06/04/24

Unit-5

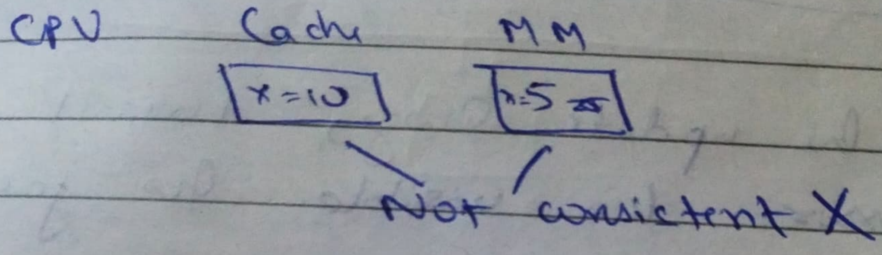
Memory & Cache Coherency

(completly theoretical)

Consistency

needs to be maintained when sharing data.

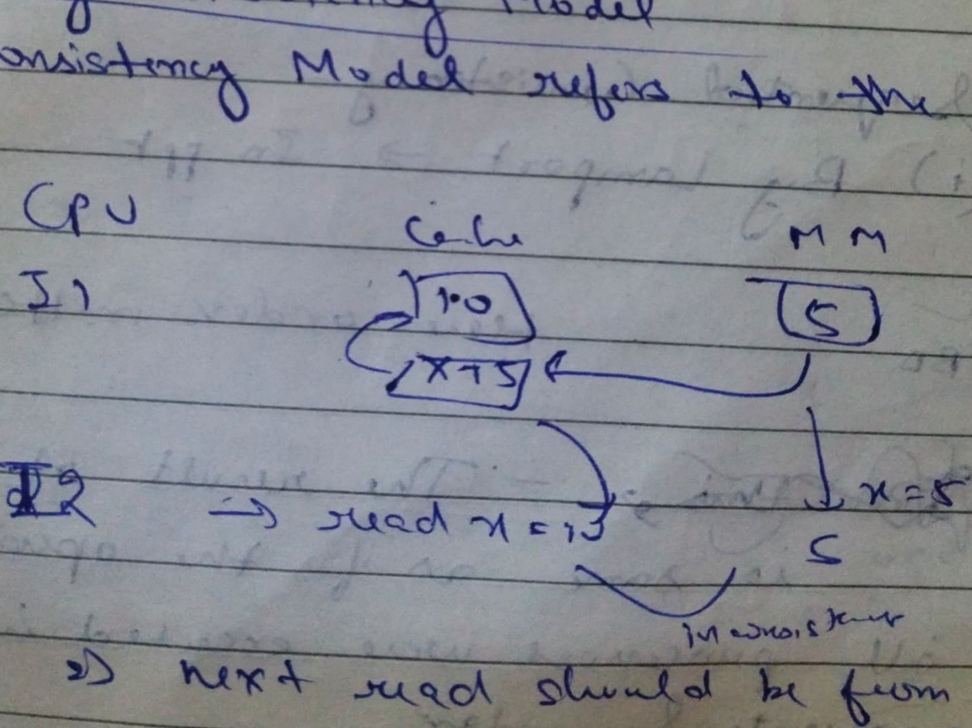
what?



