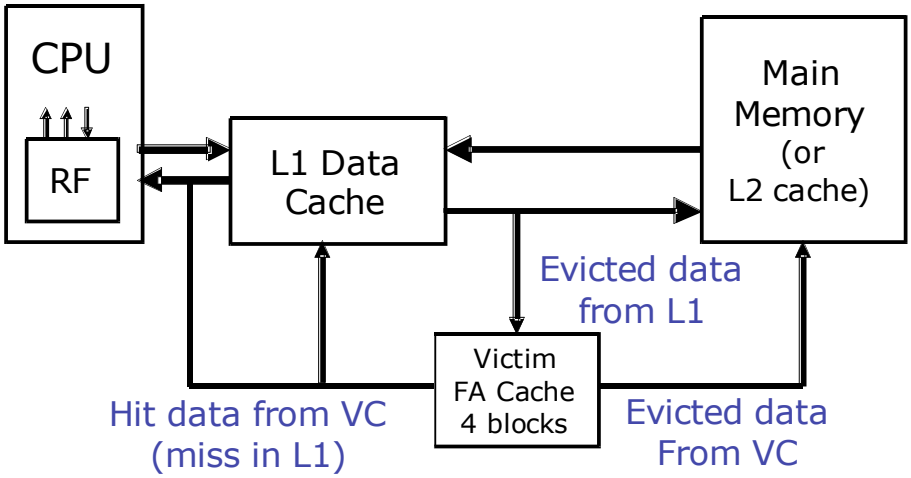


Figure Q2.1-A: A Victim Cache Organization

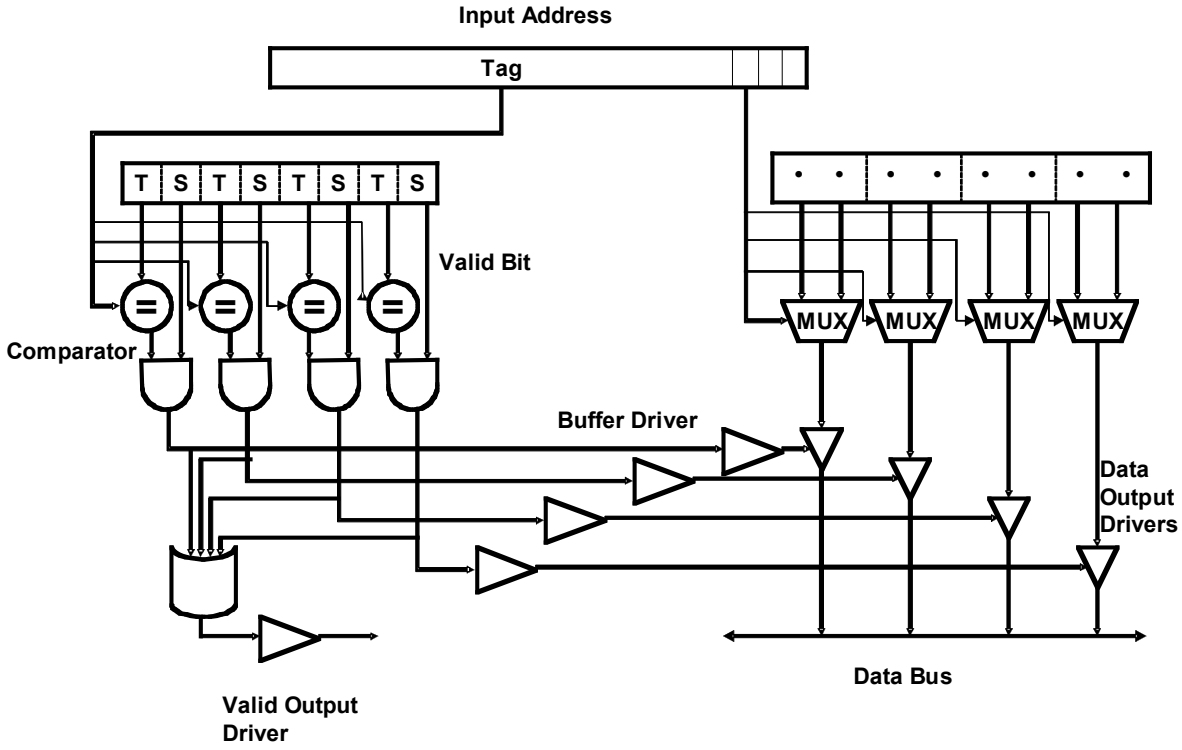
1. The L1 data cache is checked. If it holds the data requested, the data is returned.
2. If the data is not in the L1 cache, the victim cache is checked. If it holds the data requested, the data is moved into the L1 cache and sent back to the processor. The data evicted from the L1 cache is put in the victim cache, and put at the end of the FIFO replacement queue.
3. If neither of the caches holds the data, it is retrieved from memory, and put in the L1 cache. If the L1 cache needs to evict old data to make space for the new data, the old data is put in the victim cache and placed at the end of the FIFO replacement queue. Any data that needs to be evicted from the victim cache to make space is written back to memory or discarded, if unmodified.

Note that the two caches are *exclusive*. That means that the same data cannot be stored in both L1 and victim caches at the same time.

NAME:_____

Problem Q2.1.A **Baseline Cache Design**
10 Points

The diagram below shows our victim cache, a 32-Byte fully associative cache with four 8-Byte cache lines. Each line contains of two 4-Byte words and has an associated tag and two status bits (valid and dirty). The Input Address is 32-bits and the two least significant bits are assumed to be zero. The output of the cache is a 32-bit word.



