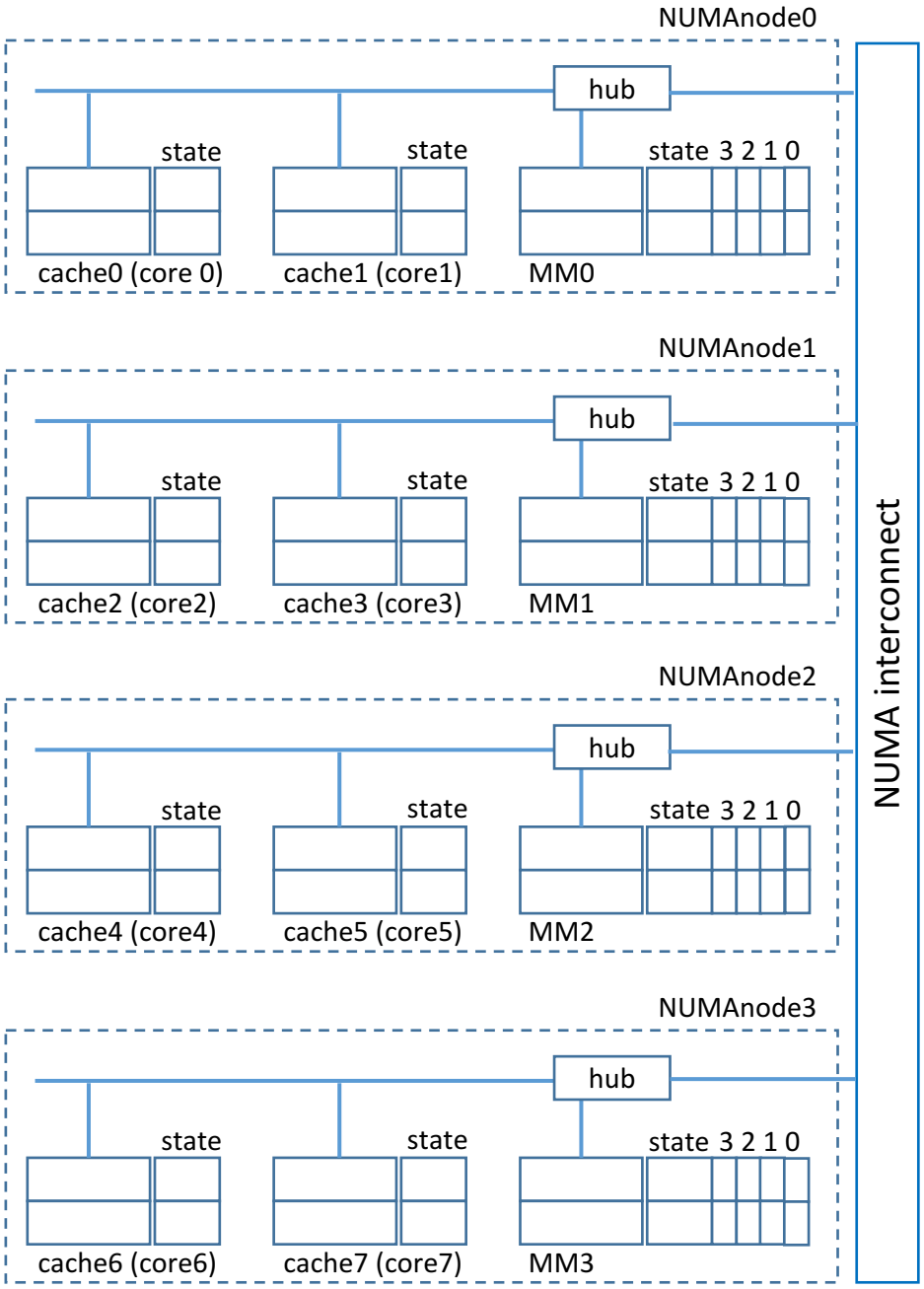


Coherence commands

- **Core:** $PrRd_i$ and $PrWr_i$, being I the core number doing the action
- **Snoopy:** $BusRd_j$, $BusRdX_j$ $BusUpgr_j$ and $Flush_j$, being j the snoopy/cache number doing the action
- **Hub/directoty:** $RdReq_{i\rightarrow j}$, $WrReq_{i\rightarrow j}$, $UpgrReq_{i\rightarrow j}$, $Dreply_{i\rightarrow j}$, $Fetch_{i\rightarrow j}$, $Invalidate_{i\rightarrow j}$, $Ack_{i\rightarrow j}$ and $WriteBack_{i\rightarrow j}$, from NUMAnode i to NUMAnode j

Line state in cache

- M (Modified), S (Shared), I (Invalid)

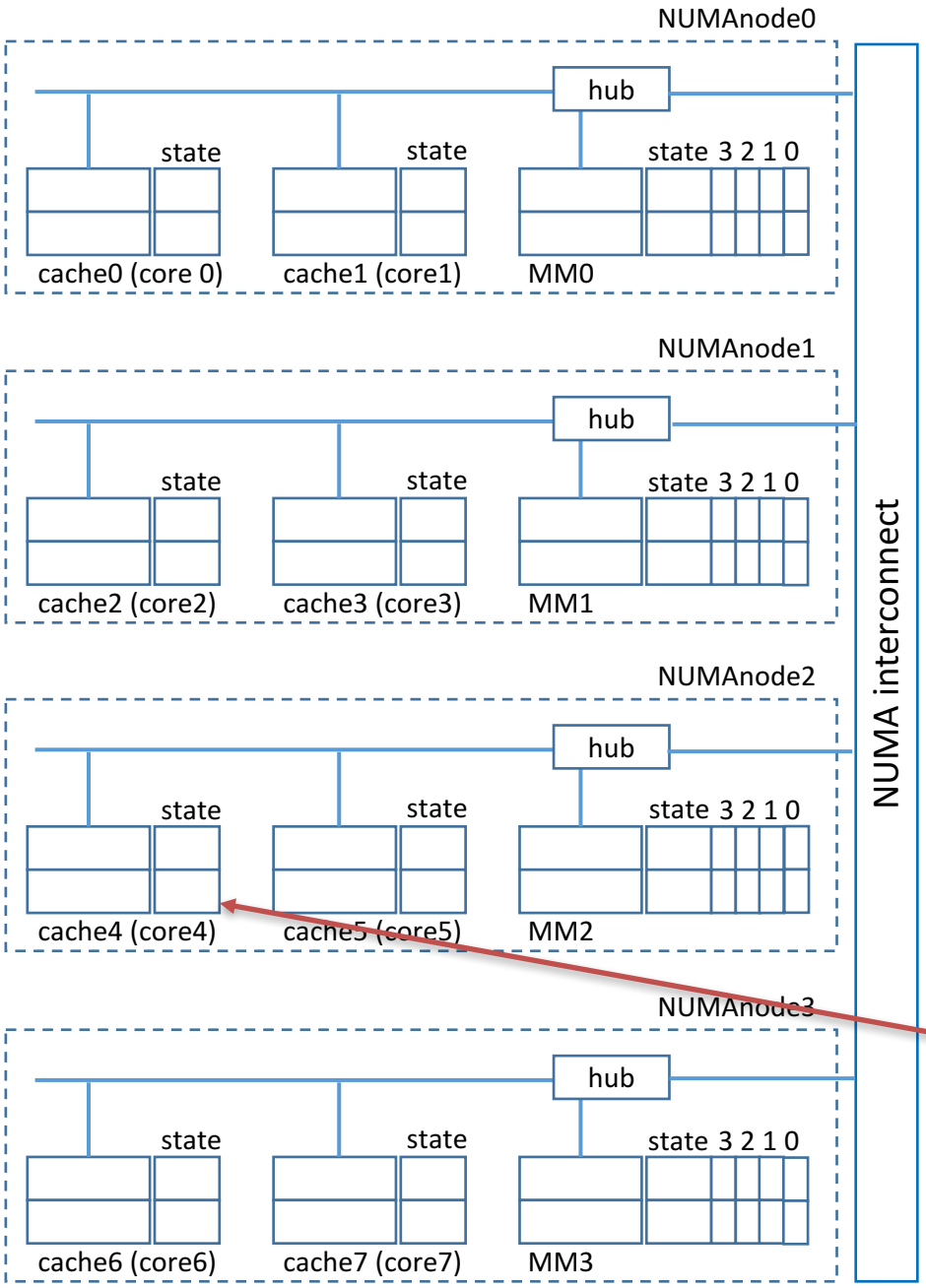
Line state in main memory

- M (Modified), S (Shared), U (Uncached)

Cache line size = 32 bytes
Main memory size = 8 GB
Cache size in each processor = 4 MB

(a) Amount of bits taken by:

- Snoopy : to keep coherence between caches in a NUMA node
- The Directory : to keep coherence between NUMA nodes.