Memory Consistency Model

A Consistency Model refers to the d. —

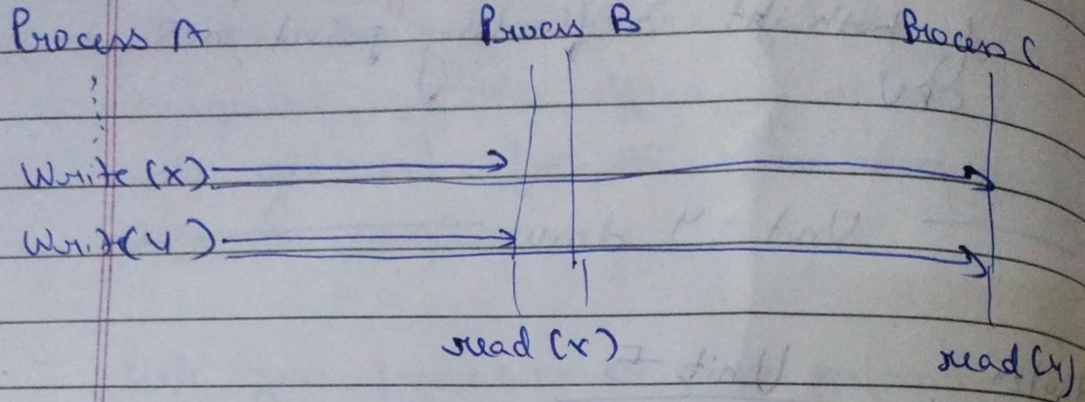
from PPT



=> next read should be from where there was last write to

Types

① Strict Consistency



Any read to memory location X returns value stored by the most recent writes are instantaneously visible to all processes & an absolute global time order is maintained.

* Jese hi update hoga, turant saare processor ko visible ho jaega,

② Sequential Consistency

2 definitions (i) By Lamport → In ppt

(ii) ~~Not~~ TBD Here order matters, ~~Ex~~

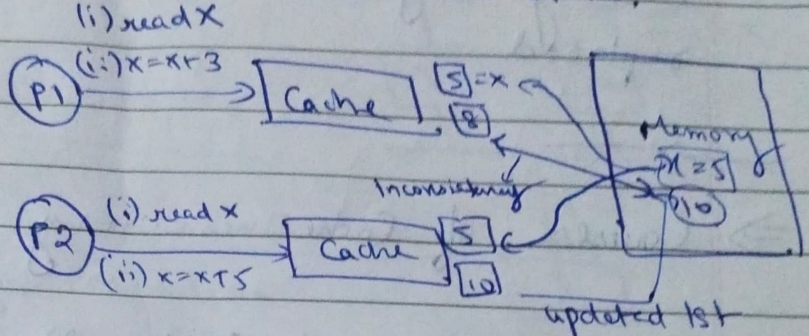
~~For this~~ - The result of any executn is same as if the operations of all processors were executed in some sequential order.

a → b → c → d → o/p

~~Rest~~ Read rest from PPT.

Cache Coherency

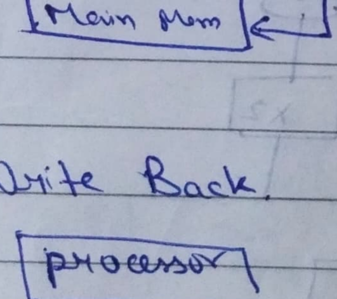
↳ It refers to the problem of keeping the data in the caches consistent.



But P1 cache and main memory values are not matching - Inconsistency

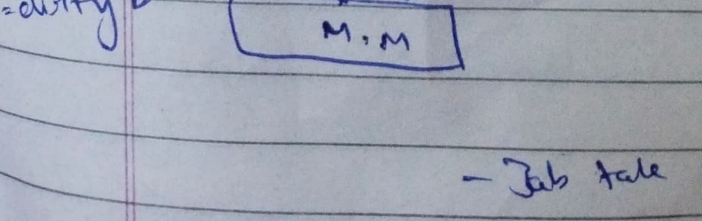
→ Cache Write Policies

① Write through



update memory immediately when a WRITE occurs.

