Computer Organization and Embedded System

Computer Arithmetic and Memory System (ACtE0402)

Ishwar Kumar Singh
Computer Engineer
Government of Nepal

Mapping Function

- An algorithm is needed for mapping main memory blocks into cache lines.
- A means is needed for determining which **main memory block** currently occupies a **cache line**.
- The choice of the mapping function dictates how the cache is organized.
- There are three techniques which can be used for mapping
  - Direct Mapping
  - Associative Mapping
  - Set-Associative Mapping

Direct Mapping

- Direct Mapping is the simplest technique which maps **each block of main memory** into **only one possible cache line**.
- **Cache Structure:** In direct mapping, the cache is divided into multiple lines and each line can store one block of data from main memory.
- **Fixed Mapping**: Each block in main memory can map to only one specific cache line.
- **Address Breakdown:** The memory address requested by the CPU is divided into three parts: Block Offset, Index and Tag.

Direct Mapping

- **Block Offset:** Identifies the specific data within the block.
- **Index:** Identifies the specific cache line where the block should be stored.
- **Tag:** Used to identify the actual block of data stored in the indexed cache line.

Cache Access:

- When the CPU accesses memory, the index part of the address points to the specific cache line.
- The tag part of the address is compared with the tag stored in that cache line to check if the block stored there matches the requested block.

Direct Mapping

- **Cache Hit**: If the tag in the cache matches the tag of the requested memory block, the data is read from the cache.
- **Cache Miss:** If the tags do not match, the data is not in the cache. The corresponding block from main memory is loaded into the cache line, replacing any existing data.

Replacement Policy:

- Since direct mapping restricts each memory block to one cache line, no complex replacement policy is needed.
- If a new block needs to be loaded into the cache, it simply replaces the block in its assigned cache line.

Direct Mapping

- **Advantages:** Simple and Inexpensive to Implement.
- **Disadvantage:**
  - Since there is a fixed cache location for any given block.
  - Thus, if a program happens to reference words repeatedly from two different blocks that map into the same line,
  - then the blocks will be continually swapped in the cache, and the hit ratio will be low (a phenomenon known as ***thrashing***).

Associative Mapping

- It overcomes the disadvantage of direct mapping by **permitting each main memory block** to be loaded into **any line of the cache**.
- **Cache Structure**: There are no fixed locations in the cache for storing a particular block from main memory.
- **Tag and Data:**
  - Each cache line in associative mapping consists of a tag and data.
  - The tag is used to identify which block of memory is currently stored in the cache line.

Associative Mapping

- **Flexibility**:
  - If a cache line is free, the block can be placed there.
  - If all lines are occupied, a replacement policy (like Least Recently Used, LRU) determines which block to evict.

Cache Access:

- When the CPU wants to read or write data, it first checks the cache.
- It compares the tags of all the cache lines with the requested memory block's tag.

Associative Mapping

- **Advantages:**
  - Flexibility to replace block when a new block is read into the cache.
  - Replacement Algorithm can be used to maximize the hit ratio.
- **Disadvantage:**
  - Complex Circuitry is required to examine the tags of all cache lines in parallel.
  - Because the cache must check all tags simultaneously to find a match, it can be slower than direct mapping, especially if the cache is large.

Set-Associative Mapping

- Set-associative mapping is a hybrid cache memory mapping technique that combines aspects of both direct mapping and fully associative mapping.
- It provides a balance between the simplicity of direct mapping and the flexibility of fully associative mapping, aiming to reduce cache conflicts while keeping the hardware manageable.

Cache Structure:

- The cache is divided into multiple sets, and each set contains several cache lines.
- A block of memory can be mapped to any line within a specific set, not just a single line as in direct mapping.

Set-Associative Mapping

- **Fixed Set, Flexible Line**:
  - Each block from main memory is assigned to a particular set (like in direct mapping),
  - But within that set, the block can be placed in any line (similar to associative mapping).
- **Address Breakdown:** The memory address is divided into three parts:
  - **Block Offset:** Identifies the specific data within the block.
  - **Set Index:** Identifies the specific set in the cache to which the memory block is mapped.
  - **Tag:** Used to identify which block is stored within a line of the set.

Set-Associative Mapping

- When the CPU generates a memory request,
  - the index values of the address is used to access the cache set
  - the tag field of the CPU address is then compared with tags inside the cache set
  - the comparison logic is defined by an associative search of the tags in the set.
- **Replacement Policy:**
  - If the selected set is full and a new block needs to be placed in the set, a replacement policy (e.g., Least Recently Used, LRU) is used to determine which block within the set should be evicted to make room for the new block

Thank You.