Please complete Table Q2.1-1 on the next page with delays across each element of the cache. Using the data you compute in Table Q2.1-1, calculate the critical path delay through this cache (from when the Input Address is set to when both Valid Output Driver and the appropriate Data Output Driver are outputting valid data).

NAME:_____

Component	Delay equation (ps)		FA (ps)
Comparator	200×(# of tag bits) + 1000		
N-to-1 MUX	$500 \times \log_2 N + 1000$		
Buffer driver	2000		
AND gate	1000		
OR gate	500		
Data output driver	500×(associativity) + 1000		
Valid output driver	1000		

Table Q2.1-1

Critical Path Cache Delay: _____

Show your work:

NAME:_____

Problem Q2.1.B **Victim Cache Behavior**
12 Points

Now we will study the impact of a victim cache on cache hit rate. Our main L1 cache is a 128 byte, direct-mapped cache with 16 bytes per cache line. The cache is word (4-bytes) addressable. The victim cache in Figure Q2.1-A is a 32-byte fully associative cache with 16 bytes per cache line, and is also word addressable. The victim cache uses the first in first out (FIFO) replacement policy.

Please complete Table Q2.1-2 on the next page showing a trace of memory accesses. In the table, each entry contains the {tag,index} contents of that line, or "inv", if no data is present. You should only fill in elements in the table when a value changes. For simplicity, the addresses are only 8 bits.

The first 3 lines of the table have been filled in for you.

For your convenience, the address breakdown for access to the main cache is depicted below.

7	6		4	3	2	1	0
TAG	INDEX			WORD SELECT		BYTE SELECT	

Problem Q2.1.C **Average Memory Access Time**
5 Points

Assume **15%** of memory accesses are resolved in the victim cache. If retrieving data from the victim cache takes **5 cycles** and retrieving data from main memory takes **55 cycles**, by how many cycles does the victim cache improve the average memory access time?

NAME:_____

Input Address	Main Cache									Victim Cache		
	L0	L1	L2	L3	L4	L5	L6	L7	Hit?	Way0	Way1	Hit?
	inv	inv	inv	inv	inv	inv	inv	inv	-	inv	inv	-
00	0								N			N
80	8								N	0		N
04	0								N	8		Y
A0												
10												
C0												
18												
20												
8C												
28												
AC												
38												
C4												
3C												
48												
0C												
24												

Table Q2.1-2

NAME:_____

Problem Q2.2: Code and Data Rearrangement 18 Points

In this problem we will examine techniques for reducing cache miss rates that do not involve any modifications to the hardware. For each of the following problems, optimize the code segment presented and answer any questions.

Problem Q2.2.A **Loop Optimization 1**
6 Points

```
for(j=0; j < N; j++) {
     for(i=0; i < M; i++) {
        x[i][j] = 2 * x[i][j];
     }
 }
```

What type of locality does your modification improve? Explain.

NAME:_____

Problem Q2.2.B **Loop Optimization 2**
6 Points

```
for(i=0; i < N; i++)
    a[2*i] = b[2*i] * c[2*i];
for(i=0; i < N; i++)
    a[2*i+1] = b[2*i+1] + c[2*i+1];
```

What type of locality does your modification improve? Explain.

NAME:_____

Problem Q2.2.C **Loop Optimization 3**
6 Points

```
for(i=0; i < N; i++)
    a[i] = b[i] + c[i];
for(i=0; i < N; i++)
    d[i] = b[i] - c[i];
```

What type of locality does your modification improve? Explain.

NAME:_____

Problem Q2.3: Caches and Memory Access Patterns 15 Points

You have just accepted a position at Caches-R-Us as a research scientist. Your first task is to assess the pathological performance of various cache organizations. You decide to start by looking at two basic caches, both with a capacity of four words. The first is a direct-mapped cache with one word per cache line. The second is a fully-associative cache also with one word per cache line and an LRU replacement policy. For both of the following questions assume the caches are initially empty, i.e., all lines are invalid.

Problem Q2.3.A 6 Points

Please specify a memory access pattern that will cause the fully-associative cache to incur fewer misses than the direct-mapped cache.

Problem Q2.3.B 9 Points

Does there exist a memory access pattern that causes the direct-mapped cache to incur fewer misses than the fully-associative cache? If so, please give one such access pattern, or else explain why this is not possible.

NAME:_____

Problem Q2.4: Cache Parameter Short Answer 15 Points

For each of the following statements about making a change to a cache design, circle **True** or **False** and provide a one sentence explanation of your choice. Assume all cache parameters (capacity, associativity, line size) remain fixed except for the single change described in each question**. Please provide a one sentence explanation of your answer**.

Problem Q2.4.A **3 Points**

Doubling the line size halves the number of tags in the cache

True / False

Problem Q2.4.B **3 Points**

Doubling the associativity doubles the number of tags in the cache.

True / False

Problem Q2.4.C **3 Points**

Doubling cache capacity of a direct-mapped cache usually reduces conflict misses.

True / False

NAME:_____

Problem Q2.4.D **3 Points**

Doubling cache capacity of a direct-mapped cache usually reduces compulsory misses.

True / False

Problem Q2.4.E **3 Points**

Doubling the line size usually reduces compulsory misses.

True / False