② Write Back



The updated value is marked. in cache and updated later = Dirty Bit

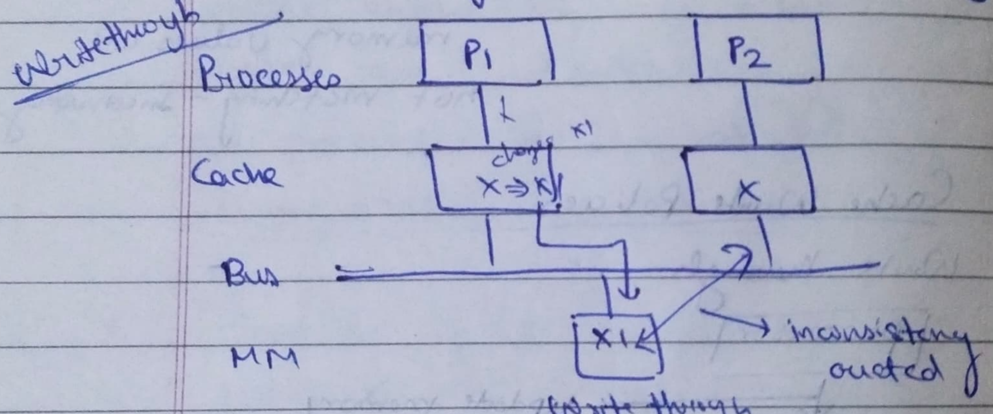
- Jab tak next time convert na ho, tab tak don't update.

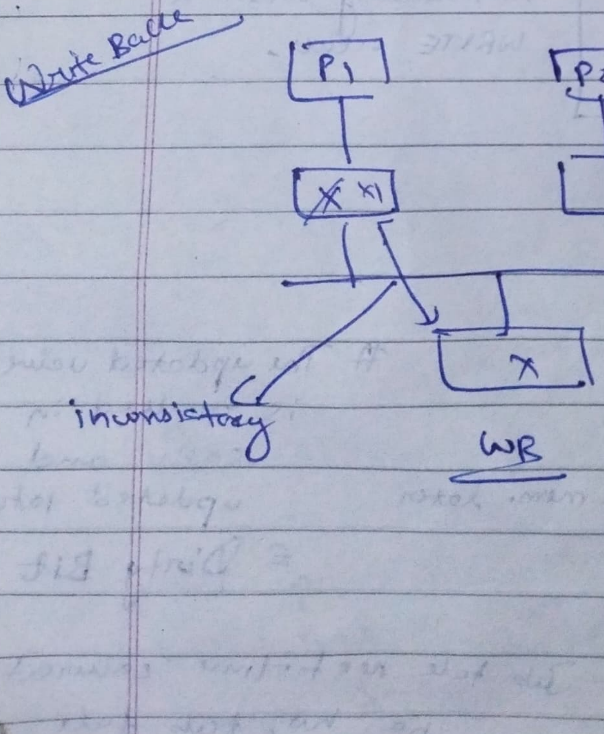
→ When data is written to a cache, a di bit is set for the affected block. The modified block is written to memory only when the block is to b replaced

