Direct-mapped cache

CPU requests one single word at a time, from MM.

Cache intercepts this request and checks if that word is in cache. If the word is in cache (a cache hit), it returns the word to the cpu. Otherwise (cache miss), a block of words is grabbed from memory (the requested word plus words nearby) and put into cache.
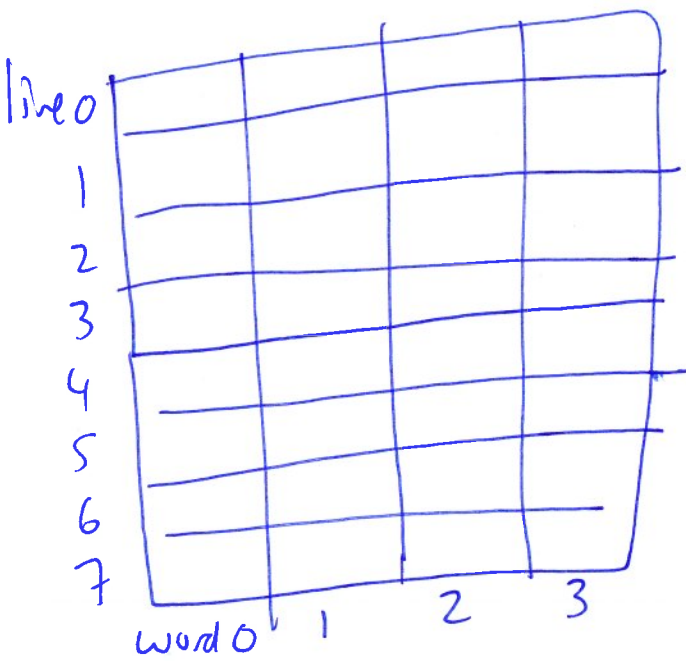
One block of memory is the same size as one line of cache
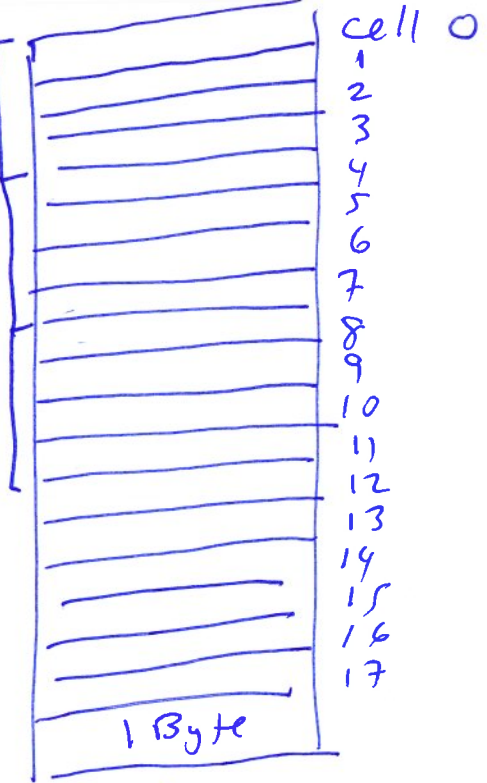
□ □ □

each line of cache holds 4 words

Memory: block 0

cache

block 1

1 Byte/word

line 0
1
2
3
4
5
6
7

∴ each block of MM holds 4 words too

2

cell 0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17

$\log_2 512 = 9$

1 Byte

word 0   1   2   3

8 lines of cache

block 49

1 Byte   195
1 Byte   196
1 Byte   197 ←
         198
         199
         200
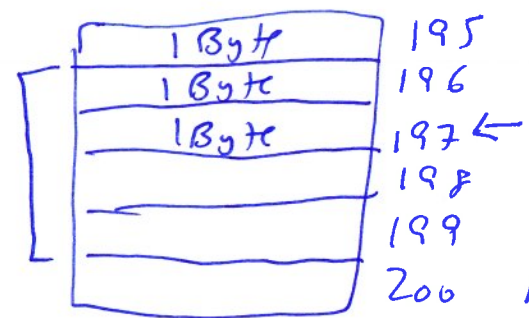
cell 197 is in block 49

$197 \div 4 = 49 R 1$

block ↑   word ↑

Cache vs memory

works with words also with blocks

⤷ {smaller
closer to cpu
faster
more expensive
works with lines

1 Byte

1 Byte   508
         509
         510

cell 511
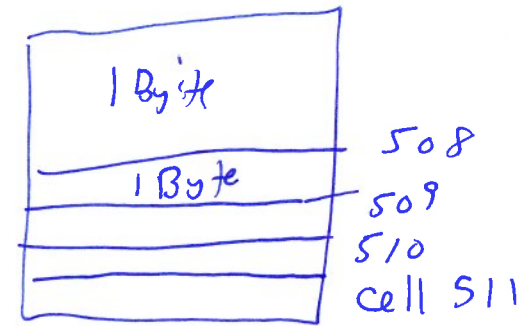
Cache

	Cell 0 / Cell 32	Cell 1 / Cell 33	cell 2	cell 3
line 0	cell 4 / CELL 197	cell 5	6	7
1	8	9	10	11
2	12	13	14	15
3	16	17	18	19
4	cell 20	21	22	23
5	cell 436 / 24	25	26	27
6	28	29	30	cell 31 / 63
7				

word 0   1   2   3

Cell 197 is in block 49   R 1   word 1

block 49 gets directly

mapped into

cache line 49 % 8 = 1   or 436 % 32
     49   = 20

line 1

line 0 holds one block at a time!
   block 0 could be there
   block 8 could be there
        16
         :
        48

where in cache
does cell #
436 go?   436/4 =
               109 R 0
line 5   ⎵
             block   word
word 0   #   #

            109 % 8 = 5

Spatial locality is being put to use in blocks / lines.

eg. when cell 197 is requested by CPU, cells 196, 198, 199 are also brought.

Very likely they'll be requested soon.

eg request for cell 436 brings in cells 436, 437, 438, 439

eg request for cell 15 also brings in cells 12, 13, 14, 15

Bigger line sizes ⇒ better use of spatial locality.

"Direct mapped" means each block of MM is directly mapped to one line of cache. There is no choice. eg. MM block 0 is always directly mapped to cache line 0.

Always ask yourself:

★ How much bigger is MM than cache?

MM: 2^9 words

cache: 2^5 words

∴ MM is $2^9 \frac{w}{m} \div 2^5 \frac{w}{c} = \boxed{2^4 \frac{c}{m} \text{ times bigger}}$

Very useful

dfn: <u>logarithm</u>: exponent

$\log_2 512 = \log_2 2^9 = 9$

$\log_{10} 10000 = \log_{10} 10^4 = 4$

Same cache: 8 lines

4 words/line

Imagine a huge memory!

where in cache does cell 3 200 000 go?

↖ word 0, line 0

cell 3200998 go?

$998 \% 32 = 6$

| line 1 |
| word 2 |

(~~the~~ position

6)