Cache Organization III

March 30, 2016 . Ganesh Kumar

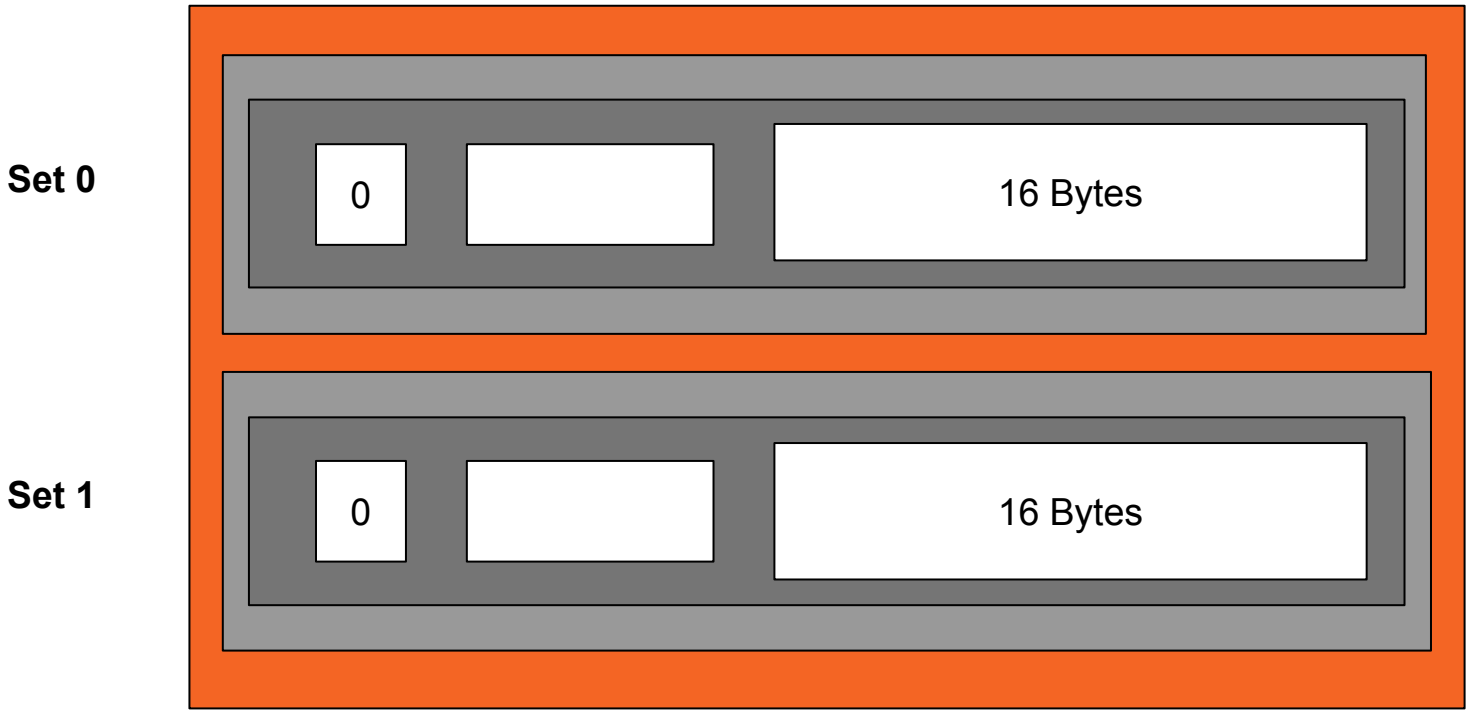
Code

```
int dotproduct(int x[8], int y[8]) {
    int sum = 0;
    int i;

    for (i = 0; i < 8; i++)
        sum += x[i] * y[i];

    return sum;
}
```

Good spatial locality?



Assume a 6-bit address space with t=1, s=1 and b=4.

Say x[0] is stored at address 0…. and x[7] at address 28.
And y[0] starts immediately after at address 32 …. and y[7] at 60.