—x—

⇒ Causes of Cache Inconsistency

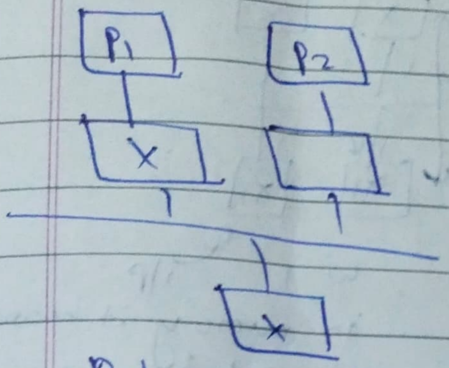
(i) Inconsistency due to Data Sharing





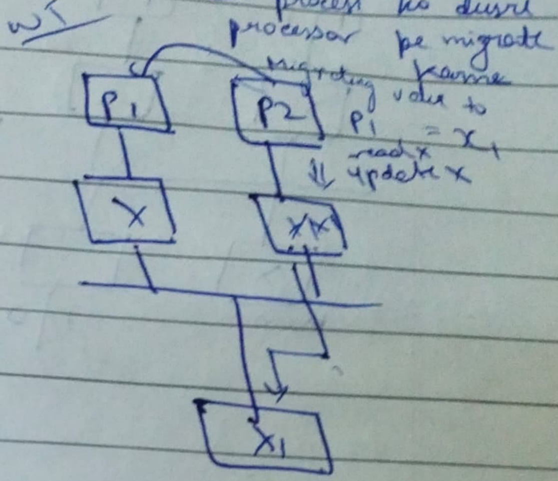
② Inconsistency after Process Migration

↳ ek process ko dusre processor pe migrate karna



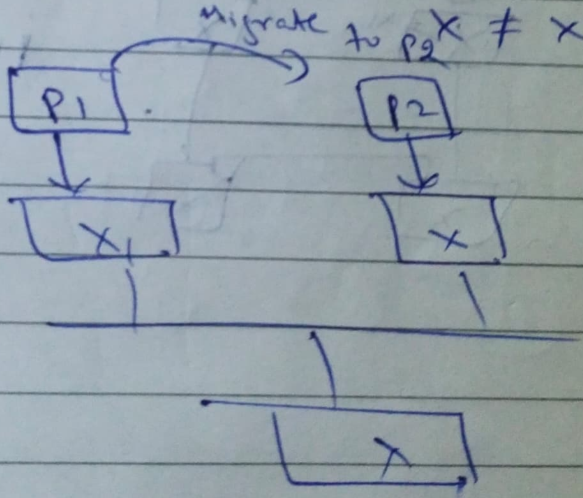
Before Migration

WT



W.T.
P1 made X1
P2 migrates ~~value~~ and updated to P1
Now inconsistency, bec
it migrates X1 to P1
but P1 has X,

WriteBack

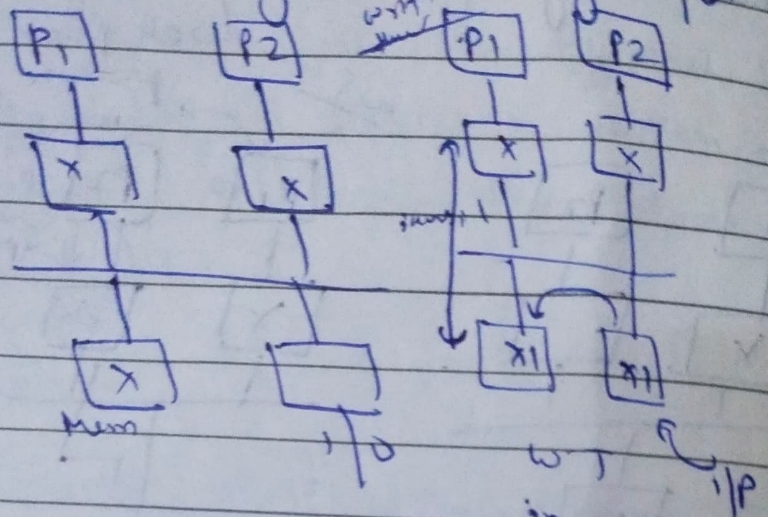


X ≠ X1 ⇒ inconsistency

P1 ne x ki value update karta
P2 me dedi = X1.

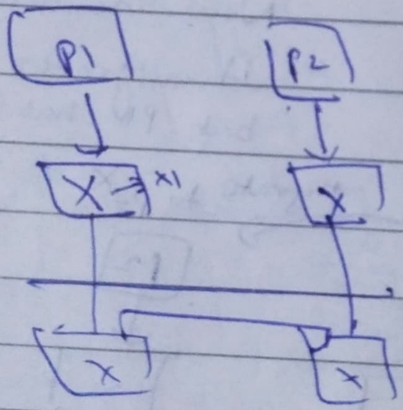
This is inconsistent.

③ Inconsistency caused by I/O.



in memory it becomes x1 by i/p but cache has x ⇒ inconsist

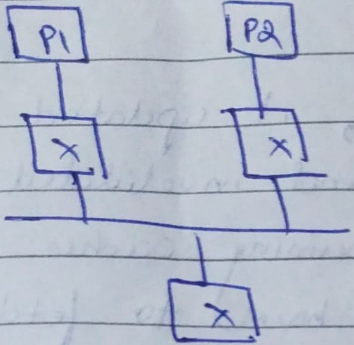
Write back



P1 cache will by passed. o/p will get inconsistent copy of x.

