Jose Johnson
1/2/2025

ECE 3058
Architecture, Systems, Concurrency, and Energy in Computation
Lab 3

Part A: Caches

Direct-mapped Cache

The following diagram shows how a direct-mapped cache is organized. To read a word from the cache, the input address is set by the processor. Then the index portion of the address is decoded to access the proper row in the tag memory array and in the data memory array. The selected tag is compared to the tag portion of the input address to determine if the access is a hit or not. At the same time, the corresponding cache block is read and the proper line is selected through a MUX.

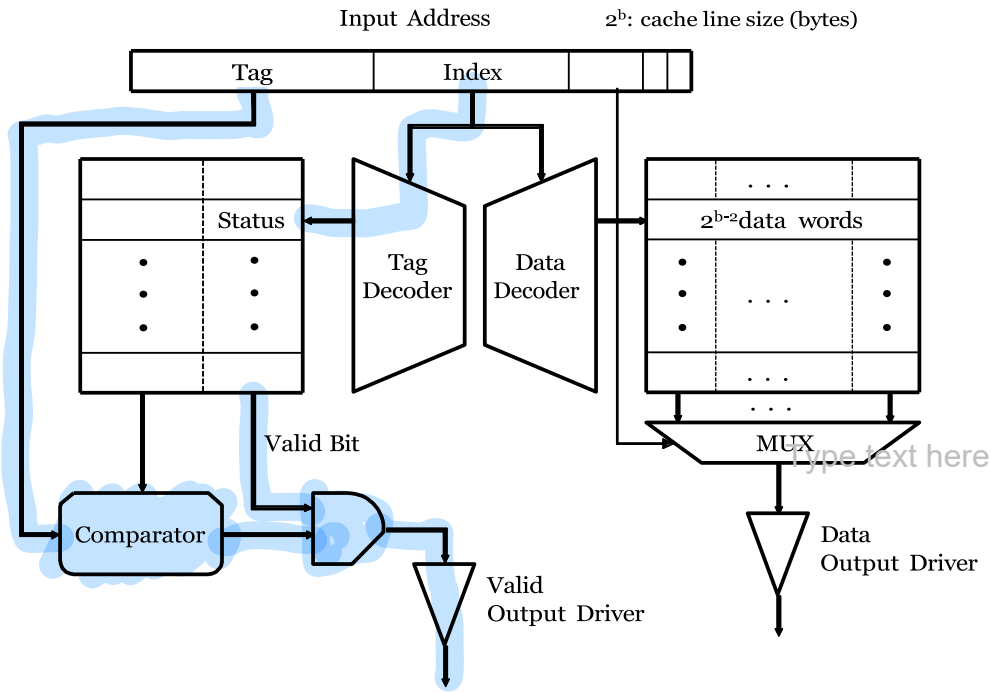

Figure A.1: A direct-mapped cache implementation

In the tag and data array, each row corresponds to a line in the cache. For example, a row in the tag memory array contains one tag and two status bits (valid and dirty) for the cache line. For direct-mapped caches, a row in the data array holds one cache line.

Four-way Set-associative Cache

The implementation of a 4-way set-associative cache is shown in the following diagram. (An n-way set-associative cache can be implemented in a similar manner.) The index part of the input address is used to find the proper row in the data memory array and the tag memory array. In this case, however, each row (set) corresponds to four cache lines (four ways). A row in the data memory holds four cache lines (for 32-bytes cache lines, 128 bytes), and a row in the tag memory array contains four tags and status bits for those tags (2 bits per cache line). The tag memory and the data memory are accessed in parallel, but the output data driver is enabled only if there is a cache hit.

