Memory Consistency

Memory consistency determines the order of accesses by different cores. Memory consistency and sequential consistency go hand-in-hand in ensuring programs execute properly. Without sequential consistency a program execution cannot be guaranteed to have the same outcome with time.

Memory Consistency
Coherence defines the order of accesses of threads to the same address. It does not say anything about accesses to different addresses.

Memory Consistency defines the order of accesses to different addresses.

Consistency Matters
Consistency ensures the program order is the same as the execution order. especially in OOO processors.

Why We Need Consistency
Additional ordering restrictions are needed to prevent the incorrect outcomes that can occur in Data Ready Flag synchronization and Thread termination.

Sequential Consistency
The result of any execution should be:

-As if the accesses were executed in-order by each processor
-As if the accesses among different processors were arbitrarily interleaved

Simplest Implementation of Sequential Consistency:

A core performs the next access only when all previous accesses are complete. Unfortunately this leads to poor performance.

Better Implementation of Sequential Consistency

-A core can reorder loads
-Detect when sequential consistency may be violated and fix it.

To detect and fix a SC violation:

The monitor the coherence traffic produced by other processors. Coherence traffic refers to the load and store commands, especially out of order commands.

Relax Consistency
Tell the programmers that they cannot expect sequential consistency for all instances.
Usually RD A → RD B must be programmed in correct order.

Data Races and Consistency
Data race: accesses to the same address by different cores that are not ordered by synchronization.

A data-race-free program cannot create data races. A key property is the program will behave the same in any consistency model. So the program can be debugged in a sequential consistency model, then run on a relaxed consistency model.

In a non-data-race-free program anything can happen.

Consistency Models

- -Sequential Consistency
- -Relaxed Consistency Models
    - -The key to these -- all of them support synchronization operations that ensure the correct order occurs.