Element	Address	Set index	Element	Address	Set index
x[0]	0	0	y[0]	32	0
x[1]	4	0	y[1]	36	0
x[2]	8	0	y[2]	40	0
x[3]	12	0	y[3]	44	0
x[4]	16	1	y[4]	48	1
x[5]	20	1	y[5]	52	1
x[6]	24	1	y[6]	56	1
x[7]	28	1	y[7]	60	1

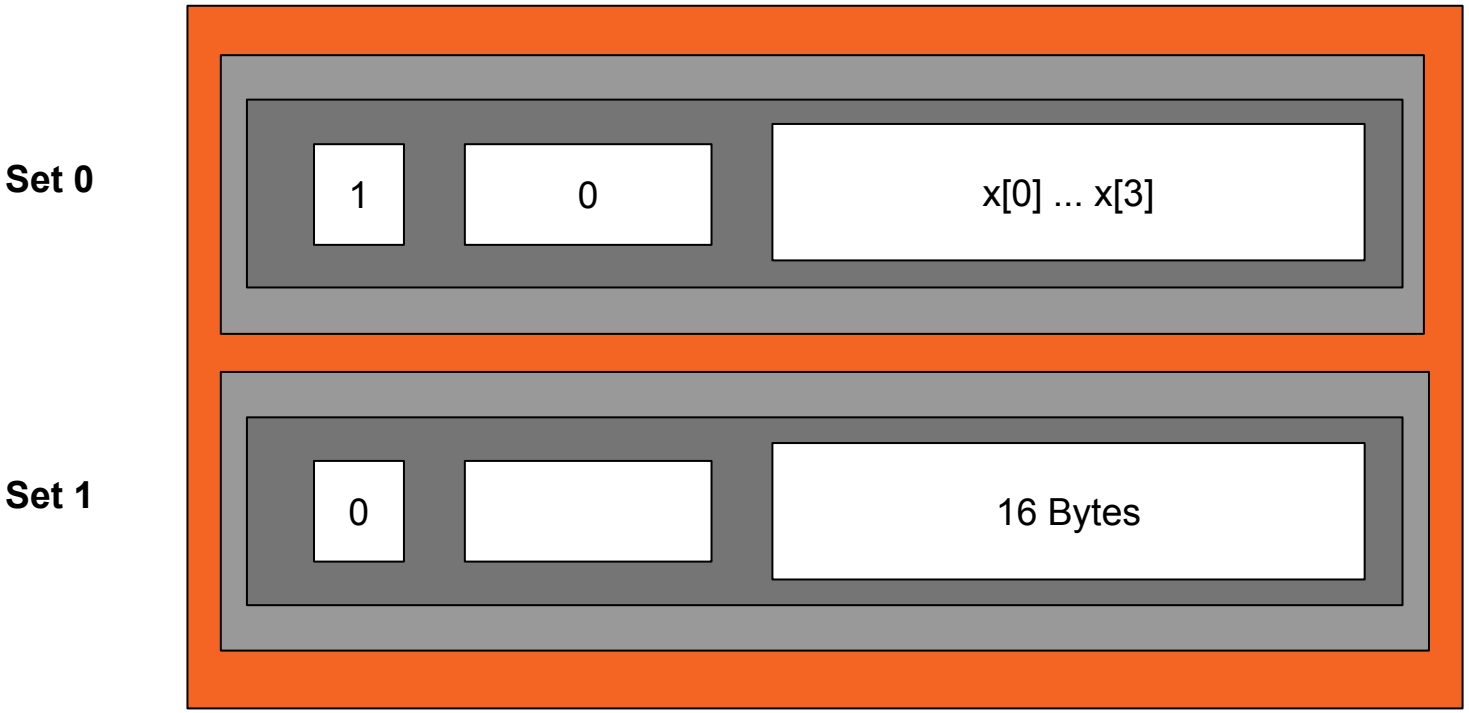
```
x[0] - Address  0 = 0 0 0000 -> Maps to Set 0
x[3] - Address 12 = 0 0 1100 -> Maps to Set 0
The first four element addresses in X map to set 0.

y[0] - Address 32 = 1 0 0000 -> Maps to Set 0
y[3] - Address 44 = 1 0 1100 -> Maps to Set 0
The first four element addresses in Y map to set 0.
```

In 1st iteration,

Search for x[0] in cache.... Valid bit not set... Cache miss!

