CS152
Computer Architecture and Engineering

February 16, 2011

Caches and the Memory Hierarchy

Assigned February 16 Problem Set #2 *Due March 2*

---

http://inst.eecs.berkeley.edu/~cs152/sp11

The problem sets are intended to help you learn the material, and we encourage you to collaborate with other students and to ask questions in discussion sections and office hours to understand the problems. However, each student must turn in his own solution to the problems.

The problem sets also provide essential background material for the quizzes. The problem sets will be graded primarily on an effort basis, but if you do not work through the problem sets you are unlikely to succeed at the quizzes! We will distribute solutions to the problem sets on the day the problem sets are due to give you feedback. Homework assignments are due at the beginning of class on the due date. Late homework will not be accepted.

Problem 2.1: Cache Access-Time & Performance

This problem requires the knowledge of Handout #2 (Cache Implementations) and Lectures 6 & 7. Please, read these materials before answering the following questions.

Ben is trying to determine the best cache configuration for a new processor. He knows how to build two kinds of caches: direct-mapped caches and 4-way set-associative caches. The goal is to find the better cache configuration with the given building blocks. He wants to know how these two different configurations affect the clock speed and the cache miss-rate, and choose the one that provides better performance in terms of average latency for a load.

Problem 2.1.A — Access Time: Direct-Mapped

Now we want to compute the access time of a direct-mapped cache. We use the implementation shown in Figure H2-A in Handout #2. Assume a 128-KB cache with 8-word (32-byte) cache lines. The address is 32 bits and byte-addressed, so the two least significant bits of the address are ignored since a cache access is word-aligned. The data output is also 32 bits (1 word), and the MUX selects one word out of the eight words in a cache line. Using the delay equations given in Table 2.1-1, **fill in the column for the direct-mapped (DM) cache in the table**. *In the equation for the data output driver, 'associativity' refers to the associativity of the cache (1 for direct-mapped caches, A for A-way set-associative caches).*

Component	Delay equation (ps)		DM (ps)	SA (ps)
Decoder	200×(# of index bits) + 1000	Tag		
		Data		
Memory array	200×$\log_2$(# of rows) + 200×$\log_2$(# of bits in a row) + 1000	Tag		
		Data		
Comparator	200×(# of tag bits) + 1000			
N-to-1 MUX	500×$\log_2$N + 1000			
Buffer driver	2000			
Data output driver	500×(associativity) + 1000			
Valid output driver	1000			

Table 2.1-1: Delay of each Cache Component

What is the critical path of this direct-mapped cache for a cache read? What is the access time of the cache (the delay of the critical path)? To compute the access time, assume that a 2-input gate (AND, OR) delay is 500 ps. If the CPU clock is 150 MHz, how many CPU cycles does a cache access take?

Problem 2.1.B **Access Time: Set-Associative**

We also want to investigate the access time of a set-associative cache using the 4-way set-associative cache in Figure H2-B in Handout #2. Assume the total cache size is still 128-KB (each way is 32-KB), a 4-input gate delay is 1000 ps, and all other parameters (such as the input address, cache line, etc.) are the same as part 2.1.A. **Compute the delay of each component, and fill in the column for a 4-way set-associative cache in Table 2.1-1.**

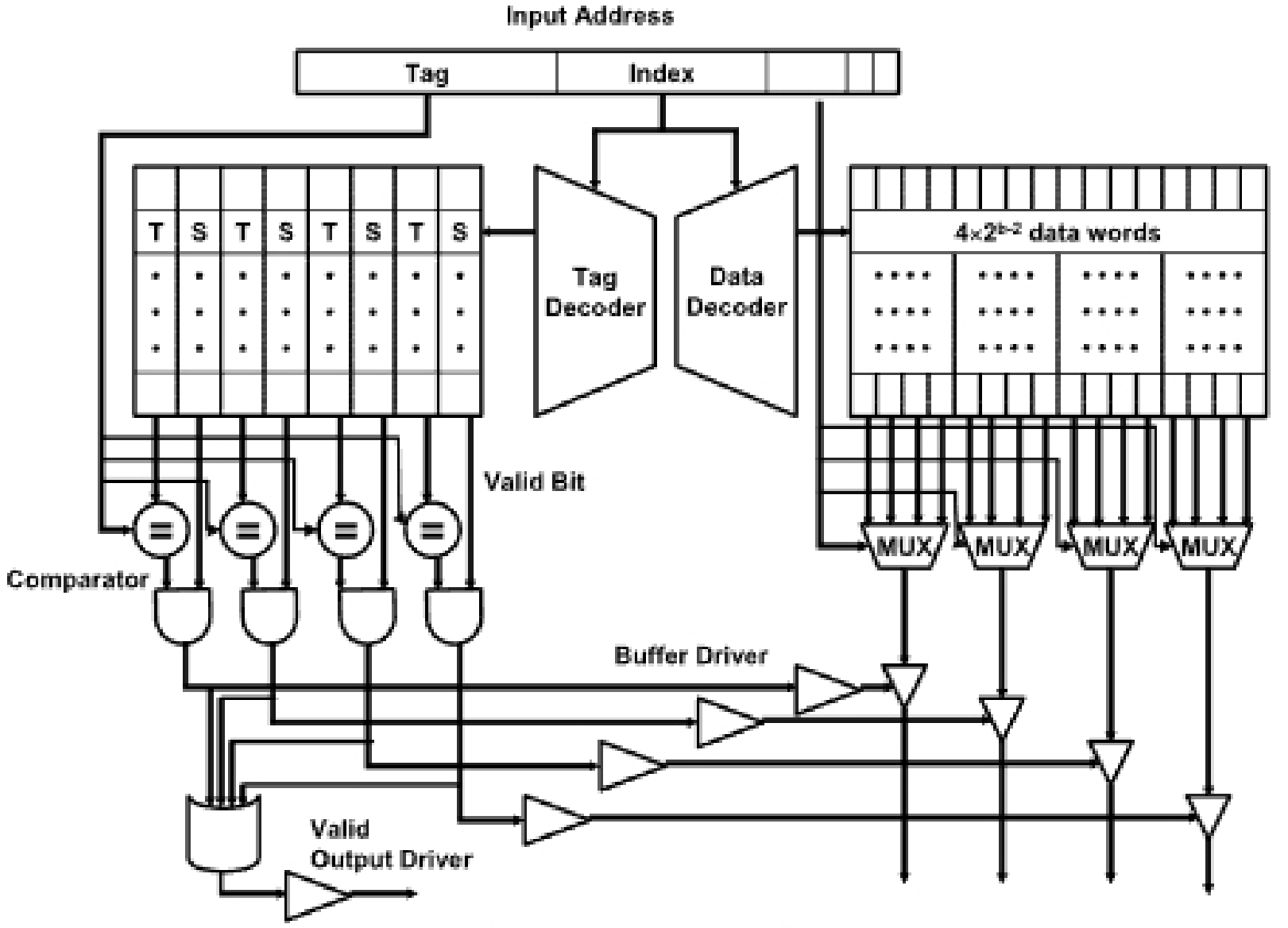

Figure 2.1: 4-way set-associative cache

What is the critical path of the 4-way set-associative cache? What is the access time of the cache (the delay of the critical path)? What is the main reason that the 4-way set-associative cache is slower than the direct-mapped cache? If the CPU clock is 150 MHz, how many CPU cycles does a cache access take?