1. Explain various cache mapping techniques with an example.
   Direct-Mapped Cache:

   - Definition: Each block of main memory maps to exactly one cache line. This means each memory address is associated with a specific cache line using a modulo operation.
   - Example:
     - Memory Size: 16KB
     - Cache Size: 4KB
     - Block Size: 1KB
     - Number of Cache Lines: 4KB / 1KB = 4 lines
     - Mapping: The address is divided into three parts: Tag, Index, and Offset. The Index part determines the cache line, and the Tag part is used to check if the correct data is present in that line.
     - Address Mapping: If memory address 0x1234 maps to cache line 0, data from this address will always go to cache line 0.

   Fully Associative Cache:

   - Definition: Any block of main memory can be placed in any cache line. The cache does not use a fixed mapping but instead checks all cache lines to find a match.
   - Example:
     - Memory Size: 16KB
     - Cache Size: 4KB
     - Block Size: 1KB
     - Number of Cache Lines: 4KB / 1KB = 4 lines
     - Mapping: The entire cache is searched for the tag of the requested address. This is more flexible but requires more complex hardware to search all lines simultaneously.
     - Address Mapping: The address 0x1234 can be stored in any of the 4 cache lines. The cache needs to search all lines to check if 0x1234 is present.

   Set-Associative Cache:
   - Definition: Combines features of both direct-mapped and fully associative caches. The cache is divided into sets, and each block of memory maps to a specific set. Within each set, the block can be placed in any cache line.
   - Example:
     i. Memory Size: 16KB
     ii. Cache Size: 4KB
     iii. Block Size: 1KB
     iv. Number of Sets: 2 (e.g., 2-way set associative)
     v. Number of Cache Lines per Set: 4 lines / 2 sets = 2 lines per set
     vi. Mapping: Address is divided into Tag, Set Index, and Offset. The Set Index determines which set to check, and the Tag is compared against the tags in the selected set.
     vii. Address Mapping: If address 0x1234 maps to set 1, the cache will check the two lines in set 1 to find a match.

2. What are pros and cons of every mapping technique?
   Direct-Mapped Cache:

- Pros:
    - Simplicity: The mapping process is straightforward, making the implementation simpler and faster. Each memory address maps to exactly one cache line, which simplifies the hardware design.
    - Speed: Access time is fast since the cache line to check is determined directly from the address using a simple index.
- Cons:
    - Conflict Misses: Multiple memory addresses mapping to the same cache line can lead to high conflict misses. This happens when frequently accessed addresses overwrite each other in the same cache line, reducing cache efficiency.
    - Limited Flexibility: With fixed mapping, there's no flexibility in choosing where to place data, which can lead to inefficient cache utilization.

Fully Associative Cache:

- Pros:
    - Flexibility: Any block of memory can be placed in any cache line, which minimizes conflict misses and allows for more efficient use of cache space.
    - Reduced Conflict Misses: Since any block can go into any line, there are fewer chances of cache misses due to address conflicts compared to direct-mapped caches.
- Cons:
    - Complexity: The hardware needs to compare the tag of the requested block with all tags in the cache simultaneously, making the implementation more complex and costly.
    - Slower Access Time: The need to search all cache lines for a match can slow down access time compared to direct-mapped caches.

Set-Associative Cache:

- Pros:
    - Balance: Offers a balance between direct-mapped and fully associative caches. It reduces conflict misses more effectively than direct-mapped caches while being simpler and faster than fully associative caches.
    - Reduced Conflict Misses: Since each memory block maps to a specific set and can be placed in any line within that set, it helps in reducing conflict misses compared to direct-mapped caches.
- Cons:
    - Increased Complexity: More complex than direct-mapped caches due to the need for set management and multiple lines per set.
    - Slower Than Direct-Mapped: While faster than fully associative caches, set-associative caches are slower than direct-mapped caches due to the need to check multiple lines within a set.

3. Which mapping technique is most optimized?
   Set-Associative Cache is generally considered the most optimized technique because it balances between the complexity and flexibility of fully associative caches and the simplicity and speed of direct-mapped caches. It provides effective conflict miss reduction without the high cost and complexity of fully associative caches.

4. I have a RAM of 1MB and 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 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)
   RAM Size: 1 MB = 2^{20} bytes
   Cache Size: 4 KB = 2^{12} bytes
   Word Size: 4 bytes
   Frame Size: 4 words = 16 bytes
   Cache Organization:
   Number of Cache Frames: 4KB/16 bytes = 2^8 = 256 frames
   Direct-Mapped Cache:

   - Mapping Process:
     - Each memory frame maps to exactly one cache line based on the frame's address.
     - The cache has 256 lines (frames), so the cache index is determined by the modulo of the number of cache lines.
     - Address Calculation: Given a memory frame address, calculate the cache line by taking the modulus with 256.

   Fully Associative Cache:

   - Mapping Process:
     - Any frame can be placed in any cache line. The cache searches all lines to find the frame.
     - Address Calculation: The frame address is compared against all tags in the cache lines.

   Set-Associative Cache (4-way):

   - Mapping Process:
     - The cache is divided into sets. Each set has 4 cache lines (frames).
     - The frame is mapped to a specific set based on the set index, and can be placed in any line within that set.
     - Number of Sets: 256 lines/4 lines per set = 64 sets.
     - Address Calculation: The set index is determined by taking the modulus of the number of sets.