CPU cache mapping (aliasing)

1. fully associative

- every mem-block can be in every cache-line

```
|______32 bits________|
|_26b_tag_|_6b_offset_|
```

When data are cached in cache-line, corresponding tag is stored for addressing purposes When data are being retrieved, tag of data being retrieved needs to be compared against every cache-line, implying O(n) time complexity, where n is the number of cache-lines

2. direct mapping

- for each mem-block there is exactly 1 cache-line
- example: cache 32kB => 512 64B cache-lines

	32 bits	
17b tag	9b index	6b offset

Cache is an array of cache-lines without associated tags. Index is basically direct pointer into the array. Given address therefore has exactly 1 place where it can be stored, so no cache-strategy is needed

3. set-associative mapping

- every block can be in a given set of 2^i cache-lines

	32 bits	
$(17+i)b$ tag	$(9-i)b$ index	$6b$ offset

$$x^2 + 4 = 15$$

Cache is an array of 2^{9-i} sets of cache-lines, eache set is identified by the index and has cardinality 2^i, so for every retrieval only 2^i tag comparisons need to be made.