Coherence commands

- **Core:** $PrRd_i$ and $PrWr_i$, being I the core number doing the action
- **Snoopy:** $BusRd_j$, $BusRdX_j$ $BusUpgr_j$ and $Flush_j$, being j the snoopy/cache number doing the action
- **Hub/directoty:** $RdReq_{i\to j}$, $WrReq_{i\to j}$, $UpgrReq_{i\to j}$, $Dreply_{i\to j}$, $Fetch_{i\to j}$, $Invalidate_{i\to j}$, $Ack_{i\to j}$ and $WriteBack_{i\to j}$, from NUMAnode i to NUMAnode j

Line state in cache

- M (Modified), S (Shared), I (Invalid)

Line state in main memory

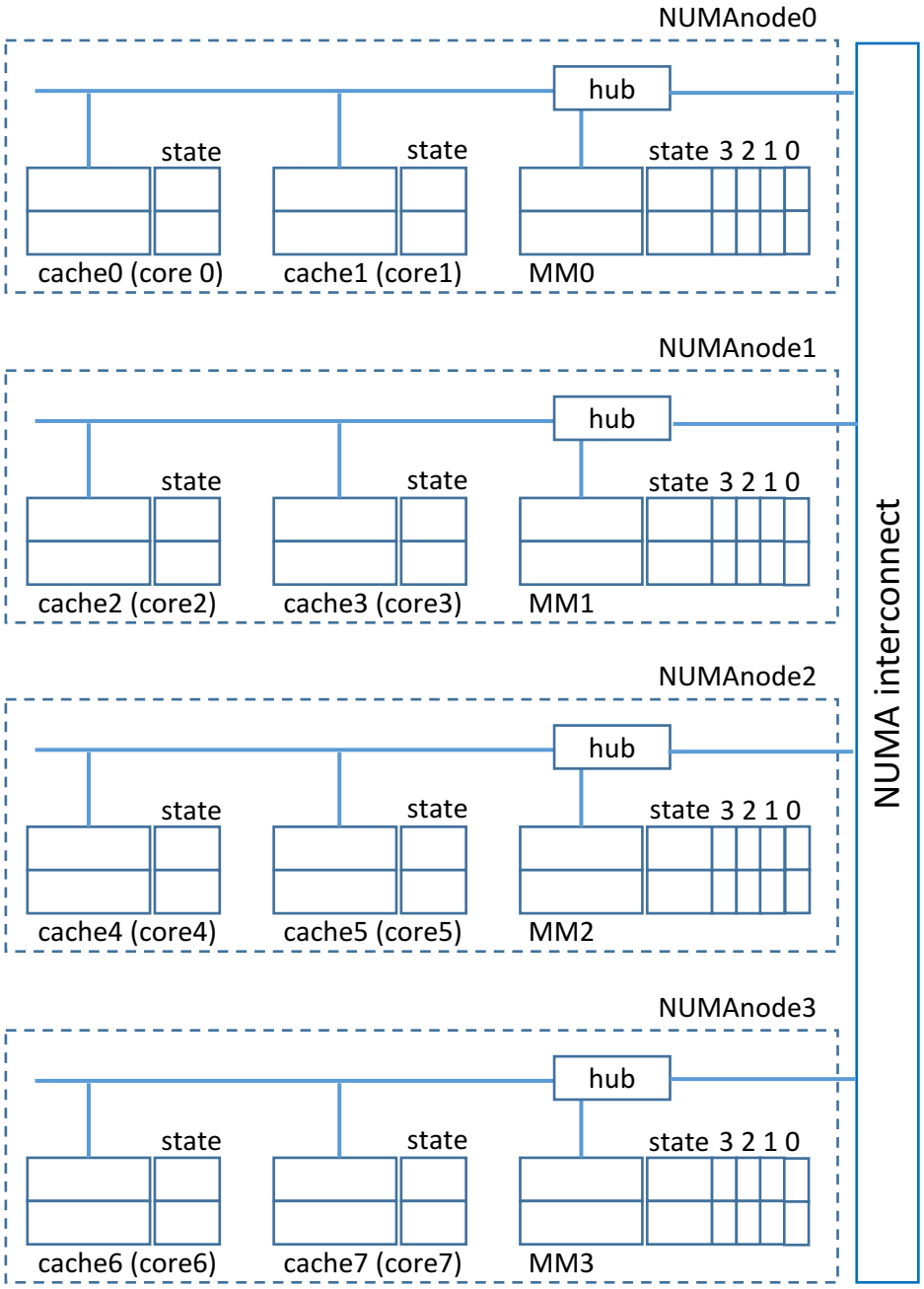
- M (Modified), S (Shared), U (Uncached)

Cache line size = 32 bytes
Main memory size = 8 GB
Cache size in each processor = 4 MB

(a) Amount of bits taken by:

- Snoopy : to keep coherence between caches in a NUMA node

States = M,S,I : 2 bits
(for each cache line)



Coherence commands

- **Core:** $PrRd_i$ and $PrWr_i$, being I the core number doing the action
- **Snoopy:** $BusRd_j$, $BusRdX_j$ $BusUpgr_j$ and $Flush_j$, being j the snoopy/cache number doing the action
- **Hub/directoty:** $RdReq_{i\to j}$, $WrReq_{i\to j}$, $UpgrReq_{i\to j}$, $Dreply_{i\to j}$, $Fetch_{i\to j}$, $Invalidate_{i\to j}$, $Ack_{i\to j}$ and $WriteBack_{i\to j}$, from NUMAnode i to NUMAnode j

Line state in cache

- M (Modified), S (Shared), I (Invalid)

Line state in main memory

- M (Modified), S (Shared), U (Uncached)

Cache line size = 32 bytes
Main memory size = 8 GB
Cache size in each processor = 4 MB

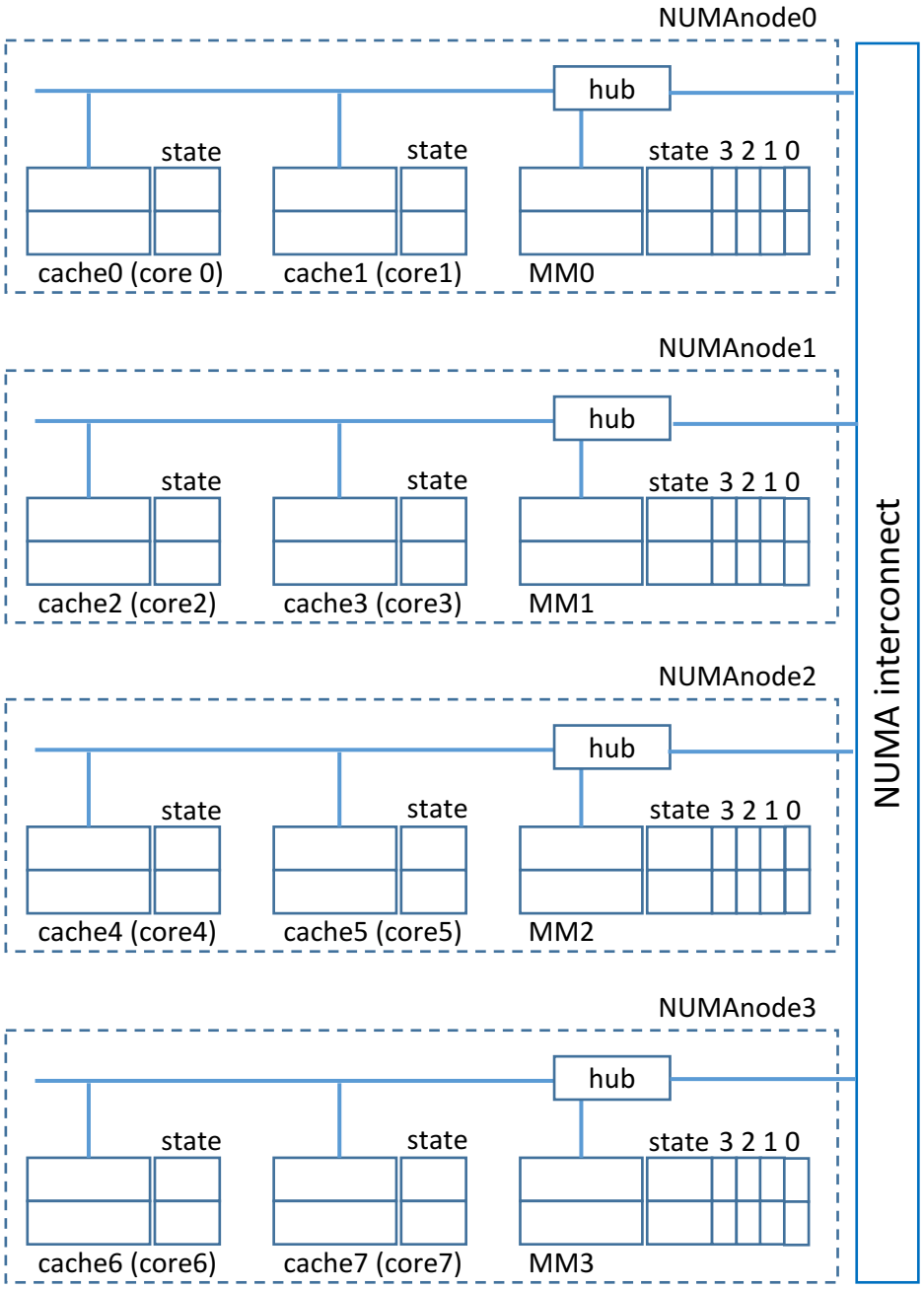
(a) Amount of bits taken by:

