

Baseline

L1-0



L1-1



L1-2

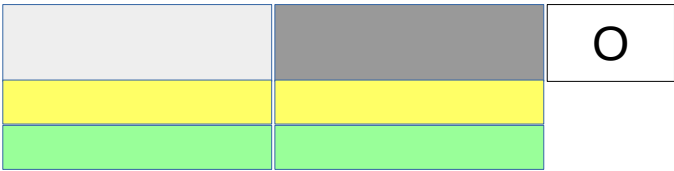


L1-3

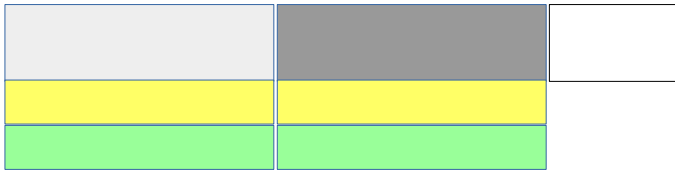


L2-0

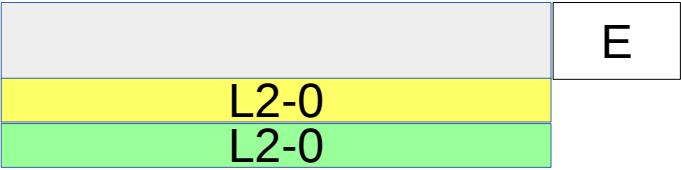
Sharer
Owner



L2-1

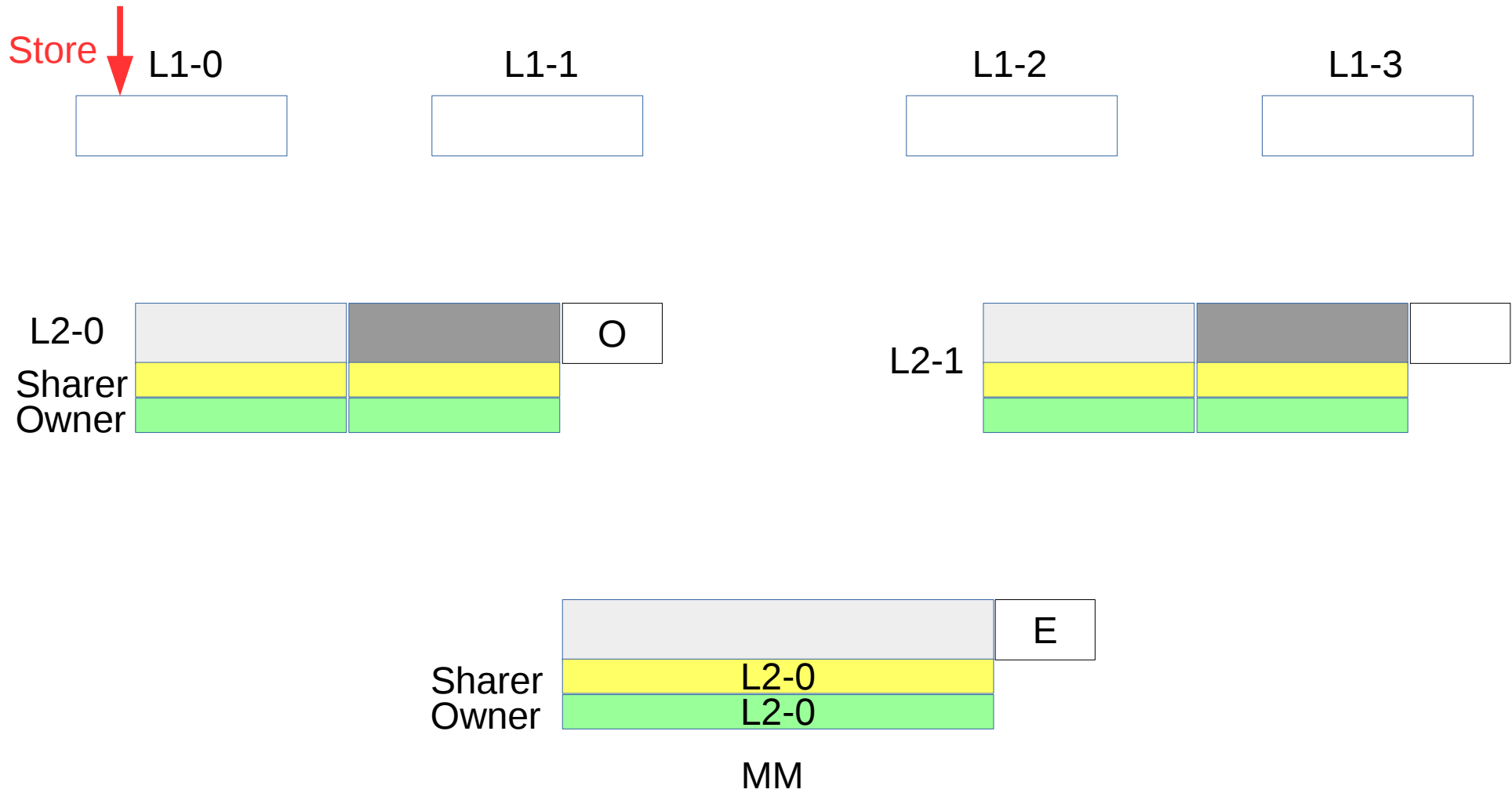


Sharer
Owner

