A set associative cache requires more energy to search the cache compared to a direct mapped cache. This is because multiple tag ways must be searched, and multiple data ways read out, which is wasteful considering that the data would be found in only one of the ways (or none in the case of a miss).

1 A. CHECK YOUR UNDERSTANDING (1 point possible)

An energy saving alternative frequently implemented in level 2 (L2) caches (which we will introduce in the Multicore module) is to first check the tags in all ways, and if there is a hit, to read only the data way where the data is located. With this approach, only one of the data ways is read on a hit, and none are read on a miss. This reduces the energy usage of the cache compared to the conventional approach of reading all data ways in parallel with all tag ways.

Which of the following is a *disadvantage* of this approach?

- ( ) It has a lower hit rate.
- ( ) It takes longer to determine whether there was a hit or miss.
- (•) It requires additional comparators. ✖
- ( ) It takes longer to retrieve the data on a hit. ✔

EXPLANATION

The hit rate does not change with this approach, and the time to check for a hit does not change.

Determining which way had a hit can be done by reading the AND gate outputs of the hit detection logic for each way; no additional comparators are required.

Since the tags are read first, followed by the data (rather than in parallel with the conventional approach to accessing the cache), it takes longer to retrieve the data on a hit.

Hide Answer *You have used 2 of 2 submissions*

Show Discussion New Post

1 B. CHECK YOUR UNDERSTANDING (1/1 point)

Help

A second energy saving alternative is called *way prediction*. Here, we predict in which way of the cache we expect to find the data. For a four-way associative cache, we have four possible ways in which the block may be found, which requires storing two extra bits—the way prediction bits—with every set of the cache. The bits determine which way of the cache to check first. For instance, if the bits are 00, then way 0 is first checked (both tag and data in parallel). If we miss in the predicted way, we check all of the other ways in parallel. We set the way prediction bits according to the way where we last had a hit when we accessed this set. For example, if the last hit for set 12 was to way 2, then we set the way prediction bits for set 12 to the binary value 10.

Which of the following is an *advantage* of way prediction over the first energy saving approach (the one in part A)?

- (•) It can potentially more quickly retrieve the data on a hit. ✔
- ( ) It has a higher hit rate.
- ( ) It always takes less time to determine whether there was a hit or a miss.
- ( ) It requires fewer tags than the first approach.

EXPLANATION

If the prediction accuracy is high, then the time to retrieve the data can be close to that of a direct mapped cache, and therefore quicker than the first approach. We say potentially more quickly because the opposite is true: if the prediction accuracy is poor, then we will spend extra time checking the cache twice, making it potentially slower than the first approach.

The hit rate would be no different than the first approach.

It may take longer to determine if there was a hit or miss depending on the prediction accuracy.

It requires the same number of tags as the first approach.

Final Check | **Save** | **Hide Answer** *You have used 1 of 2 submissions*

Show Discussion

New Post