Read the block containing x[0] onto the cache.

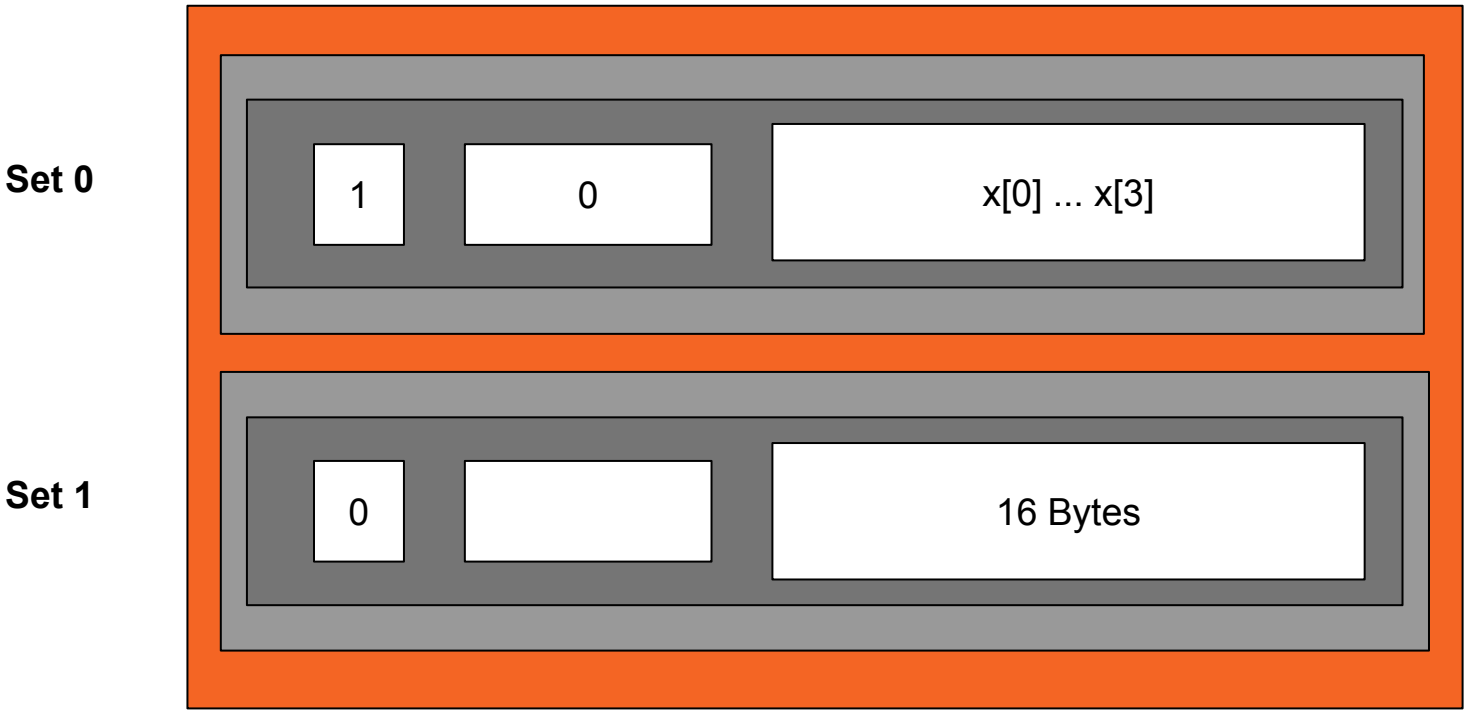


Still in 1st iteration,

Search for y[0] in cache… Address 32 - 1 0 0000… Set 0

Tag bits don’t match… Cache miss!