Cache Coherence Protocols (to remove inconsistency soln)

(1) Snoopy Bus Protocol : UMA m/c



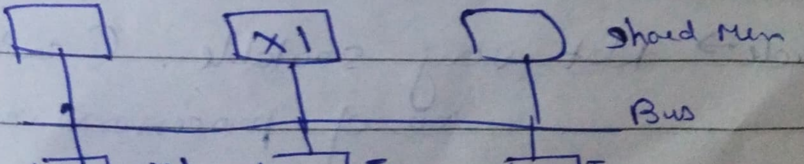
Snoopy ; means investigate / monitor

~~It is a~~ cache controller monitors the transaction through snoopy bus

- A bus allows all processors in the system to observe ongoing transactions. If a bus transaction threatens the consistent state of a locally cached object, the cache controller can take appropriate actions to invalidate or update the local copy.

- 2 basic protocols

a) Write Invalidate.



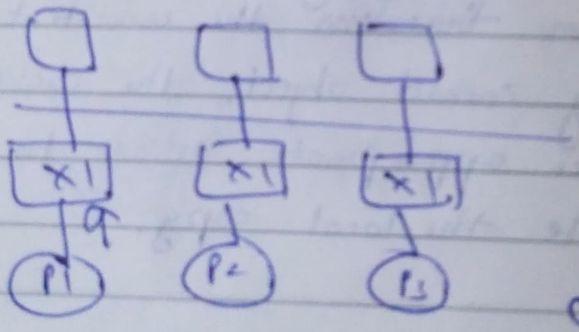
Consistent copies of block x are in shared memory.

When a local cache copy is modified, Write Invalidate invalidates all remote copies of cache (invalidated items are called 'dirty')

If P1 did x → x1, it updated in shared memory, and invalidated the value of all remaining caches.
At nex time, you have to fetch from the shared memory

(b) Write-update (Write Broadcast)

~~sso token~~



The new modified x1 be broadcasted (updated) to all cache copies via the bus

Drawback → Not very scalable. (cant inc no. of processors much)

↓

Solⁿ ≡ Directory Based Protocol

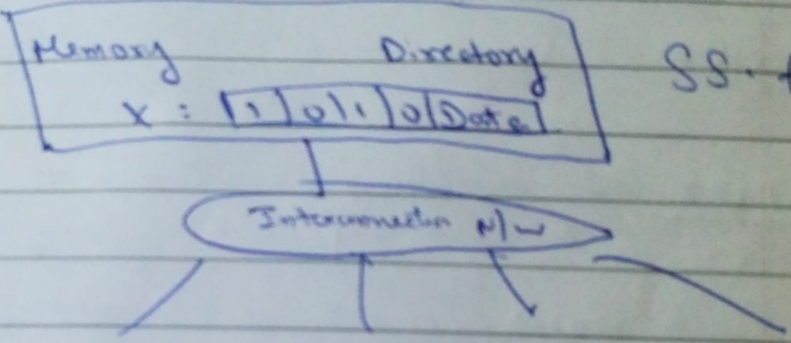
② Directory Based Protocol : NUMA

- used in scalable Multiprocessor systems (NUMA m/c) interconnected using Crossbar Switch.

- uses cache directory to keep track of the status of all cache blocks.

3 Types

① Full Map Directory



SS. taken

→ Ther has burden of maintaing cache of such by mm.

② Limited

jaha value present hai, sirf unhi cache values ko directory me rakhna hai ↓ C0, C2

limitation → C1 and C3 cant access that directory

Chained Directory

Directories are already mentioned

CT → chain tree

No load on mem
All caches are able to access
sequential is consistency

—— Unit-5 ——