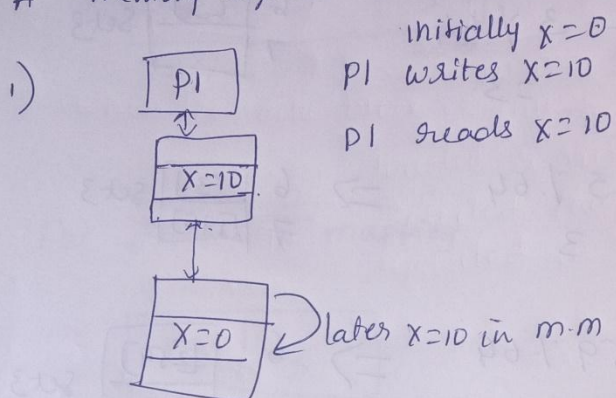


- 1) Cache coherence:- It defines the behavior of reads and writes to the same memory location. It is one of the issues in multiprocessor systems. Two processors can see different values in their individual cache for the same variable.

A memory system is coherent if



| 2) Event | P1 | P2 | memory contents |
|-------------|----|----|-----------------|
| P1 reads x | 1 | | 1 |
| P2 read x | 1 | 1 | |
| P1 writes x | 0 | 1 | 0 |

P1 writes
 P2 reads (P2 should read updated value)

3) P1 ↓ writes P2 ↓ write

i.e $X=10$
 P1 → $X=100$
 P2 → $X=200$
 $X=100$ both should be seen
 $X=200$ in other processor's
 cache otherwise

P3 → should see in its cache

① $X=100$ then $X=200$ not as 200 & then 100

write through cache

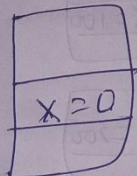
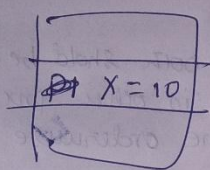
P1 reads $X=0$ at time to

At t_1 P1 $X=1$

immediately in main memory also is updated $X=1$

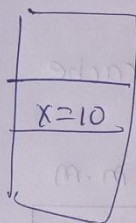
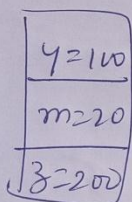
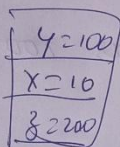
$X=0$ P_1 read from m.m

P_1 changes at to $X=10$ in m.m $X=0$

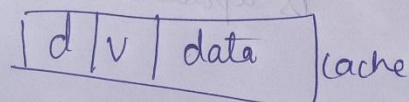
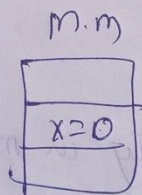
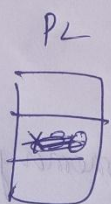
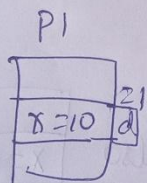


m.m.

When the cache block is full & new data to be loaded & victim block $X=10$ is selected that time, it will be updated in m.m.



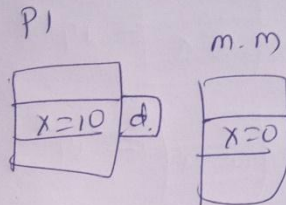
in multiprocessor sm



$d=1$ means it is modified so in m.m

X value 0 is aborted/invalidated

if p2 wants to read the data, that time
updated copy of p1 cache i.e $x=10$ is stored
in m.m



P2 has requested x , ~~so~~ and dirty bit of $x=10$ is
1 so $x=10$ is copied to m.m