So, read the block containing y[0] onto the cache.



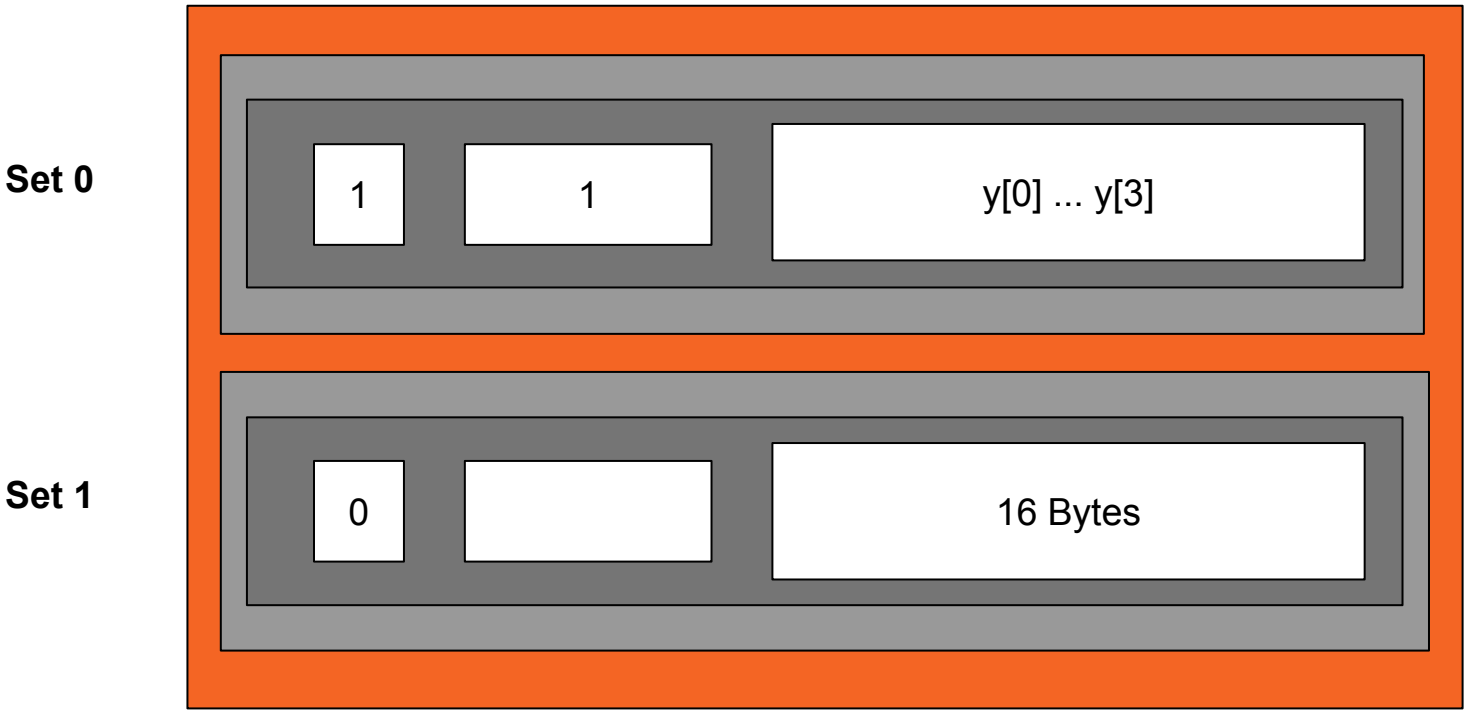
Still in 1st iteration,

Search for y[0] in cache… Address 32 - 1 0 0000

Tag bits don’t match… Cache miss!

