LE14.3

Bookmark this page

LE14.3.1 Set-associative cache

1.0/1.0 point (ungraded)

Consider a **2-way set-associative** cache that has **16 cache lines** in each of the two direct-mapped subcaches and a **block size of 2 words** (i.e., a miss brings in an even-odd word pair from main memory).

1. The Beta produces 32-bit byte addresses, A[31:0]. To ensure the best cache performance for the 2-way set associative cache described above, which address bits should be used for selecting the cache line? For matching with the tag field?
   Address bits used for selecting cache line: A[
   6 ✔
   :
   3 ✔
   ]

   Address bits used for matching with tag field: A[
   31 ✔
   :
   7 ✔
   ]
2. Using the 2-way set-associative cache with a block size of 2 described above, estimate the approximate long-term hit ratio for the following program. Assume that the cache is empty before execution (all the valid bits are 0) and that an LRU replacement strategy is used. Remember the cache is used for both instruction and data (LD) accesses.

```
      . = 0x2000
      CMOVE(0x1000,R0)
loop: LD(R0,0,R1)
      SUBC(R0,4,R0)
      BNE(R0,loop)
      HALT()
```

Approximate long-term hit ratio:
7/8 ✔

Submit

LE14.3.2 Fully-associative cache

1.0/1.0 point (ungraded)

A Beta processor (generating 32-bit byte addresses) is connected to a fully associative cache having four cache lines each containing one 32-bit data word as well as dirty and valid bits. The cache uses Least Recently Used (LRU) replacement. Access times are 5 ns on a hit and a total of 50ns on a miss (including the 5ns cache access time).

1. What hit ratio is necessary to yield an average access time of 14ns?
   Hit ratio for 14ns t_{avg}:
   0.8 ✔
2. The Beta produces 32-bit byte addresses, **A[31:0]**. Which of these bits are compared with tags stored in the cache?
   Bits compared with stored tags: A[
   31 ✔
   :

Calculator

2

]

3. How many such comparisons are performed simultaneously for each memory read?
number of simultaneous comparisons with stored tags:

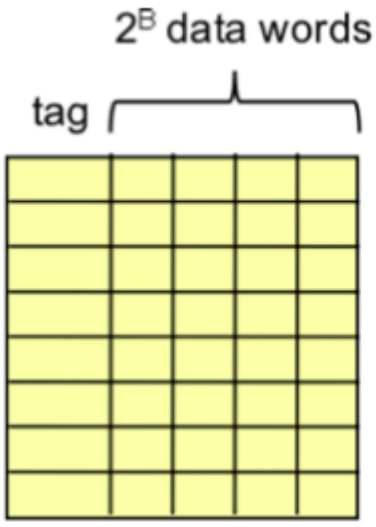
4

Submit

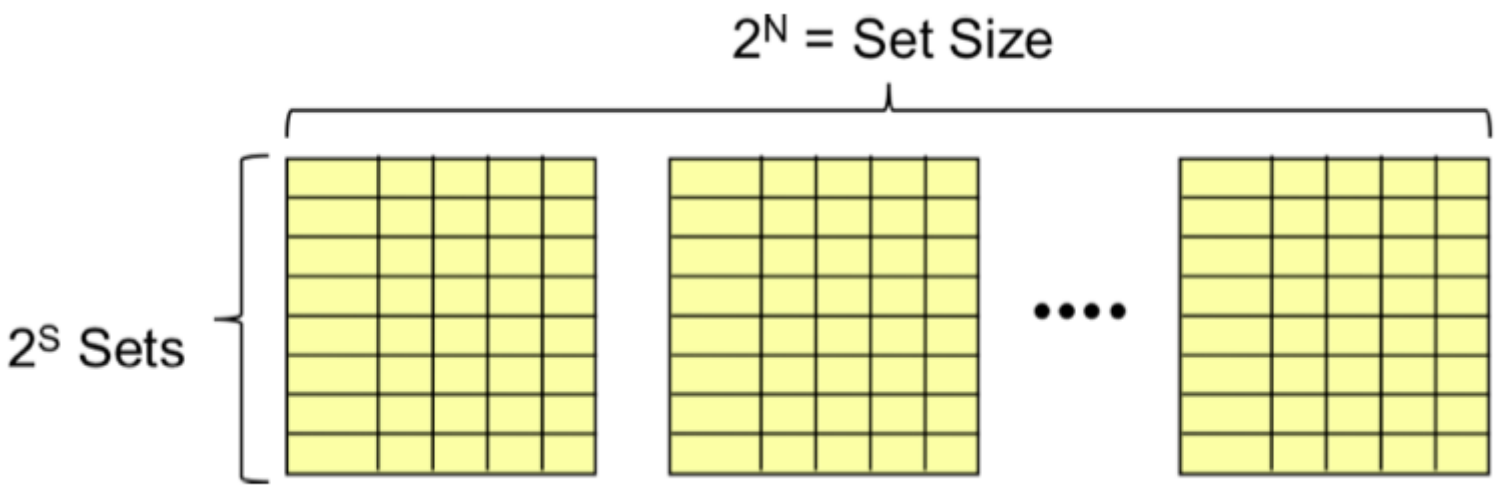
LE14.3.3 Set-associative cache parameters

1.0/1.0 point (ungraded)

Consider the universe of caches constructed from one or more static RAM lookup tables organized as shown on the right, i.e. as table allowing access to a single entry (line) at a time, each line having a tag portion as well as 2^B data entries.