- Snoopy : to keep coherence between caches in a NUMA node

States = M,S,I : 2 bits

$$\#\text{cache lines} = \frac{4\text{ MB}}{32\text{ bytes}} = \frac{4 \times 2^{20}}{2^5} = 2^{17}$$

$$\#\text{bits} = \#\text{cache lines} \times 2\text{ bits} = 2^{17} \times 2 = 2^{18}$$



Coherence commands

- **Core:** $PrRd_i$ and $PrWr_i$, being I the core number doing the action
- **Snoopy:** $BusRd_j$, $BusRdX_j$ $BusUpgr_j$ and $Flush_j$, being j the snoopy/cache number doing the action
- **Hub/directoty:** $RdReq_{i\to j}$, $WrReq_{i\to j}$, $UpgrReq_{i\to j}$, $Dreply_{i\to j}$, $Fetch_{i\to j}$, $Invalidate_{i\to j}$, $Ack_{i\to j}$ and $WriteBack_{i\to j}$, from NUMAnode i to NUMAnode j

Line state in cache

- M (Modified), S (Shared), I (Invalid)

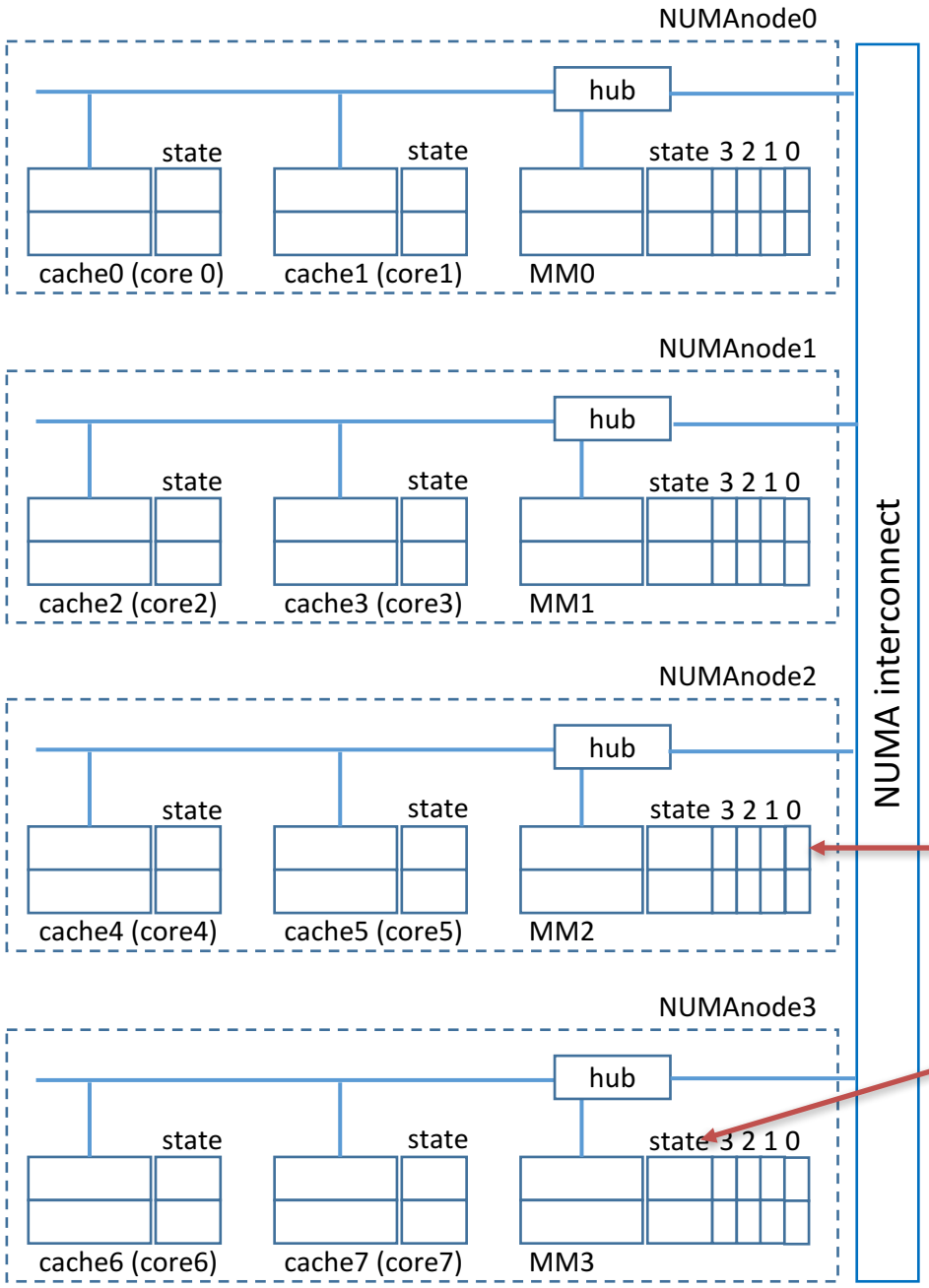
Line state in main memory

- M (Modified), S (Shared), U (Uncached)

Cache line size = 32 bytes
Main memory size = 8 GB
Cache size in each processor = 4 MB

(a) Amount of bits taken by (at each NUMA node):

- The Directory : to keep coherence between NUMA nodes.



Coherence commands

- **Core:** $PrRd_i$ and $PrWr_i$, being I the core number doing the action
- **Snoopy:** $BusRd_j$, $BusRdX_j$ $BusUpgr_j$ and $Flush_j$, being j the snoopy/cache number doing the action
- **Hub/directoty:** $RdReq_{i\rightarrow j}$, $WrReq_{i\rightarrow j}$, $UpgrReq_{i\rightarrow j}$, $Dreply_{i\rightarrow j}$, $Fetch_{i\rightarrow j}$, $Invalidate_{i\rightarrow j}$, $Ack_{i\rightarrow j}$ and $WriteBack_{i\rightarrow j}$, from NUMAnode i to NUMAnode j

Line state in cache

- M (Modified), S (Shared), I (Invalid)

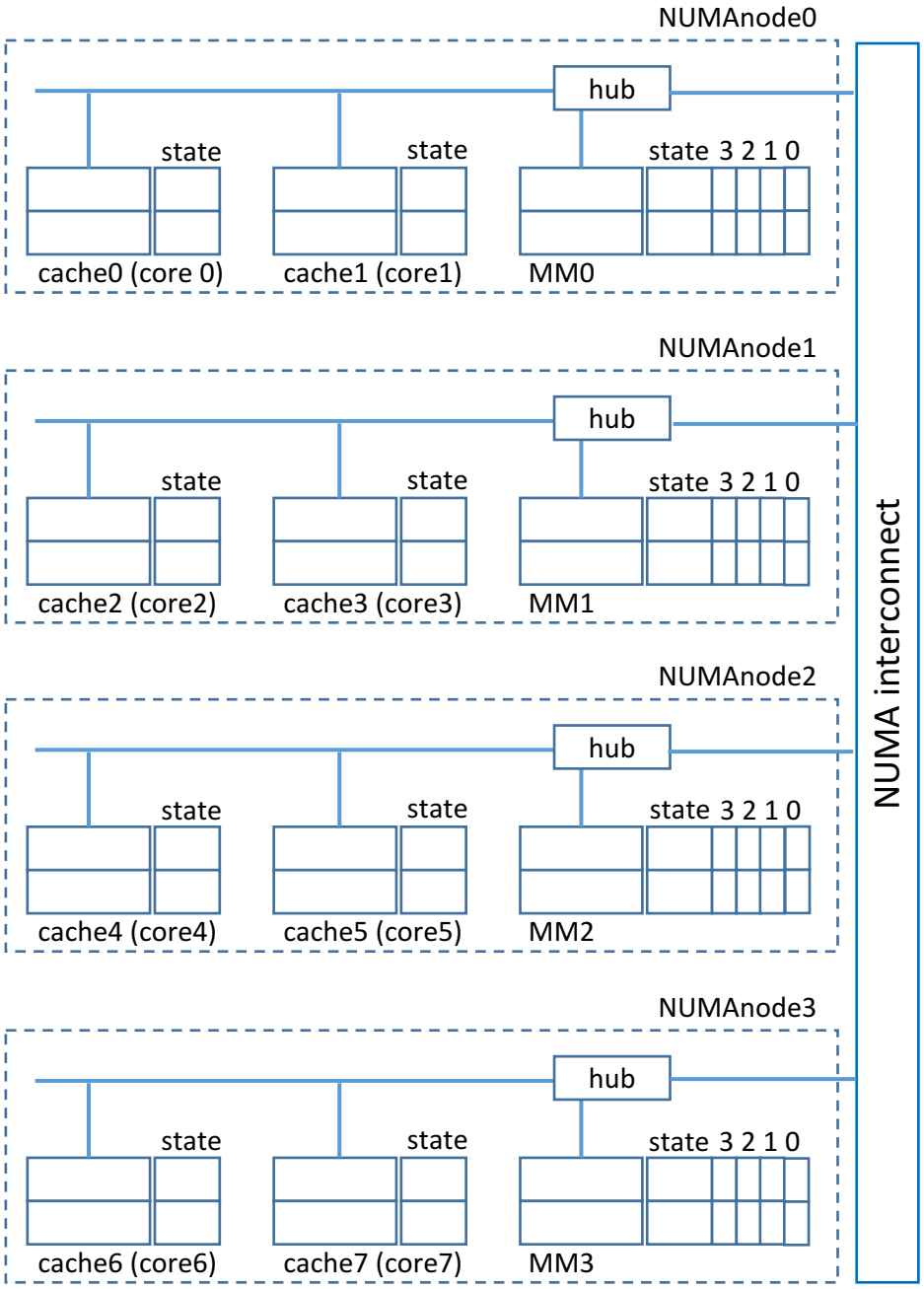
Line state in main memory

- M (Modified), S (Shared), U (Uncached)