So, read the block containing y[0] onto the cache.

Replace existing line in Set 0



Solution to these problems?

Increase E !

Set Associate Cache

$$1 < E < C/B$$

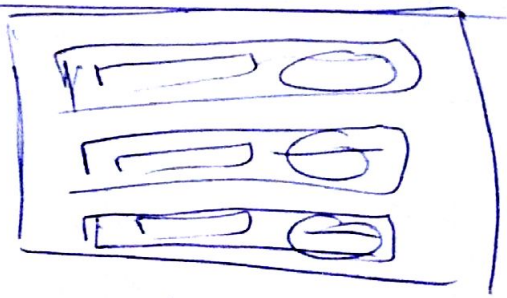
E = 2 → Two-way associative cache

E = 3 → Three-way associative cache.

E is also called associativity.

Set Matching

Line Matching

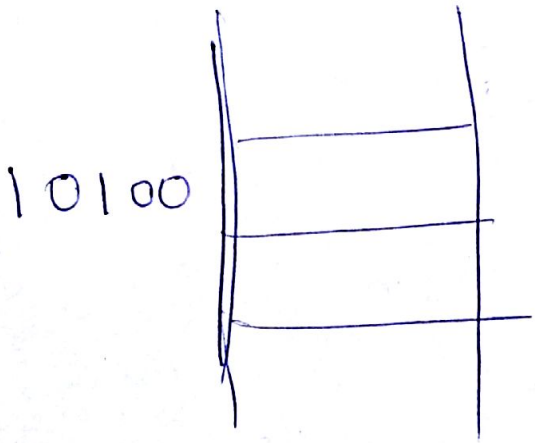


(key, value) → Data Block.

↓

valid + tag bits

(Associative memory)



Conventional memory.

Line Replacement (on miss).

Which line to replace?

① Empty line

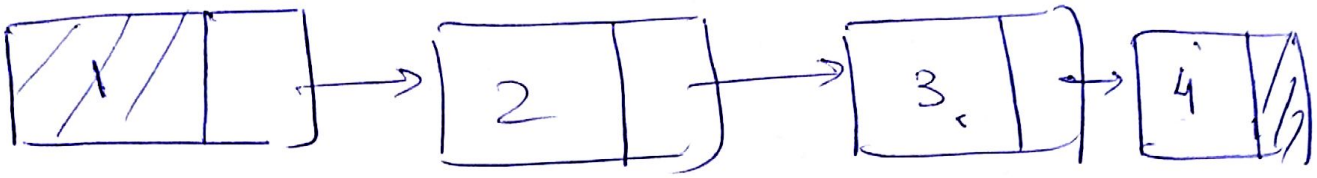
② Policy → Random

→ Least Frequently Used.

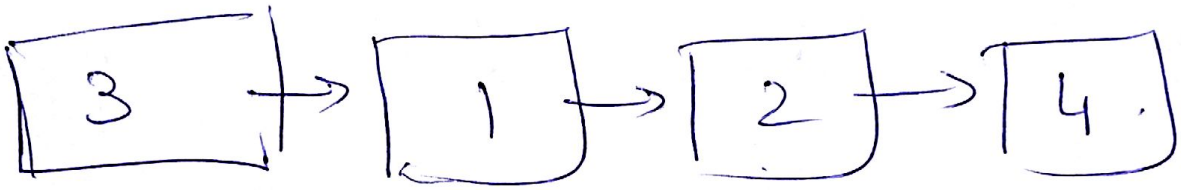
→ Least Recently Used..

Need to keep track of how many times each line is used.

LRU.



Search for a line → 3



Fully Associative Cache

$$E = C / B$$

We know that capacity $C = S \times E \times B$

$$\frac{C}{B} = S \times E$$

$$E = S \times E$$

↳ $s = 1$

→ We just have 1 set!

→ We have no set bits ($s = 0$)

$E > C/B$ Why?