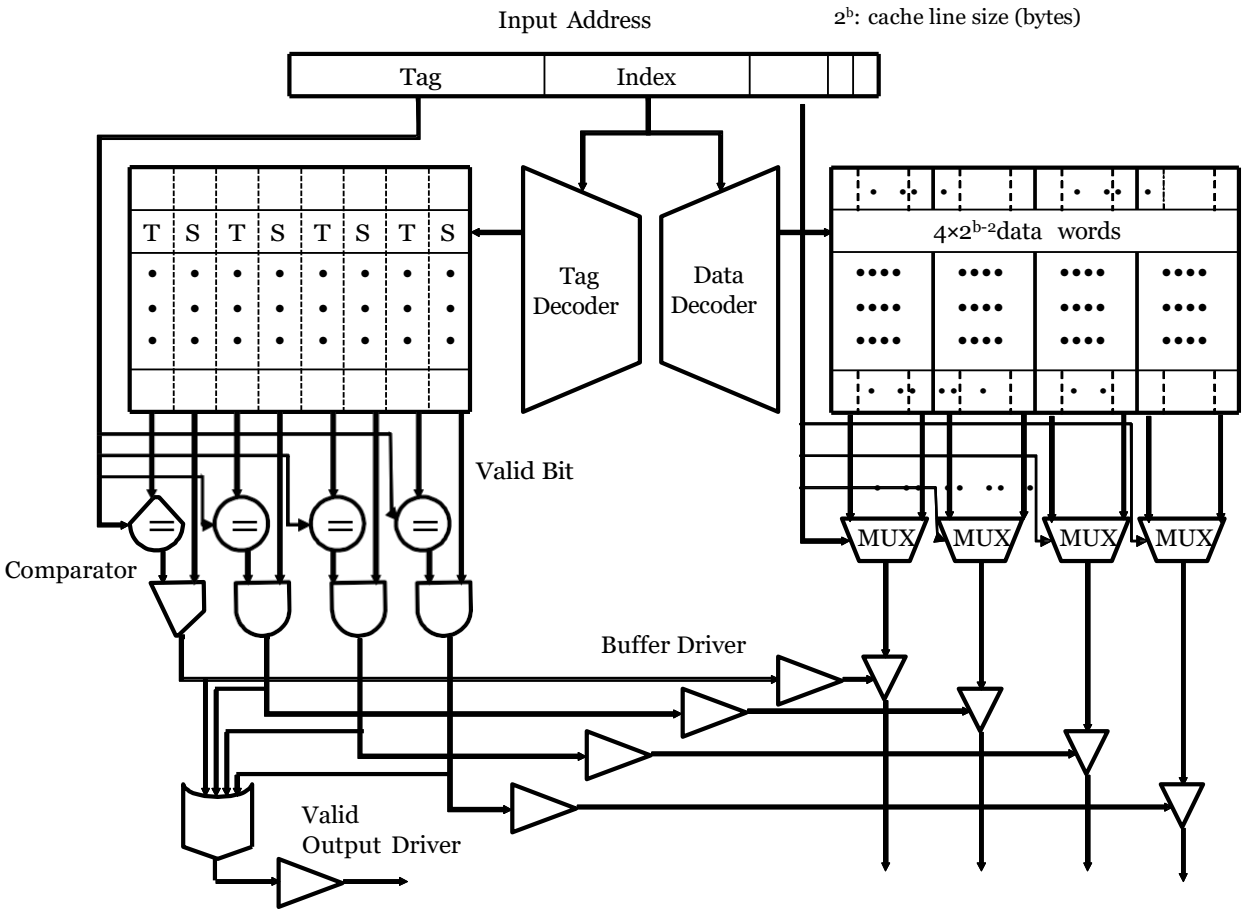

Figure A.2: A 4-way set-associative cache implementation

Cache Type	Data Array	Num Sets	Num of index bits	Num of tag	Tag Array
Direct Mapped	128KB	4096 $\frac{(128 \times 2^{10})}{32}$	12 a=1 $\log_2(4096)$	15 $32-\log_2(4096)-\log_2(32)$	64 KB $\frac{[2^{12}\cdot(1+15)\cdot 1]}{1024}$
4-Way Set	128KB	1024 $\frac{(128 \times 2^{10})}{32 \cdot 4}$	10 a=4 $\log_2(1024)$	17 $32-\log_2(1024)-\log_2(32)$	$\frac{[2^{10}\cdot(1+17)\cdot 4]}{1024} = 72$KB

Problem A.1:
Assume a 128-KB cache with 8-word (32-byte) cache lines. The data output is also 32 bits, and the MUX selects one word out of the eight words in a cache line. Using the delay equations given in Table A.1, fill in the columns for the direct-mapped (DM) and 4-way set-associative (SA) caches 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	3400 ps	3000 ps
		Data	3400 ps	3000 ps
Memory array	200×$\log_2$(# of rows) + 200×$\log_2$(# of bits in a row) + 1000	Tag	4217 ps	4249 ps
		Data	5000 ps	5000 ps
Comparator	200×(# of tag bits) + 1000		4000 ps	4400 ps
N-to-1 MUX	500×$\log_2$N + 1000		2500 ps	2500 ps
Buffer driver	2000			2000 ps
Data output driver	500×(associativity) + 1000		1500 ps	3000 ps
Valid output driver	1000		1000	1000

Table A.1: Delay of each Cache Component

We want to compute the access time of the direct-mapped (DM) cache. 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?