Cache line size = 32 bytes
Main memory size = 8 GB
Cache size in each processor = 4 MB

Presence bits 0/1 (or Sharers list)
(1 bit for each NUMA node)

Status bits: M, S, U = 2 bits
(Modified, Shared , Uncached)



Coherence commands

- **Core:** $PrRd_i$ and $PrWr_i$, being I the core number doing the action
- **Snoopy:** $BusRd_j$, $BusRdX_j$ $BusUpgr_j$ and $Flush_j$, being j the snoopy/cache number doing the action
- **Hub/directoty:** $RdReq_{i\to j}$, $WrReq_{i\to j}$, $UpgrReq_{i\to j}$, $Dreply_{i\to j}$, $Fetch_{i\to j}$, $Invalidate_{i\to j}$, $Ack_{i\to j}$ and $WriteBack_{i\to j}$, from NUMAnode i to NUMAnode j

Line state in cache

- M (Modified), S (Shared), I (Invalid)

Line state in main memory

- M (Modified), S (Shared), U (Uncached)

Cache line size = 32 bytes
Main memory size = 8 GB
Cache size in each processor = 4 MB

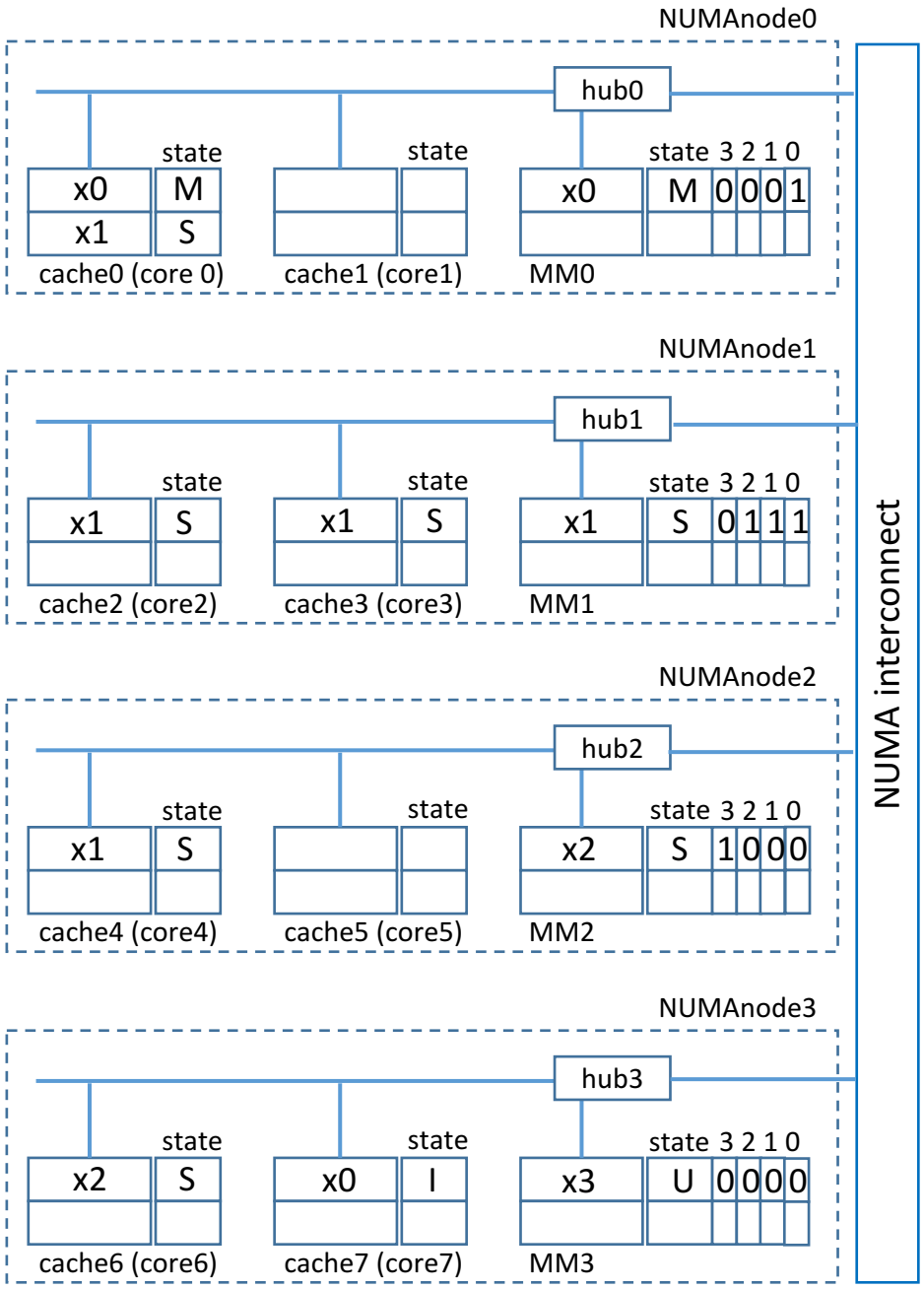
(a) Amount of bits taken by (at each NUMA node):

- The Directory : to keep coherence between NUMA nodes.

States = M,S,U : 2 bits
Presence bits = 4 bits
} 6 bits

$$\#\text{memory lines} = \frac{2\ \text{GB}}{32\ \text{bytes}} = \frac{2 \times 2^{30}}{2^5} = 2^{26}$$

$$\#\text{bits} = \#\text{memory lines} \times 6\ \text{bits} = 2^{26} \times 6$$

