Set-Associative Cache

In direct-mapped cache, each block of memory has precisely ONE line of cache to which it gets mapped.

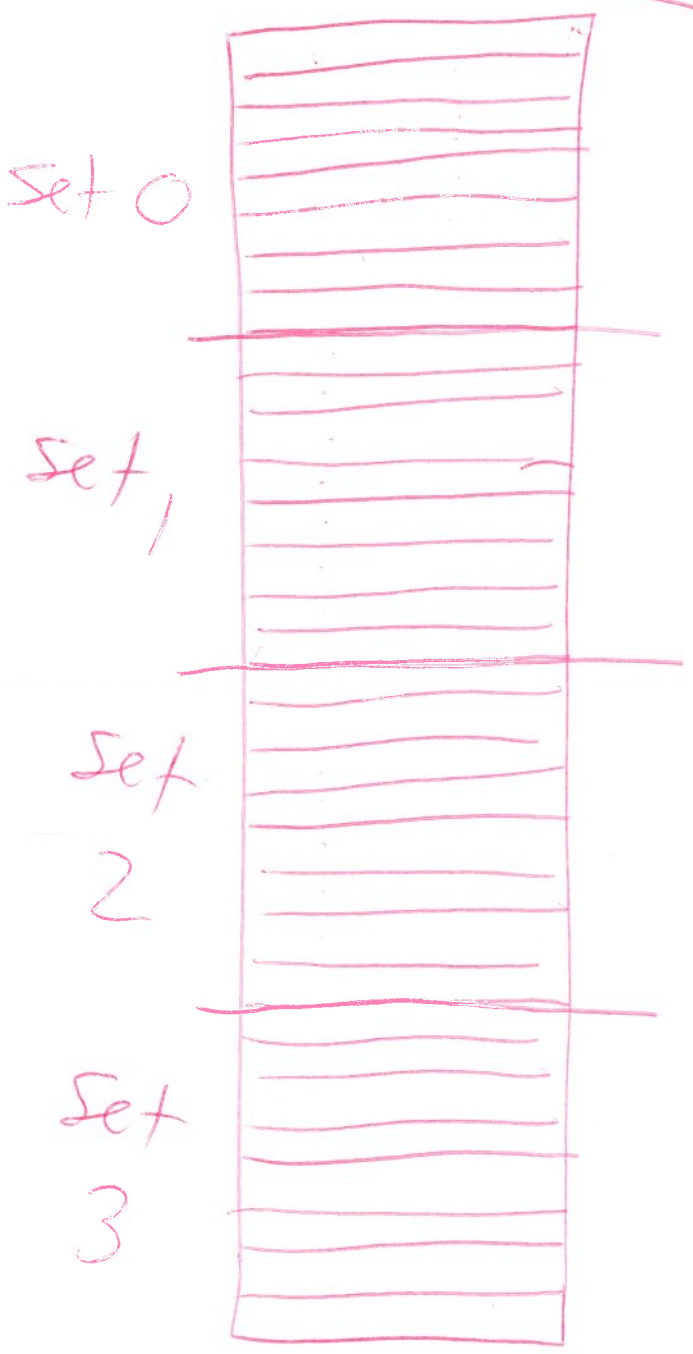
In a 4-way set-associative cache (for example), each memory block has 4 lines ~~to~~ ~~choose~~ from which to choose.

D-M: no choice or flexibility in where to go, or over which block to evict.

4-way SA: choice of 4 places in which to go, choice of 4 lines (to evict one)

☆ A direct-mapped cache is a 1-way S-A cache.

eg. 8-way 5-A cache: with 32 lines:



8 lines per set

Each MM block is mapped to a set, not a particular line. ie 8 choices

---

eg consider MM block 0
it used to get mapped into line 0 (direct mapped). Now it goes anywhere in <u>set</u> 0.

An algorithm determines where it goes; which of set 0's eight lines.

Example cache-replacement algorithm:

LRU

least-recently-used algorithm says to replace the most stale (ie least-recently-used) line in a set.

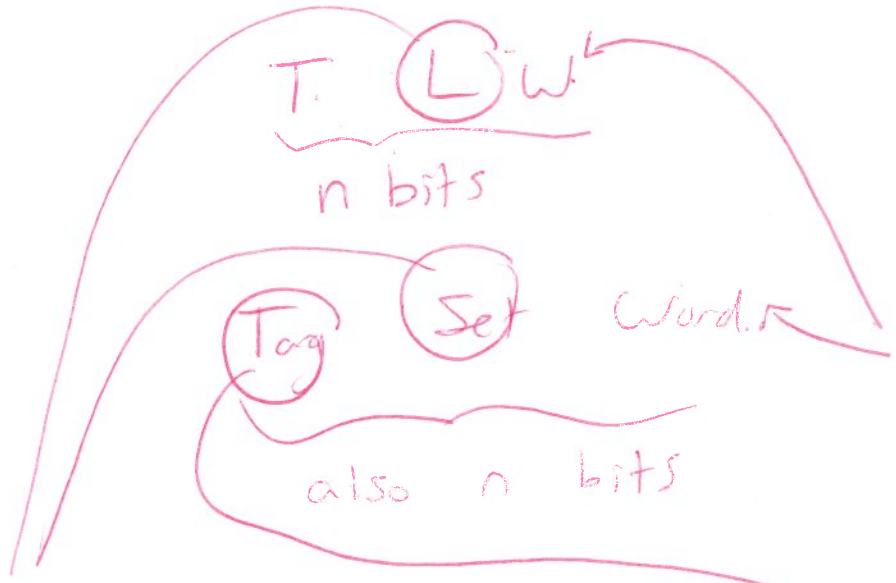
Makes use of the principle of temporal locality.

ie keep the most-recently-used.