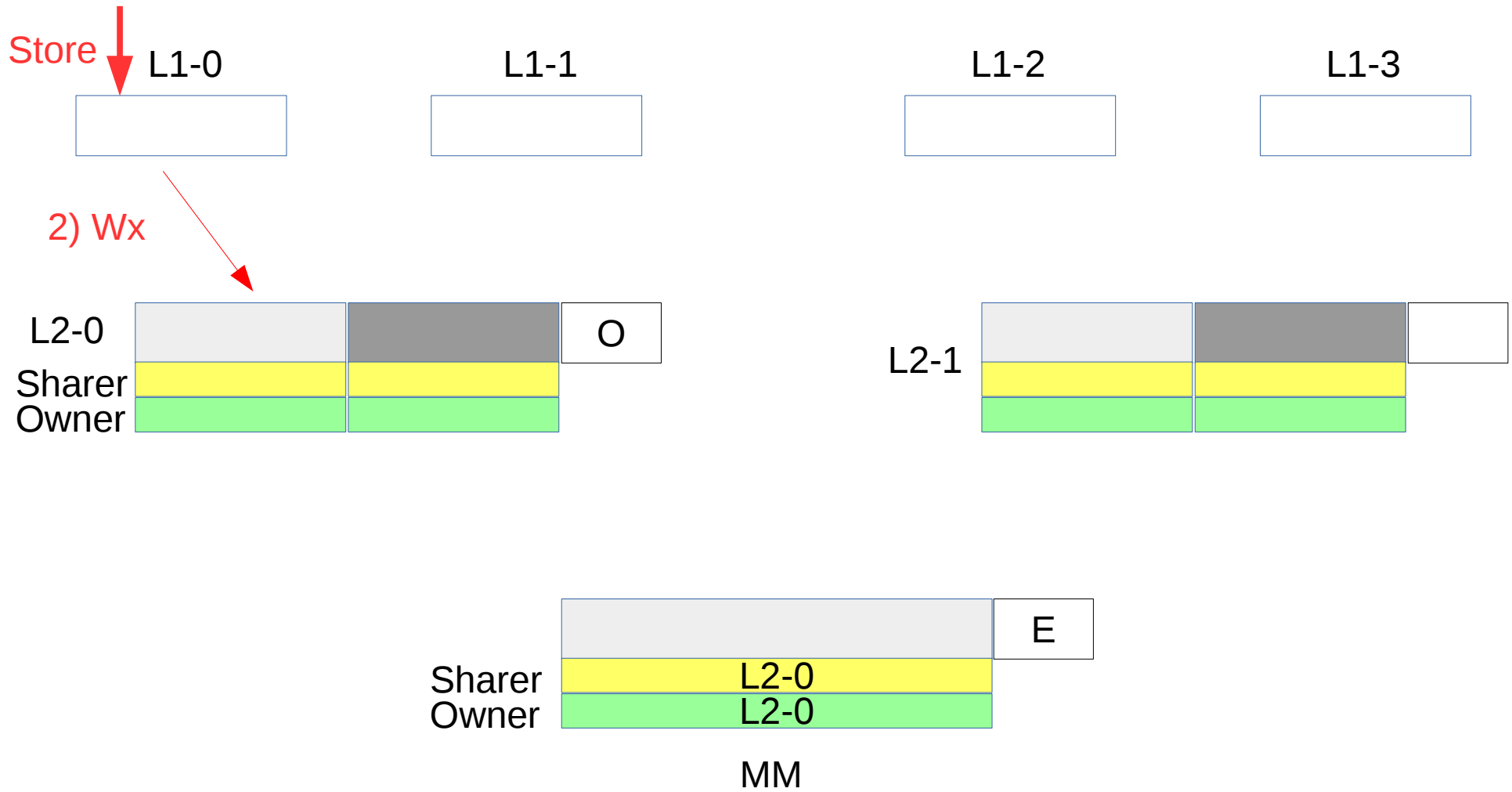


MM

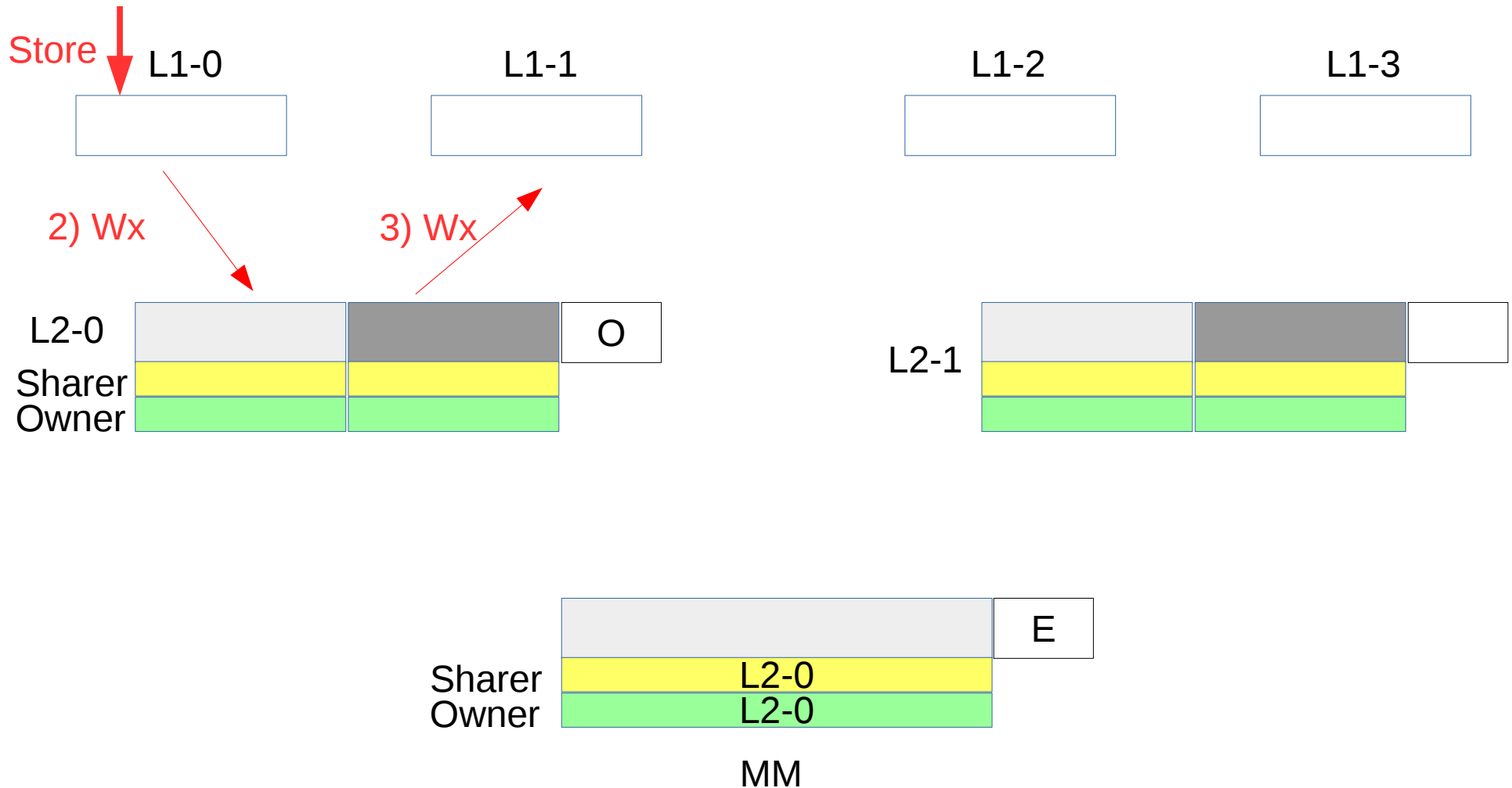
1) store



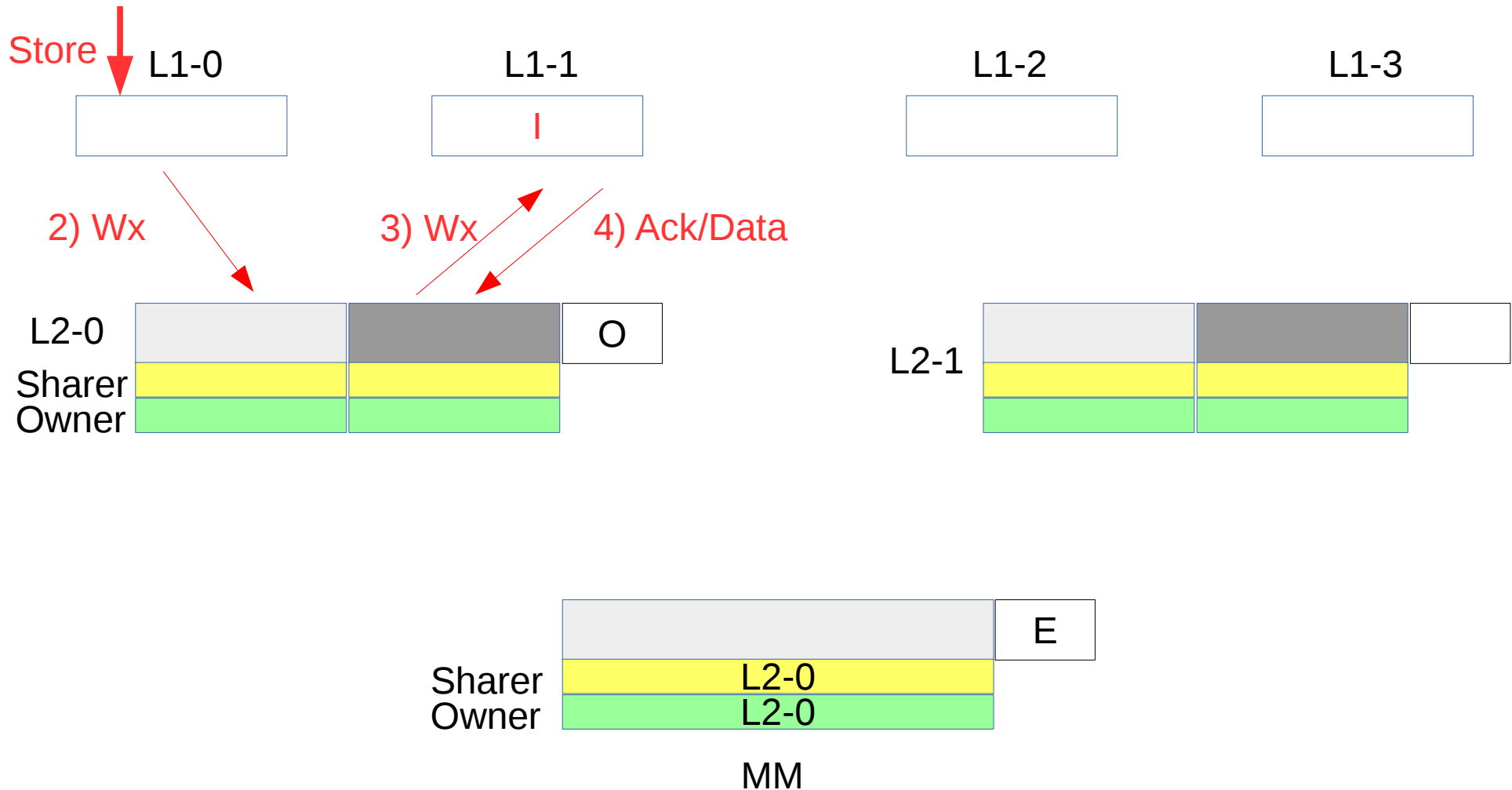