Combining multiple of these devices along with necessary logic, comparators, and replacement strategy, we can build a parameterized family of set-associative caches. Each cache in this family implements 2^S sets, where each set comprises 2^N lines and each line has a tag as well as 2^B consecutive words of data from main memory as depicted in the “big picture” view below:



For each of the following questions, assume a machine that uses **A** bit addresses (to keep it simple, we avoid the byte addressing of the Beta; thus consecutive addresses differ by one). You may answer each question by a number, or a formula involving the parameters A, B, N, and S.

1. What is the total number of data words that can be held in the cache?
Formula for total cache size, in words:

2^(S+N+B)

2^{S+N+B}

2. What constraint on the above parameters characterizes a **direct-mapped** cache?

- $A = 0$
- $B = 0$
- $N = 0$
- $S = 0$

3. What constraint on the above parameters characterizes a **fully-associative** cache?

- $A = 0$

Calculator

- ○ $B = 0$
- ○ $N = 0$
- ◉ $S = 0$ ✔

4. What is the minimum number of bits required in the tag portion of each cache line?
Formula for size of each tag:
A - S - B ✔
$A - S - B$

Submit

LE14.3.4 Cache comparisons

1.0/1.0 point (ungraded)
Three otherwise identical Beta systems have slightly different cache configurations. Recall that the Beta supplies 32-bit byte addresses when accessing memory. Each cache has a total of 8 lines caching a single 32-bit data word; a single cache is used when responding to both instruction and data fetches. However, the caches differ in their associativity as follows:

- **Cache C1:** 8-line direct mapped.
- **Cache C2:** 2-way set associative (4 sets of 2 lines), LRU replacement.
- **Cache C3:** 8-line fully associative, LRU replacement.

1. How many bits are there in the TAG field of each line in cache C1? Which address bits from the Beta are used by the cache's comparator to determine if there is a cache hit?
Number of tag bits in C1 cache line:
27 ✔

Beta address bits used in hit logic: A[
31 ✔
:
5 ✔
]

2. After the Beta with the C2 cache runs for a while the tag and data fields of the cache are as shown in the table below. Assume that any unspecified bits are 0 and that all cache entries are valid. For each of the given Beta read requests to memory, indicate whether they would hit or miss in the cache and the data returned on a hit (enter CAN'T TELL for the data returned if there is a cache miss).

Line #	Tag	Data	Tag	Data
3	0×40	0×739F0083	0×00	0×73FF0121
2	0×00	0×73FF012C	0×40	0×619F0044
1	0×00	0×73FF029E	0×40	0×515F003C
0	0×41	0×627F0060	0×00	0×73FF02C3

Hit for address 0×00C?
- ◉ HIT
- ○ MISS

✔

Calculator

Data at location 0×00C or "CAN'T TELL" for "MISS": 0x

Data at location 0×00C or "CAN'T TELL" for "MISS": 0x

73FF0121 ✔

Hit for address: 0×400?

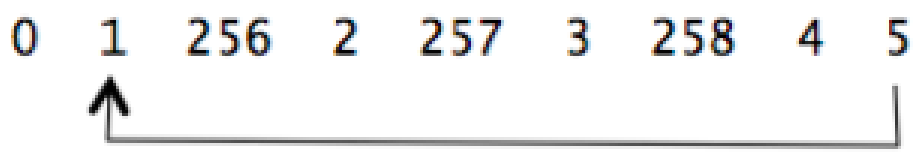
- ○ HIT
- ◉ MISS ✔

Data at location 0×400 or "CAN'T TELL" for "MISS": 0x

CAN'T TELL ✔

3. Consider the short benchmark program given below. The reference string of word addresses generated as the benchmark runs is shown below the benchmark with an arrow indicating how the access pattern repeats. Please indicate the hit ratio delivered by the each of the caches after the benchmark has run for a while, e.g., during the 10th iteration of the loop.

```
      .=0
      CMOVE(100, R1)
LOOP: LD(R31, 1024, R0)
      LD(R31, 1024+4, R0)
      LD(R31, 1024+8, R0)
      SUBC(R1, 1, R1)
      BNE(R1, LOOP)
```



Hit Ratio? for C1:

4/8 ✔

Hit Ratio? for C2:

5/8 ✔

Hit Ratio? for C3:

8/8 ✔

Submit

Discussion

Topic: 14. Caches and the Memory Hierarchy / LE14.3

Hide Discussion

Add a Post

Show all posts ⌄ by recent activity ⌄

Last Exercise Cache C1

Calculator

Community TA

not clear why hit ratio in c2 is 5/8 and not 6/8?	Hi, In cache C2, the two-way set associative cache, the data fetches map to lines 0-2, while the instruction fetches map to lines 1,2,...	5
LE14.3.4.C Not clear why hit ratio for C1 is 4/8	Wouldn't the conflict of the cache causes the next instruction/data to MISS? If even only 256, 257 & 258 is cached, it's 3/8, not 4/8.	2
Not clear - LE14.3.1 Part B	Hi , How are we calculating the hit ratio here and how are we mapping the instructions and data here , since there are two ways is da...	8
Average Memory Access Time ?		4
Not clear why 256 maps to 0?	The benchmark has a loop containing 5 instructions and 3 data fetches. In cache C1, the direct mapped cache, the fetch of word **2...	3

Previous Next