

Chat Mode

RAG Mode

Parameters

Max Length

1024

82048

Temperature

0.70

0.001.00

Actions

New Chat

Clear History

Response Time (Non-RAG): 6.77 seconds

CPU Usage (Non-RAG): 6.50%

Memory Usage (Non-RAG): 0.00%

Local Chatbot

A chatbot powered by LlamaIndex and Ollama

Hello, I'm your assistant, how can I help you?

9. In the context of CPU cache design, which of the following cache mapping policies would result in the highest hit rate but potentially the most expensive implementation? a) Direct mapping b) Set associative mapping c) Fully associative mapping d) Sequential mapping

Hello!

In the context of CPU cache design, the correct answer is:

c) Fully associative mapping

Here's why:

• **Direct Mapping (a):** Each cache line is assigned to a unique index in the cache. This means that if two lines need to be stored at the same index, one will have to be replaced, resulting in a lower hit rate.

• **Set Associative Mapping (b):** A set of cache lines can store multiple blocks from the same memory location. While this is more efficient than direct mapping, it still has limited associativity and may not achieve the highest hit rate.

• **Fully Associative Mapping (c):** This policy allows any cache line to store any block from main memory. This means that the most frequently accessed lines are more likely to be stored in the cache, resulting in a higher hit rate.

However, as you mentioned, this policy can be the most expensive implementation because it requires a large number of tags and comparators to keep track of which cache line stores which block.

So, while Fully Associative Mapping has the potential for the highest hit rate, its complexity and cost make it less practical in many CPU architectures.

Would you like me to elaborate on any of these points or explore more aspects of CPU cache design?

Ask a question:

localhost:8501

1/1