Advantage of 8-way SA cache vs D-M cache!

eg We won't replace the MRU words

Virtual address is no longer

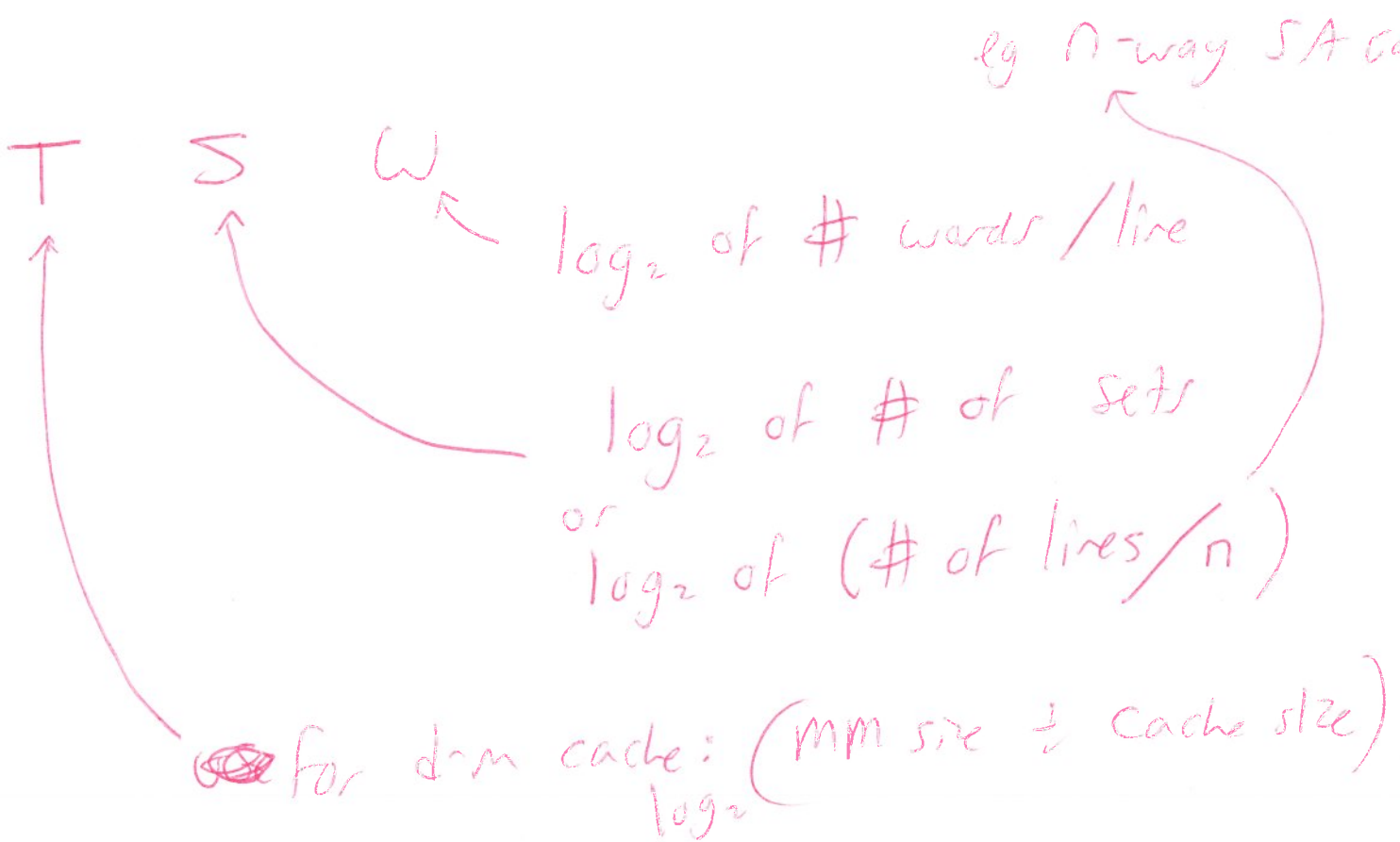


the number of words per line doesn't change

we have eg 16 times more lines than sets

tag is 4 bits longer than DM cache (for an example 16-way SA cache)

eg 16-way SA cache



for n-way SA cache:

$\log_2 (\lceil \text{MM size} \div \text{cache size} \rceil \times n)$

A 4-way S-A cache:

- virtual address comprises TSW not TLW
- Each VM block has 4 places to go
- use algo to evict one of the four lines in the set (eg LRU algo)
- # of bits in the VM address tag goes up +2 bits ($\log_2 4$)

A3: due in teams of 3! new partners due April 13 1159 pm to Jason's red box in Connector bldg.