2) Wx (updown) from L1-0 to L2-0



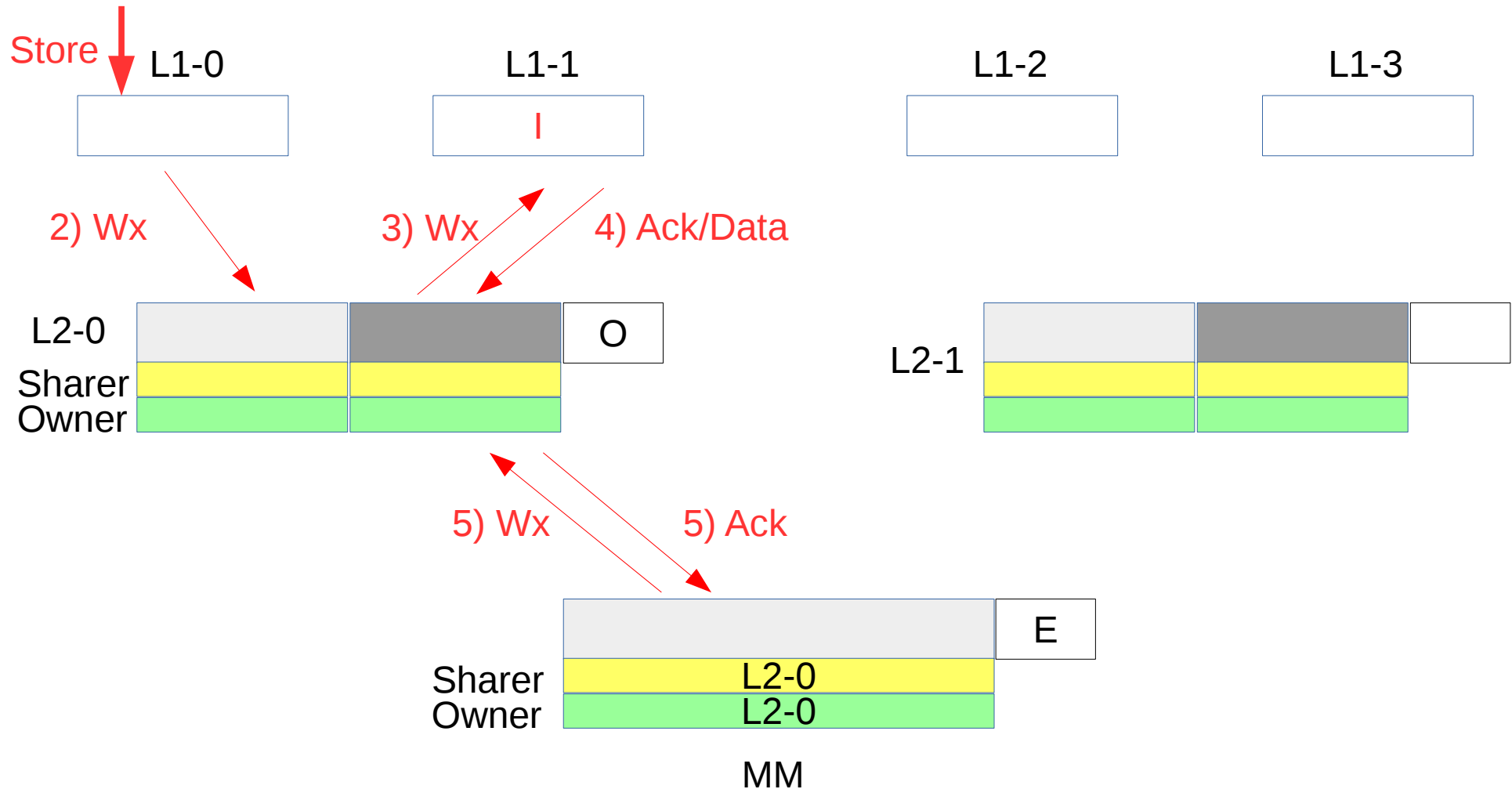
3) L2-0 calls to invalidate all the sharers (any other mod than the L1-0) With write-rq (downup), and remove the sharer/owner (nothing if there are no sharers/owners)