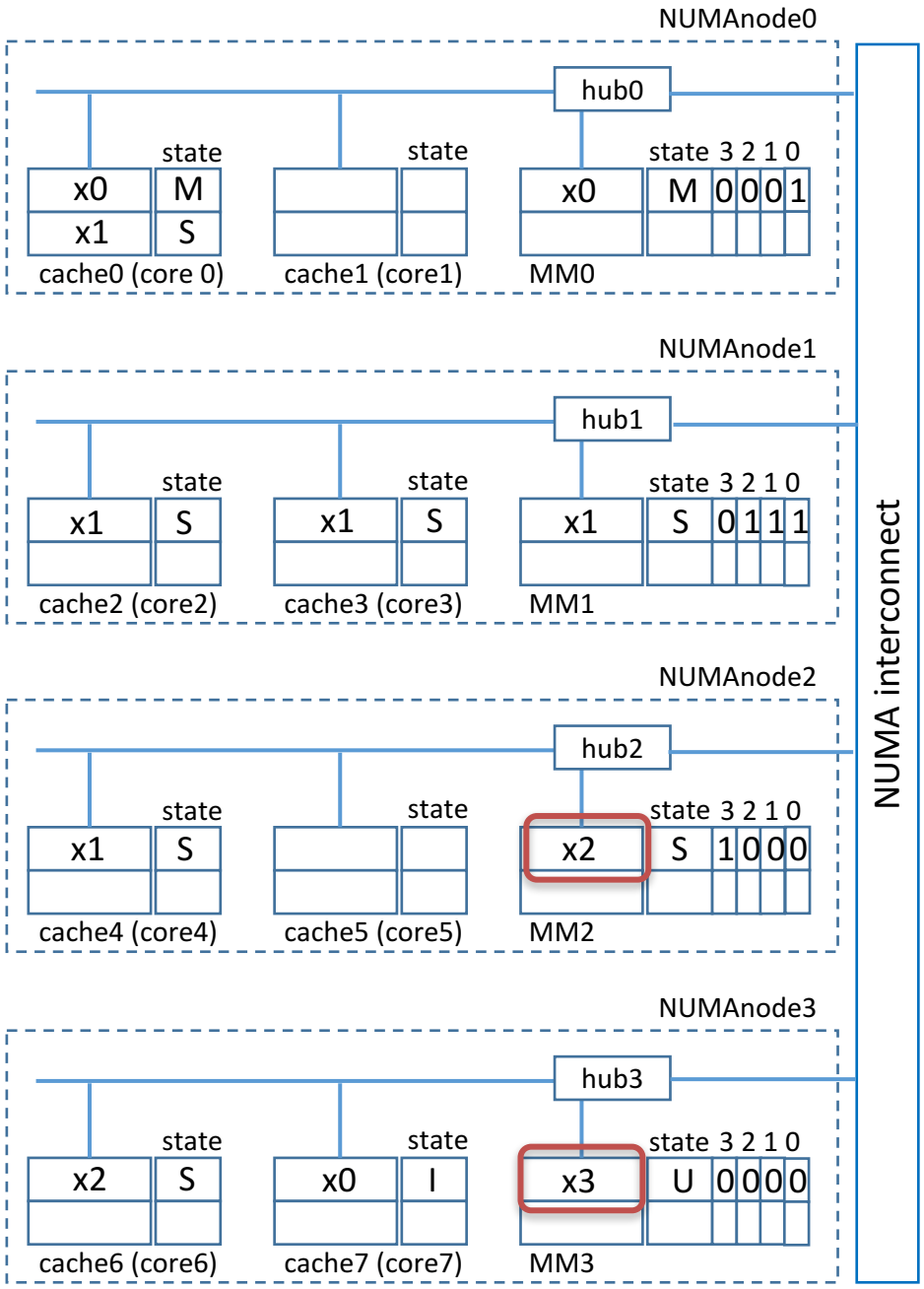


Assuming the initial state of the memories (caches and MM) shown:

- only two lines are represented for each memory (cache or MM),
- variables x0 , x1 , x2 and x3 , 4 bytes wide each.
- variables x2 and x3 reside in the same cache line
- variables x0 and x1 reside in a different cache line each.

(b) Indicate which variable (only one) is not in a correct state either in a cache or in main memory and the reason for that.

Based on you answer, write that variable in the correct state in the appropriate memories (cache and/or MM).

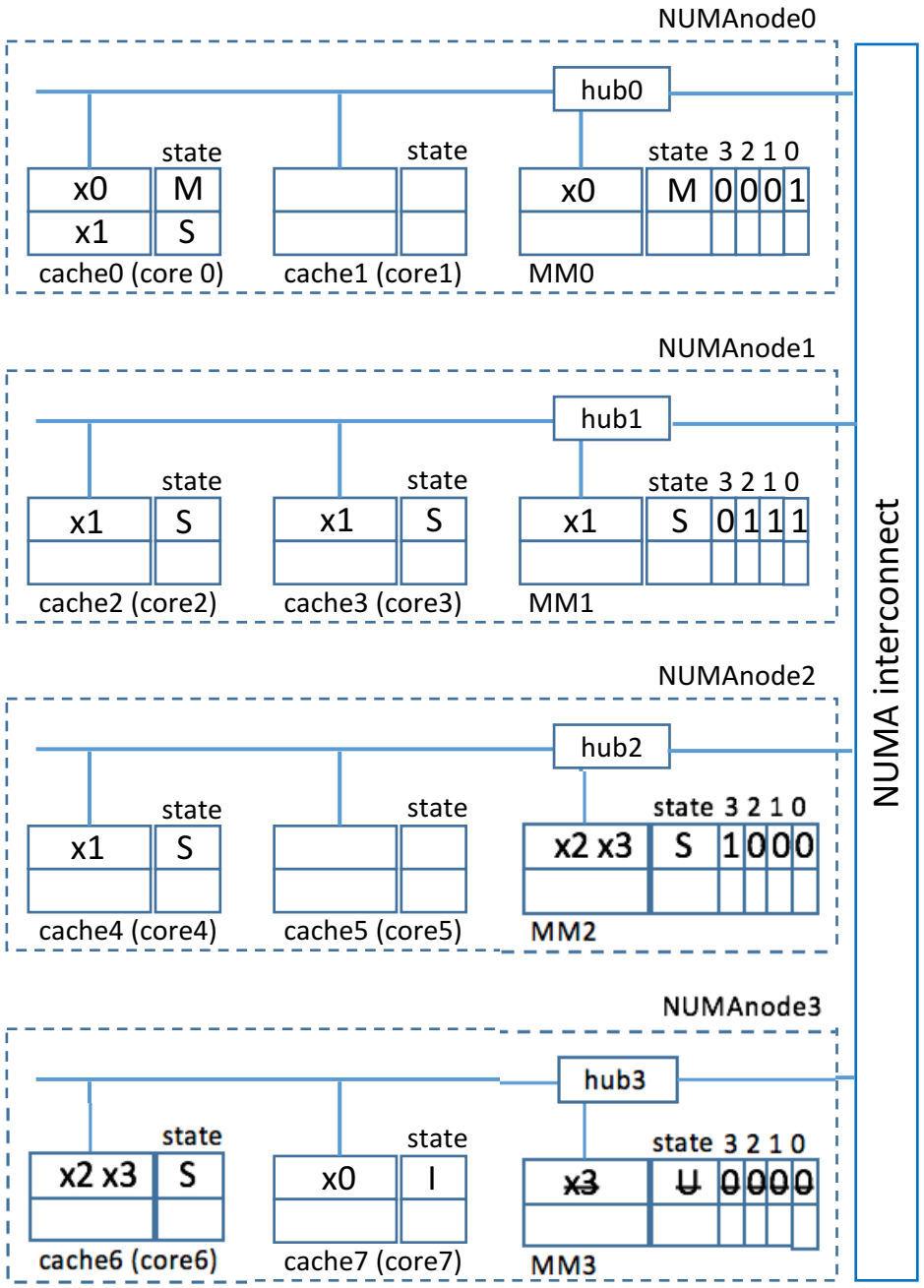


Assuming the initial state of the memories (caches and MM) shown:

- only two lines are represented for each memory (cache or MM),
- variables x0 , x1 , x2 and x3 , 4 bytes wide each.
- **variables x2 and x3 reside in the same cache line**
- variables x0 and x1 reside in a different cache line each.

(b) Indicate which variable (only one) is not in a correct state either in a cache or in main memory and the reason for that.

Based on you answer, write that variable in the correct state in the appropriate memories (cache and/or MM).

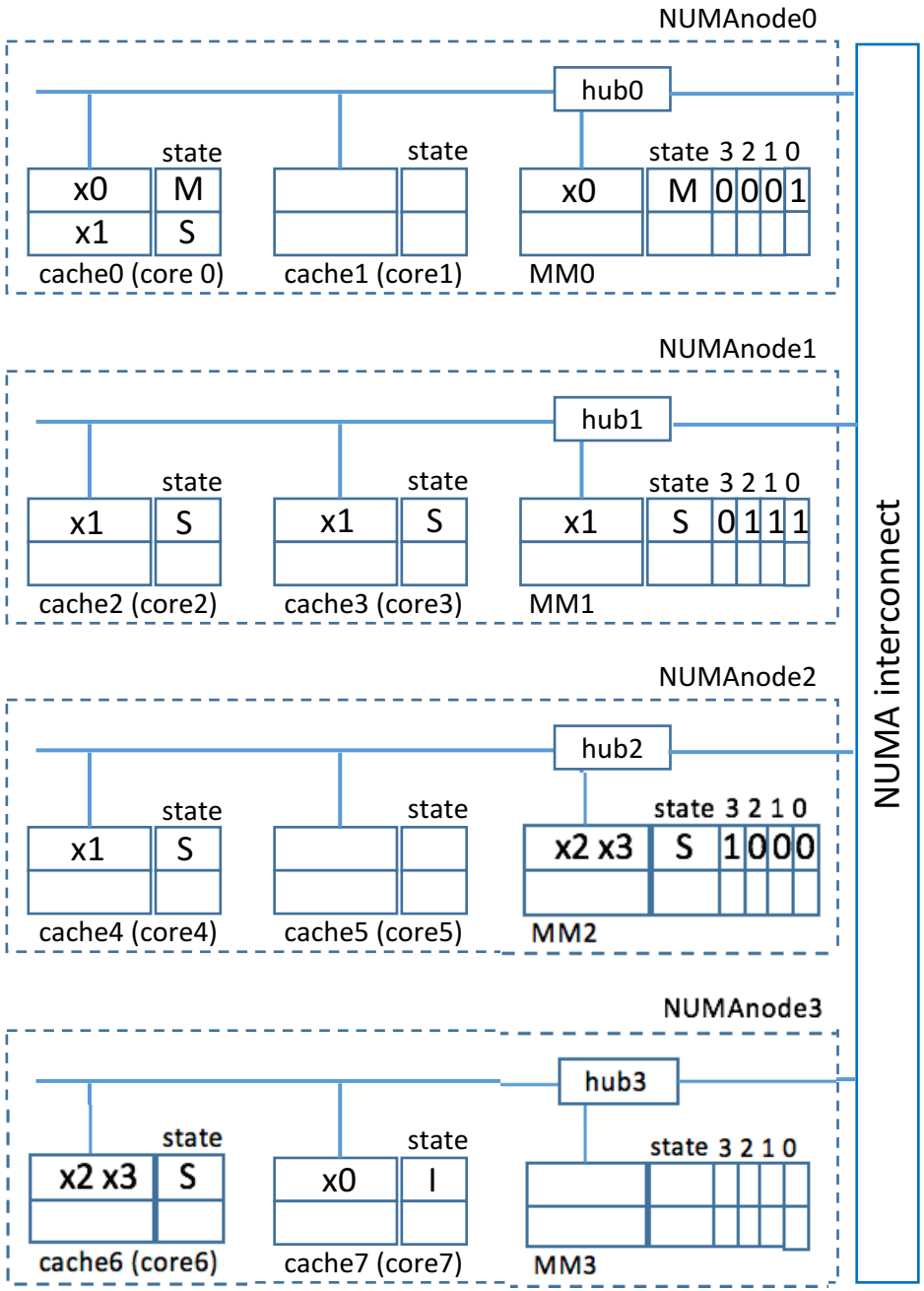


Assuming the initial state of the memories (caches and MM) shown:

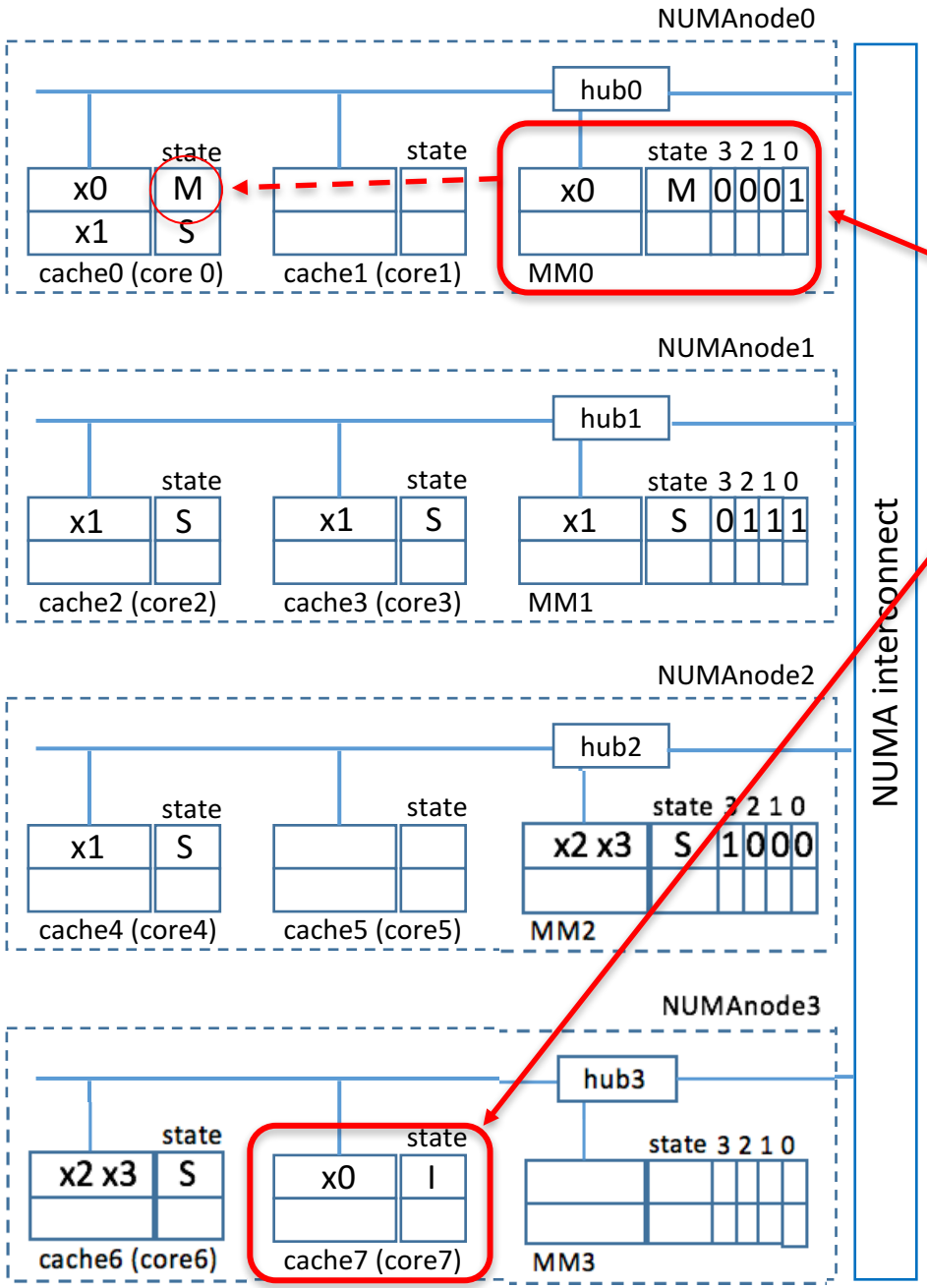
- only two lines are represented for each memory (cache or MM),
- variables x0 , x1 , x2 and x3 , 4 bytes wide each.
- **variables x2 and x3 reside in the same cache line**
- variables x0 and x1 reside in a different cache line each.

(b) Indicate which variable (only one) is not in a correct state either in a cache or in main memory and the reason for that.

Variables x2 and x3 cannot be mapped in two different NUMA nodes
Each memory line **can only exist in one home node.**
Variable x3 does not exist in NUMAnode3
We just add it in the same line as the one containing x2 in NUMAnode2.



(c) core 7 writes x0. Indicate final state in memories.

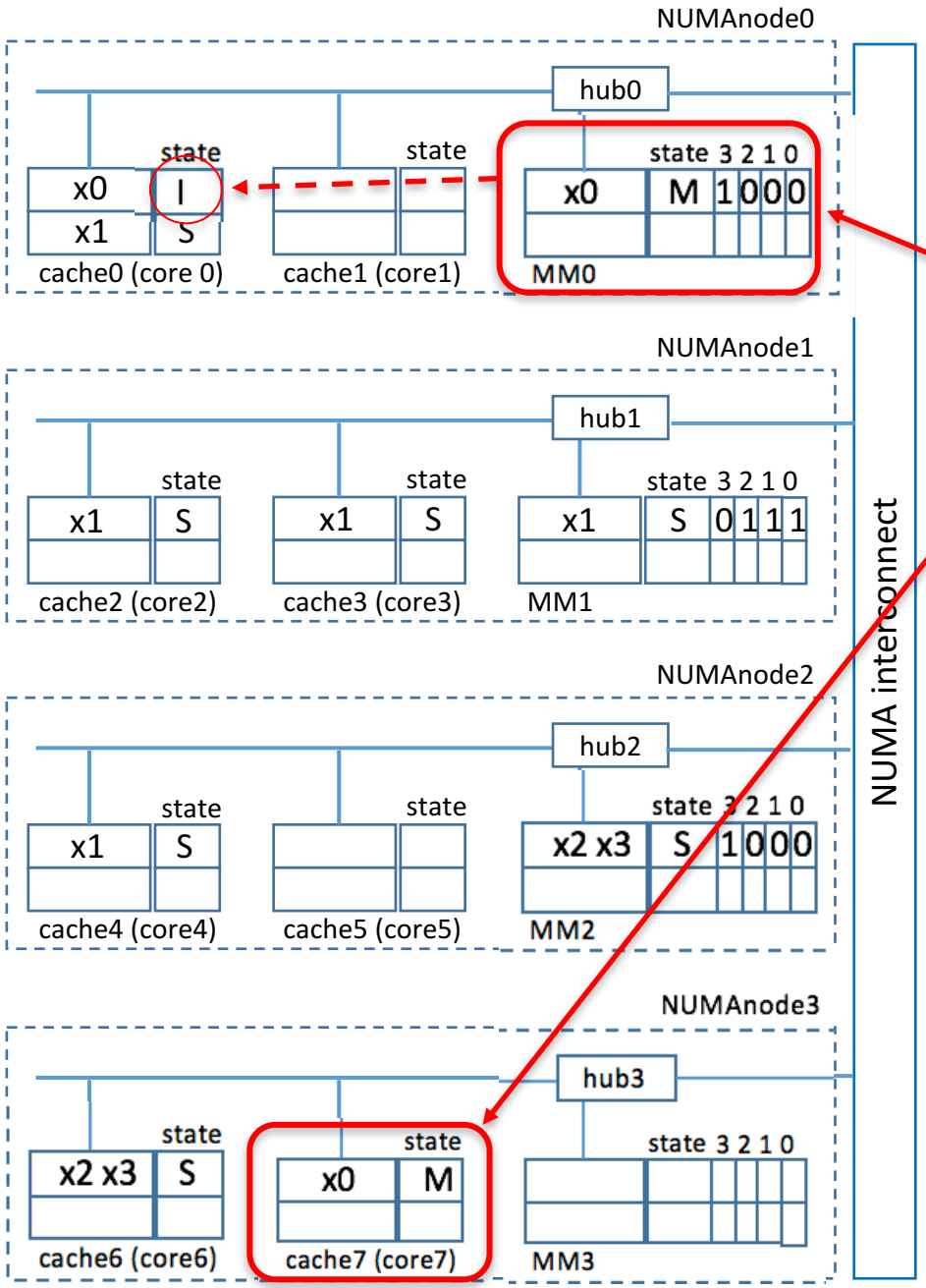


(c) core 7 writes x0. Indicate final state in memories.