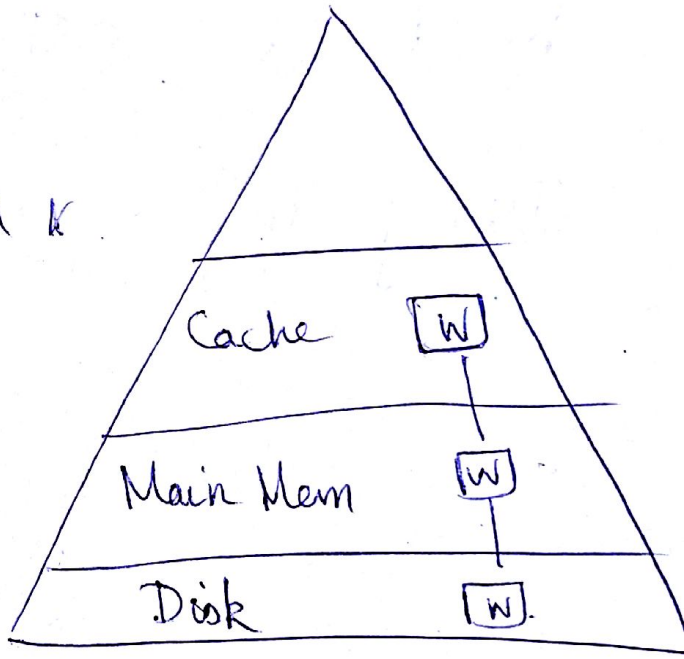
↓

Not possible

So far → reads!

Writes?

Say we write to a word w in the cache (write hit)



when should we update the value of w in MM & the Disk?

1) Write-through

Do it immediately!

Drawback?

(i) Super slow!

(ii) Bus traffic!

(2) Write - back

Write only the updated line is going to be replaced when a ↑

Drawbacks :

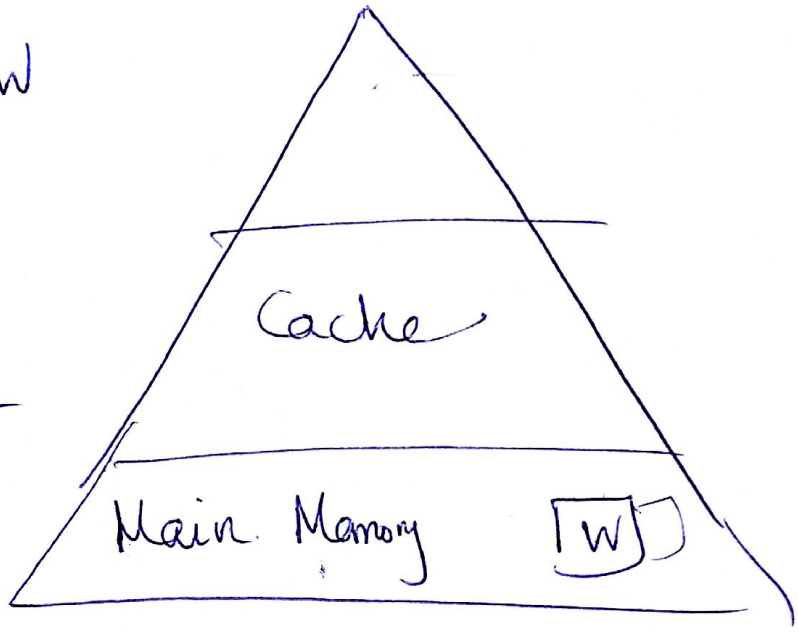
Need an extra bit !

Dirty bit → If updated, set to 1,

If not, set to 0.

Write misses ?

Need to write to w



① Write - allocate.

Load w onto the cache and then update it
(Assumes good spatial locality)

② No - write allocate

Bypass the cache and write directly to the next memory.