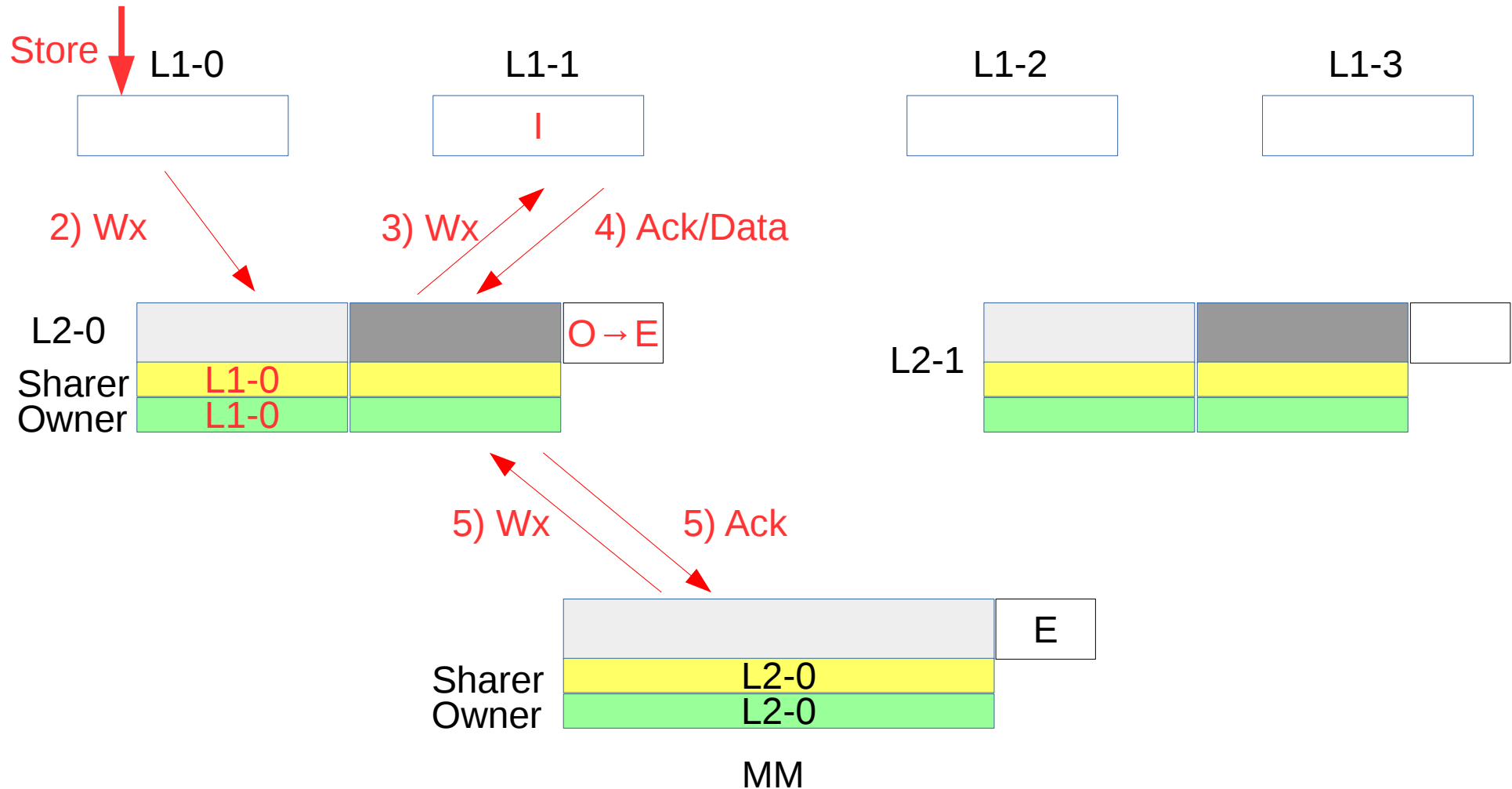
4) L1-0 responds either with Data (M/N/O) or Ack(E/S) goes to I. It continues till all Invalidations are over (no state changes at this point)



