Skewed Associative Cache

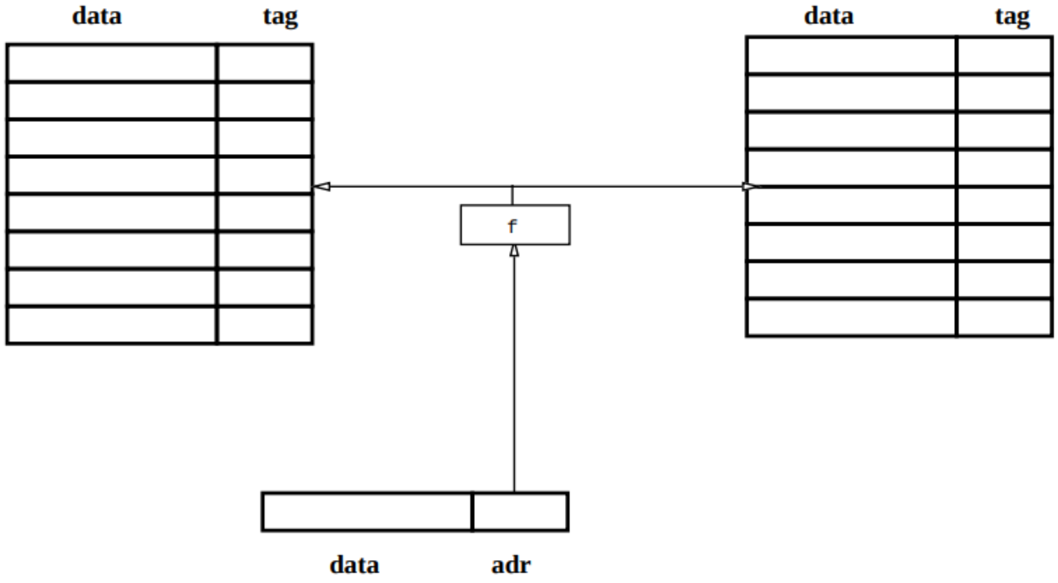
There goes an inherent trade-off between direct mapped cache and N-way set associative cache in terms of hit rate and time required to access a given memory address. Direct mapped cache has a fast access to a given address but at the same time, due to allocation of only a single block to a given address in the whole cache, it has a low hit rate. On the other hand, set associative cache has to search in at most N-blocks while searching for a given address which makes it a little bit slow, but since a given memory address can now be put into N different blocks, it has a high hit rate.

Now here comes yet another type of cache, the Skewed associative cache. It has almost the same hardware implementation as that of a set associative cache but is twice as faster than a set associative cache.

So how is it exactly different from Set Associative Cache ?

A Set Associative Cache and a Skewed Associative cache are both multi bank cache. A multi bank cache is one which has more than one block allocated for a given memory address. If a particular address of main memory can be allocated to N blocks in the cache, then the cache is said to have N banks. That memory address can then go to any of these N banks depending upon the replacement policy used in putting the new memory address.

The way in which Skewed and Set associative cache differ from each other is in the way of allocation of block in these different banks. The set associative cache uses the same function to allocate block in different banks. A case for 2-way set associative cache (having 2 different banks) can be illustrated from the following figure.



A two−way associative cache~:
two distinct cache banks

Here the same function (say f) is used to map a given memory address to a block in different banks. The block chosen finally depends upon the replacement policy used.

On the other hand, a N-way Skewed Associative cache also has N banks, but the block assignment in these banks is done by different functions (say f1 and f2).