DAY 20 - 111 DAYS VERIFICATION CHALLENGE

Topic: Cache mapping

Skill: Cache memories

DAY 20 CHALLENGE:

1. Explain various cache mapping techniques with an example.

1. **Direct Mapping**
2. **Associative Mapping**
3. **Set-Associative Mapping**

Let's explore each technique with examples:

1. Direct Mapping

Description: In direct mapping, each block of main memory maps to only one possible cache line. The cache is divided into lines, and each line can hold one block of memory.

Example:

- Assume a cache with 8 lines (0-7).

For example:

- Block 0 maps to line 0 (0 % 8 = 0).
- Block 8 maps to line 0 (8 % 8 = 0).
- Block 9 maps to line 1 (9 % 8 = 1).

So, blocks 0, 8, 16, 24, etc., all map to cache line 0.

2. Associative Mapping

Description: In fully associative mapping, any block of main memory can be placed in any cache line. There are no fixed positions, and any free line can be used to store the block.

Example:

- Assume a cache with 8 lines.
- Any memory block can go into any of these 8 lines.

If block 0 is placed in line 2, block 1 could be placed in line 5, block 2 in line 0, and so on. The decision of placement depends on the replacement policy (e.g., LRU, FIFO).

3. Set-Associative Mapping

Description: Set-associative mapping is a combination of direct and fully associative mapping. The cache is divided into several sets, and each set contains multiple lines. A block of memory maps to a specific set but can occupy any line within that set.

Example:

- Assume a cache with 8 lines and 2 lines per set (4 sets).
- Memory blocks are mapped to sets using the formula: Set number = (Block number) % (Number of sets).

For example:

- Block 0 maps to set 0 (0 % 4 = 0) and can be placed in any of the lines in set 0.
- Block 4 maps to set 0 (4 % 4 = 0).
- Block 1 maps to set 1 (1 % 4 = 1).

If set 0 contains lines 0 and 1:

- Block 0 can go into either line 0 or line 1.
- Block 4 can go into either line 0 or line 1.

2. What are pros & cons of every mapping technique?

1. Direct Mapping

Pros:

- Simple to implement.
- Fast cache lookup.
- Requires less hardware complexity.

Cons:

- High conflict misses (multiple data items map to the same cache line).

- Less flexible due to fixed location mapping.

2. Fully Associative Mapping

Pros:

- Flexible: any memory block can be placed in any cache line.
- Lower conflict misses compared to direct mapping.

Cons:

- More complex and expensive hardware (requires associative memory).
- Slower cache lookup due to searching all lines.

3. Set-Associative Mapping

Pros:

- Balance between direct and fully associative mapping.
- Reduced conflict misses compared to direct mapping.
- More flexible placement within a set.

Cons:

- More complex and expensive than direct mapping.
- Slightly slower lookup compared to direct mapping.

3. Which mapping technique is most optimized?

The most optimized mapping technique is Set-Associative because it consist of both method advantage.

4. I have a RAM of 1 MB & cache memory of 4 KB in a word addressable system, where 1 word is 4 bytes. Consider every frame consists of 4 words, which means every frame has 4*4 bytes = 16 bytes. How are we going to map the frames from RAM in cache memory for each mapping technique?

(In case of set-associative consider 4-way)

Tips to Solve this numerical:

Fully Associative - any frame can be placed in any cache line, so no need to solve. Any block of RAM can be moved to any cache line.

Direct mapping - find out which frame nos. in RAM will be mapped to which cache lines.

Set Associative - first calculate how many bits are for tag, set & frame offset. Now, calculate which frames in RAM can be mapped to which set (it can go to any of the 4 ways in the set)