

Cache Coherence

In a multiprocessor system, data inconsistency may occur among adjacent levels or within the same level of the memory hierarchy.

In a shared memory multiprocessor with a separate cache memory for each processor, it is possible to have many copies of any one instruction operand: one copy in the main memory and one in each cache memory. When one copy of an operand is changed, the other copies of the operand must be changed also.

Example:- : Cache and the main memory may have inconsistent copies of the same object.

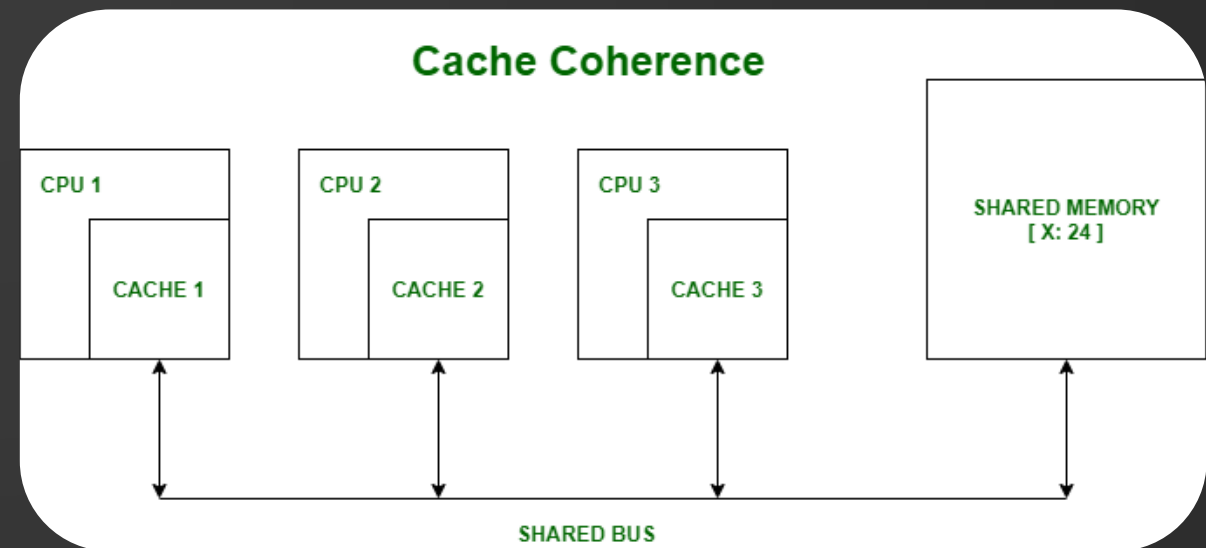
Suppose there are three processors, each having cache. Suppose the following scenario:-

Processor 1 read X : obtains 24 from the memory and caches it.

Processor 2 read X : obtains 24 from memory and caches it.

Again, processor 1 writes as X : 64, Its locally cached copy is updated. Now, processor 3 reads X, what value should it get?

Memory and processor 2 thinks it is 24 and processor 1 thinks it is 64.



As multiple processors operate in parallel, and independently multiple caches may possess different copies of the same memory block, this creates a cache coherence problem.

Cache coherence is the discipline that ensures that changes in the values of shared operands are propagated throughout the system in a timely fashion.

There are three distinct level of cache coherence :-

1. Every write operation appears to occur instantaneously.
2. All processors see exactly the same sequence of changes of values for each separate operand.
3. Different processors may see an operation and assume different sequences of values; this is known as non-coherent behavior

Cache Coherence Protocols

There are various Cache Coherence Protocols in multiprocessor system. These are :-

1. MSI protocol (Modified, Shared, Invalid)
2. MOSI protocol (Modified, Owned, Shared, Invalid)
3. MESI protocol (Modified, Exclusive, Shared, Invalid)
4. MOESI protocol (Modified, Owned, Exclusive, Shared, Invalid)

These important terms are discussed as follows:

- ❑ **Modified** – It means that the value in the cache is dirty, that is the value in current cache is different from the main memory.
- ❑ **Exclusive** – It means that the value present in the cache is same as that present in the main memory, that is the value is clean.
- ❑ **Shared** – It means that the cache value holds the most recent data copy and that is what shared among all the cache and main memory as well.
- ❑ **Owned** – It means that the current cache holds the block and is now the owner of that block, that is having all rights on that particular blocks.
- ❑ **Invalid** – This states that the current cache block itself is invalid and is required to be fetched from other cache or main memory.

Coherency Mechanisms

There are three types of coherence :

Directory-based

In a directory-based system, the data being shared is placed in a common directory that maintains the coherence between caches. The directory acts as a filter through which the processor must ask permission to load an entry from the primary memory to its cache. When an entry is changed, the directory either updates or invalidates the other caches with that entry.

Snooping

First introduced in 1983, snooping is a process where the individual caches monitor address lines for accesses to memory locations that they have cached. It is called a write invalidate protocol. When a write operation is observed to a location that a cache has a copy of and the cache controller invalidates its own copy of the snooped memory location.

Snarfing

It is a mechanism where a cache controller watches both address and data in an attempt to update its own copy of a memory location when a second master modifies a location in main memory. When a write operation is observed to a location that a cache has a copy of the cache controller updates its own copy of the snarfed memory location with the new data.