DECODER : 200(12)+1000 = 3400 (for all)

S.A. Tag: # of bits in a row - (17+2)·a, (19)·4, 76

MEMORY ARRAY:

$4217.49256\ldots = 200 \cdot \log_2(4096) + 200 \cdot \log_2(15+2) + 1000$ $\quad 200 \cdot \log_2(1024) + 200 \cdot \log_2(76) + 1000 = 4249.58550\ldots$

$5000 = 200 \cdot \log_2(4096) + 200 \cdot \log_2(8 \cdot 32) + 1000$

8 word, 32 bits in a word

Comparator:

200(15)+1000
3000+1000
4000

200(17)+1000
3400+1000
4400

N-to-1 MUX:

$500\log_2(8)+100$

1500+1000 = 2500

Data Output Driver:

500(1)+1000
1500

500(4)+1000
2000+1000
3000

150MHz

$150 \times 10^6 \times 10^{-12}$

$\frac{1}{150 \times 10^{-6}} \rightarrow \frac{1}{150} \times 10^6 = 6666.67$ ps

Critical Path for Cache Read: Tag

(3400) + (4217) + (4000) + (500) + (1000) = 13117 ps

Decoder, Memory, Comparator, AND, VALID OUTPUT BUFFER

Cycle Cycles: $\frac{13117}{6666.67} = 1.96754902$ ps

≈ 2 clock cycles

Problem A.2:

Now George P. Burdell is studying the effect of set-associativity on the cache performance. Since he now knows the access time of each configuration, he wants to know the miss-rate of each one. For the miss-rate analysis, George is considering two small caches: a direct-mapped cache with 8 lines with 16 bytes/line, and a 4-way set-associative cache of the same size. For the set-associative cache, George tries out a least recently used (LRU) policy.

George tests the cache by accessing the following sequence of hexadecimal byte addresses, starting with empty caches. For simplicity, assume that the addresses are only 12 bits. Complete the following tables for the direct-mapped cache and the LRU 4-way set-associative cache showing the progression of cache contents as accesses occur (in the tables, 'inv' = invalid, and the column of a particular cache line contains the {tag,index} contents of that line). You only need to fill in elements in the table when a value changes.

	D-map									
		line in cache								hit?
	Address	L0	L1	L2	L3	L4	L5	L6	L7	
	110	inv	11	inv	inv	inv	inv	inv	inv	no
	136				13					no
	202	20								no
1 010 : 2	1A3			1A						no
0 000 : 0	102	10								no
0 110 : 6	361							36		no
0 000 : 0	204	20								no
0 001 : 1	114		11							yes
1 010 : 2	1A4			1A						yes
0 111 : 7	177								17	no
0 000 : 0	301	30								no
0 000 : 0	206	20								no
0 011 : 3	135				13					yes

D-map	
Total Misses	10
Total Accesses	13

	4-way Address	line in cache hit?								LRU hit?
		Set 0				Set 1				
		way0	way1	way2	way3	way0	way1	way2	way3	
0 0 0 1	110	inv	inv	inv	inv	11	inv	inv	inv	no
0 0 1 1	136					11	13			no
0 0 0 0	202	20								no
1 0 1 0	1A3		1A							no
0 0 0 0	102			10						no
0 1 1 0	361				36					no
0 0 0 0	204									yes
0 0 0 1	114									yes
1 0 1 0	1A4									yes
0 1 1 1	177							17		no
0 0 0 0	301			30						no
0 0 0 0	206									yes
0 0 1 1	135									yes

4-way LRU	
Total Misses	8
Total Accesses	13

M L

20 . .

1A 20 . .

10 1A 20 .

36 10 1A 20

20 36 10 1A

1A 20 36 10

30 1A 20 36