Current state of x0:

Directory information: x0 has only a copy in some cache of NUMA node0 with Modified state

Snoopy information: x0 has Invalid state in cache of core7 → Miss in cache of core7



(c) core 7 writes x0. Indicate final state in memories.

Current state of x0:

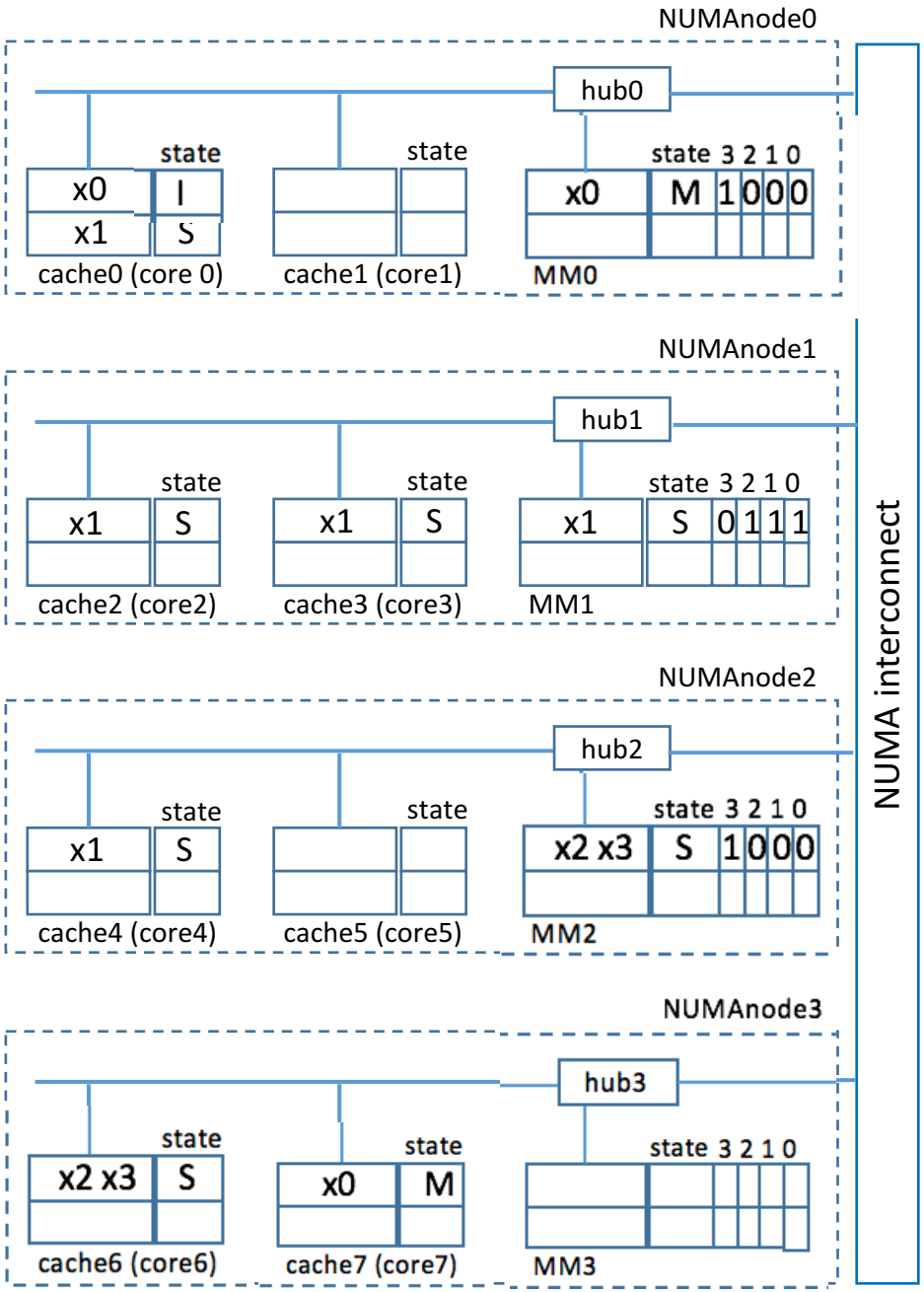
Directory information: x0 has only a copy in some cache of NUMA node0 with Modified state

Snoopy information: x0 has Invalid state in cache of core7 → Miss in cache of core7

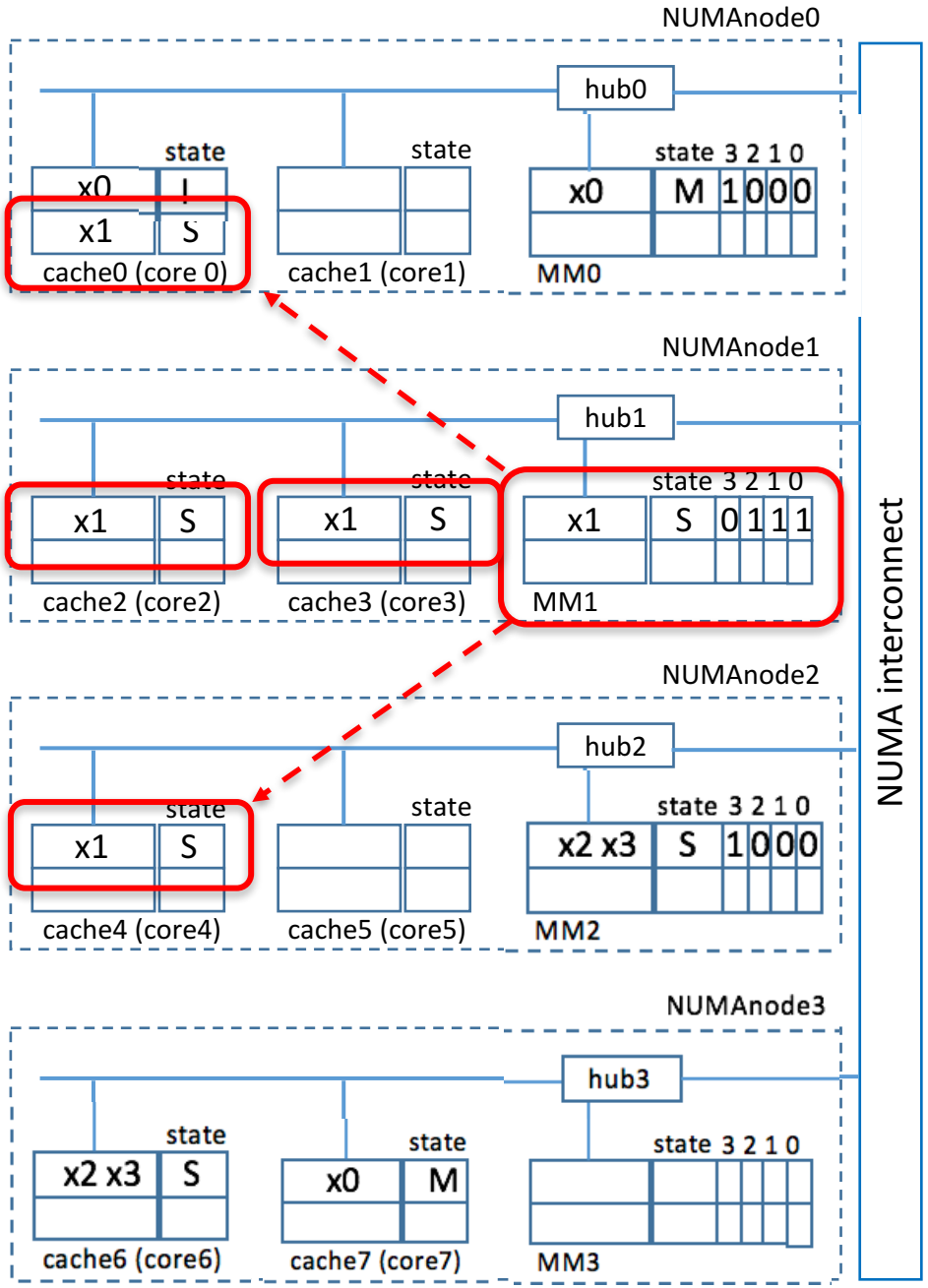
Final state of x0:

Snoopy information:
x0 old copy in cache of core0 is invalidated after flushing value to memory
x0 loaded a copy from MM to cache of core7 with M state

Directory information:
X0 keep M state but with presence bit on NUMAnode3 now.



(d) core 2 writes x1. Indicate final state in memories.