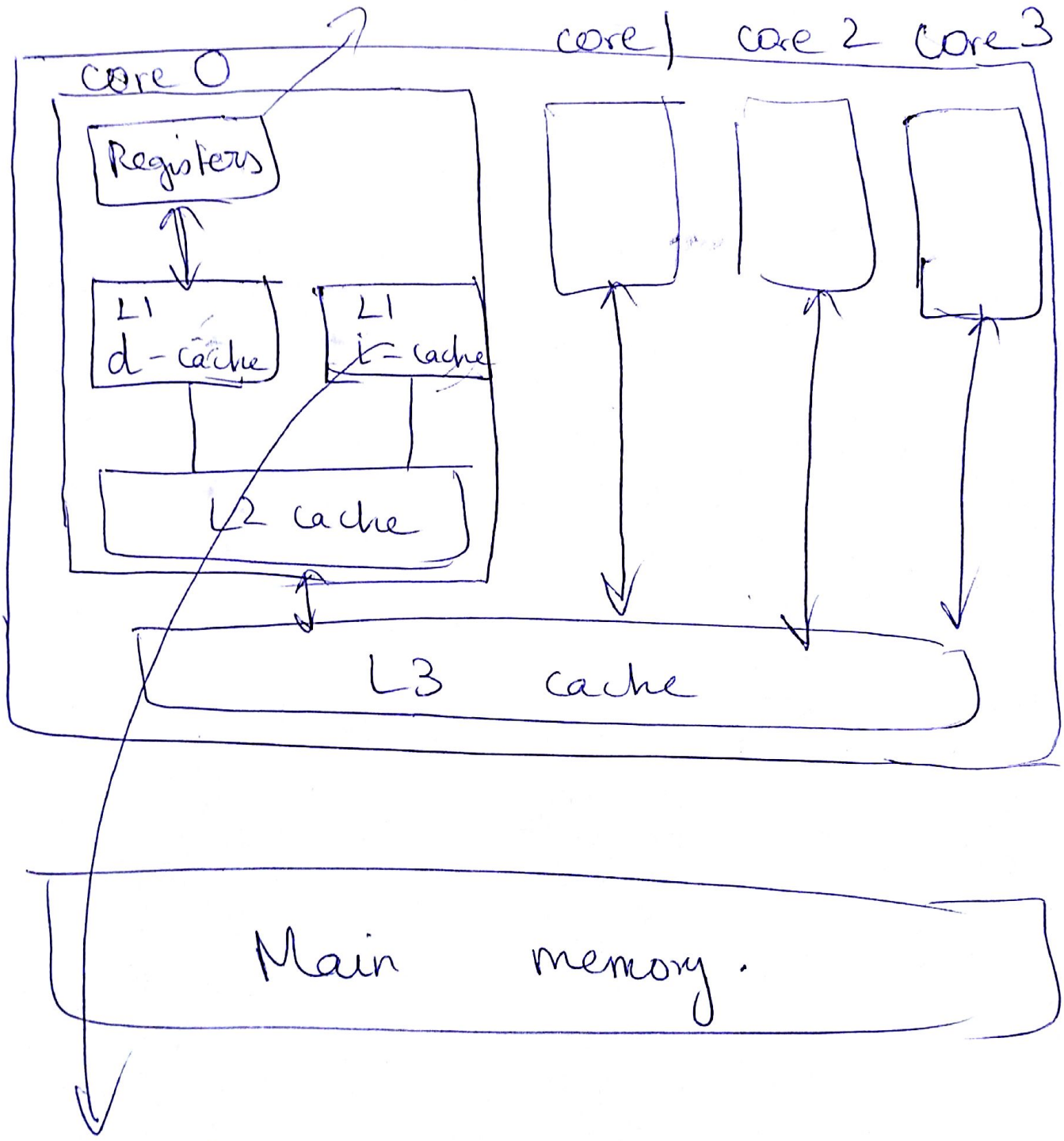
Write through / No - write - allocate.

Write back / Write - allocate

↓

Cache levels at lower memory hierarchy use this

i7 cache diagram



Could be only read only. //.