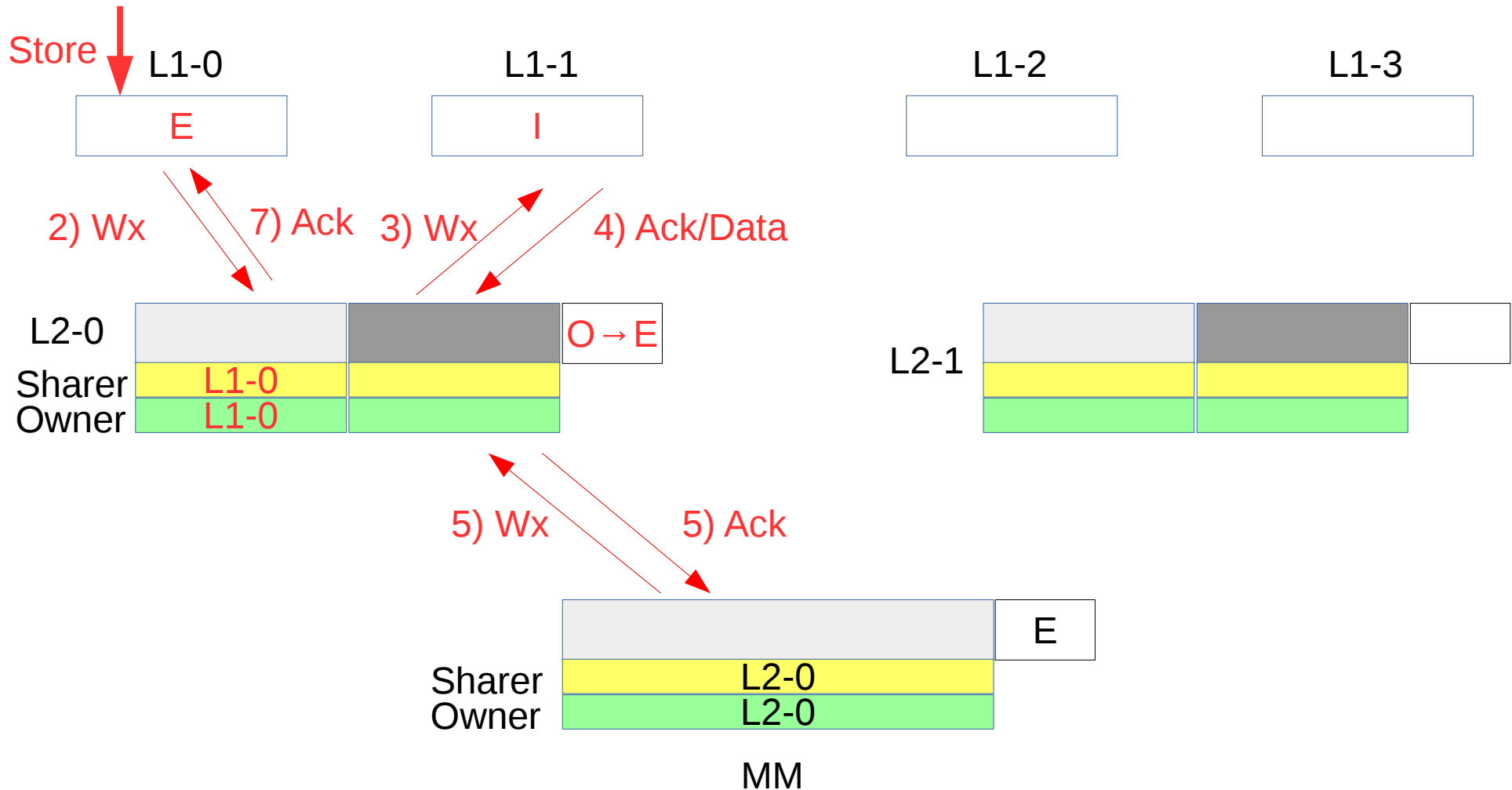
5) After this point when the data is provided (or just ack) based on the state (O) another write request is generated to lower mods to update the whole hierarchy. There are no other sharer than L2-0 so no invalidation is sent to L2-1



6) The response is received by L2-0, the tag is set and we go to E state no matter what (since any other cache has been invalidated). The requesting mod (L1-0) becomes sharer/owner



7) The response is sent by L2-0 (which is either data from L1-1 or Ack), and state changes to E since all other units in this level are invalidated



8) Store will be completed in this step.