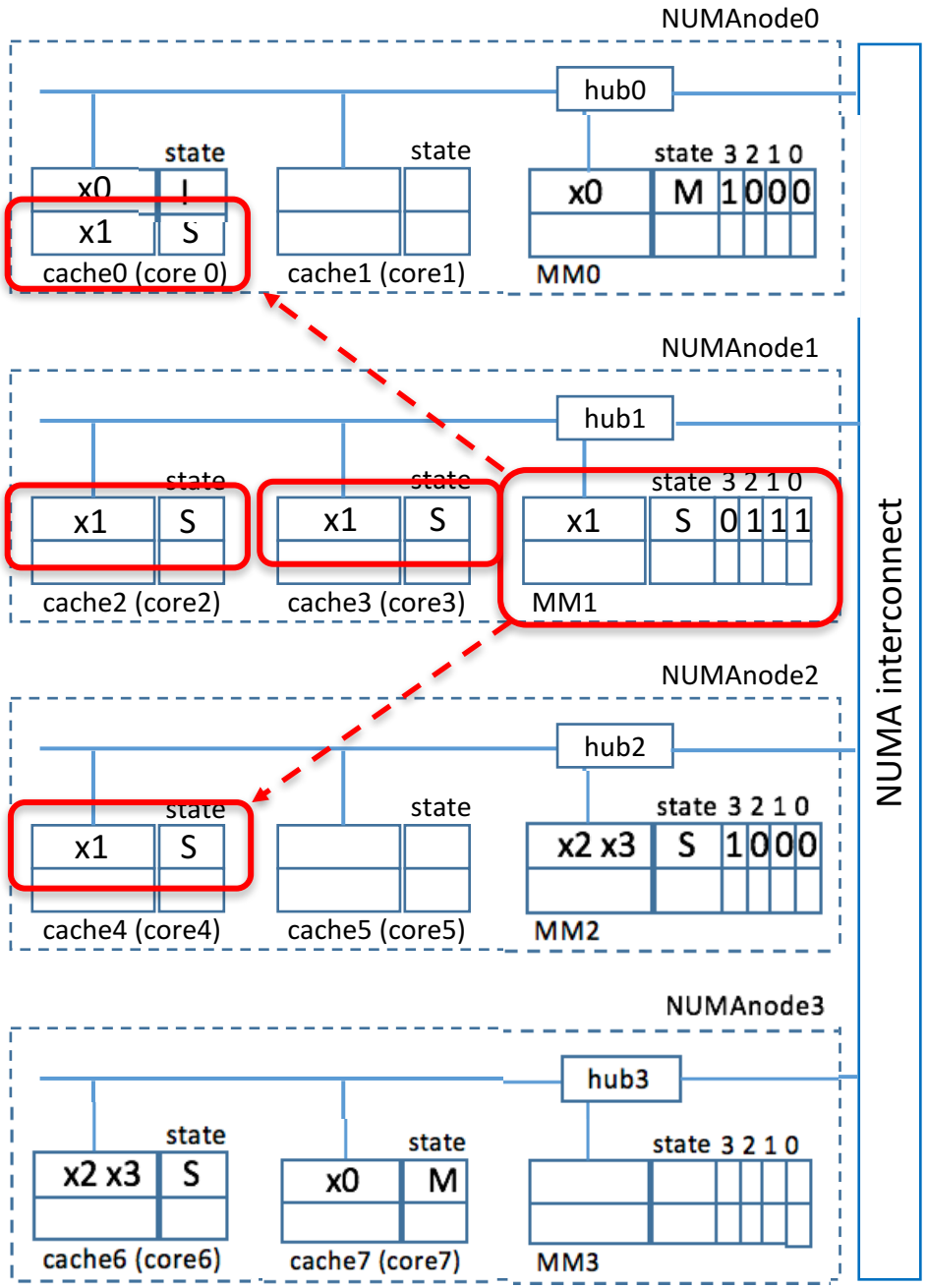


(d) core 2 writes x1. Indicate final state in memories.

Current state of x1:

Directory information:
x1 has a copy in some cache of NUMA nodes 2, 1 and 0 with state Shared.

Snoopy information:
Cache line with x1 has Shared state in caches of core0, core2, core3 and core4
→ Hit in cache of core2



(d) core 2 writes x1. Indicate final state in memories.

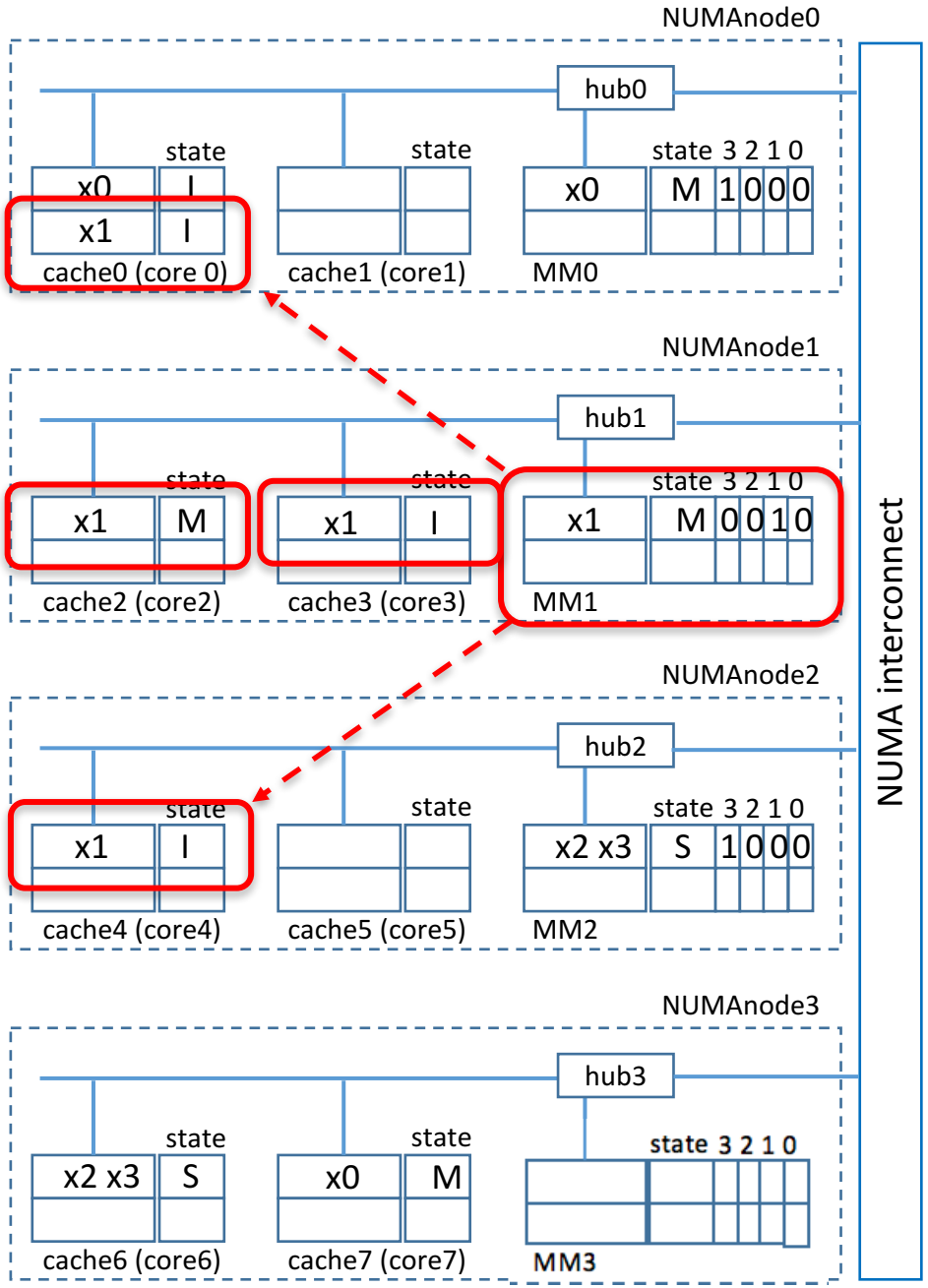
Current state of x1:

Directory information:
x1 has a copy in some cache of NUMA nodes 2, 1 and 0 with state Shared.

Snoopy information:
Cache line with x1 has Shared state in caches of core0, core2, core3 and core4
→ Hit in cache of core2

Final state of x1:

x1 copies in cache of core 4, cache of core0 and cache of core3 are invalidated
Presence bits are updated: only NUMA node1 is valid with Modified state.
x1 state in cache2 is changed to Modified



(d) core 2 writes x1. Indicate final state in memories.

Current state of x1:

Directory information:
x1 has a copy in some cache of NUMA nodes 2, 1 and 0 with state Shared.

Snoopy information:
Cache line with x1 has Shared state in caches of core0, core2, core3 and core4
→ Hit in cache of core2

Final state of x1:

x1 copies in cache of core 4, cache of core0 and cache of core3 are invalidated
Presence bits are updated: only NUMA node1 is valid with Modified state.
x1 state in cache2 is changed to Modified