CACHE COHERENCE SIMULATION

TEAM MEMBERS:

BN: 24

DIAA ELDIN HASSAN

BN: 35

KAREEM MOHAMED

BN: 41

MARINA BEDEER

BN: 65

NOURAN ALAA

BN: 67

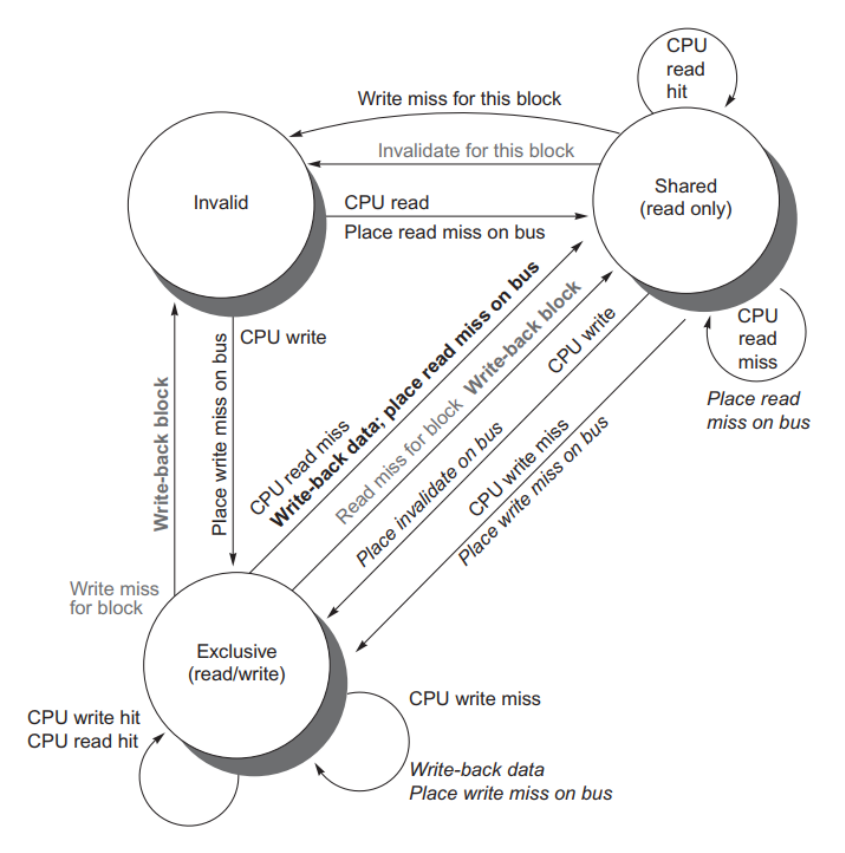
HALA ALAA ELDIN

MSI PROTOCOL

In this repository we are presenting the Cache Coherence MSI Protocol to analyze how would cache perform under various workloads.

We have studied about different snooping based Cache Coherence Protocols in class. Whenever a processor wants to read or write something, it tries to use its own cache to avoid having to go to the memory each time (as it's very slow). But, when we have multiple processors, we need to synchronize the caches, so that all processors have a coherent view of memory. For this, one approach is to use a snooping cache, where each cache monitors the memory reads and writes done by other caches and takes some action based on those requests. MSI is one simple choice but there are other protocols too which offer different kinds of benefits under specific workloads - MESI, MOSI, MOESI, etc.

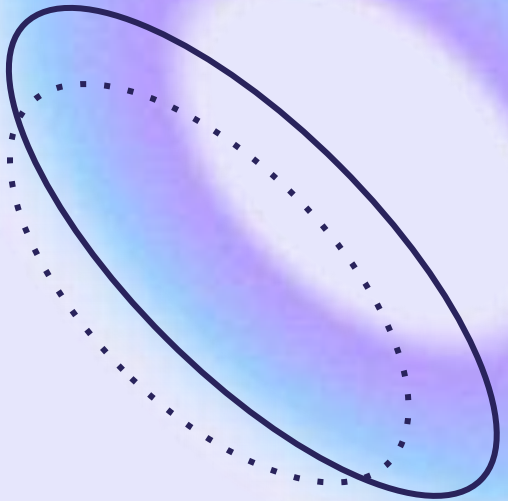
To simulate the output we are using SMPCache simulator which comes with a pre-configured memory traces and different protocols configurations to be used for the simulation.



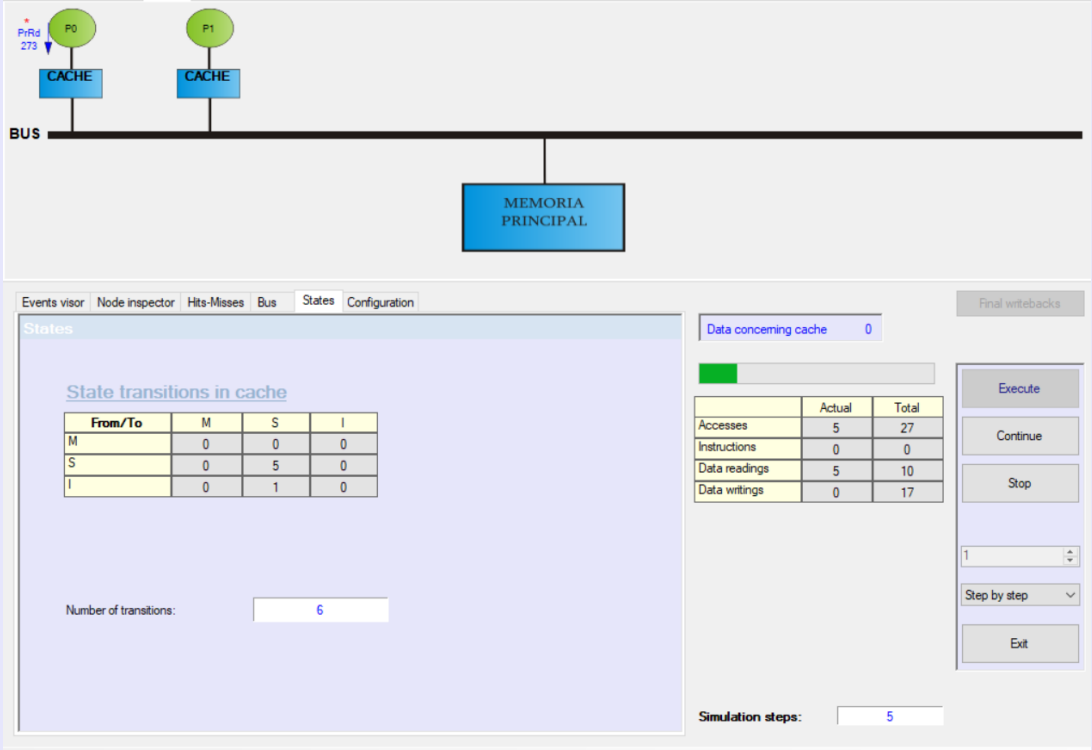
The following is the State Transition Diagram for MSI Protocol

SIMULATION RESULTS

ACCESS NUMBERS: 5&6
ACCESS NUMBER: 12&13
ACCESS NUMBER: 17&18



ACCESS NUMBER 5:



Node P[0] Seek block -> 273

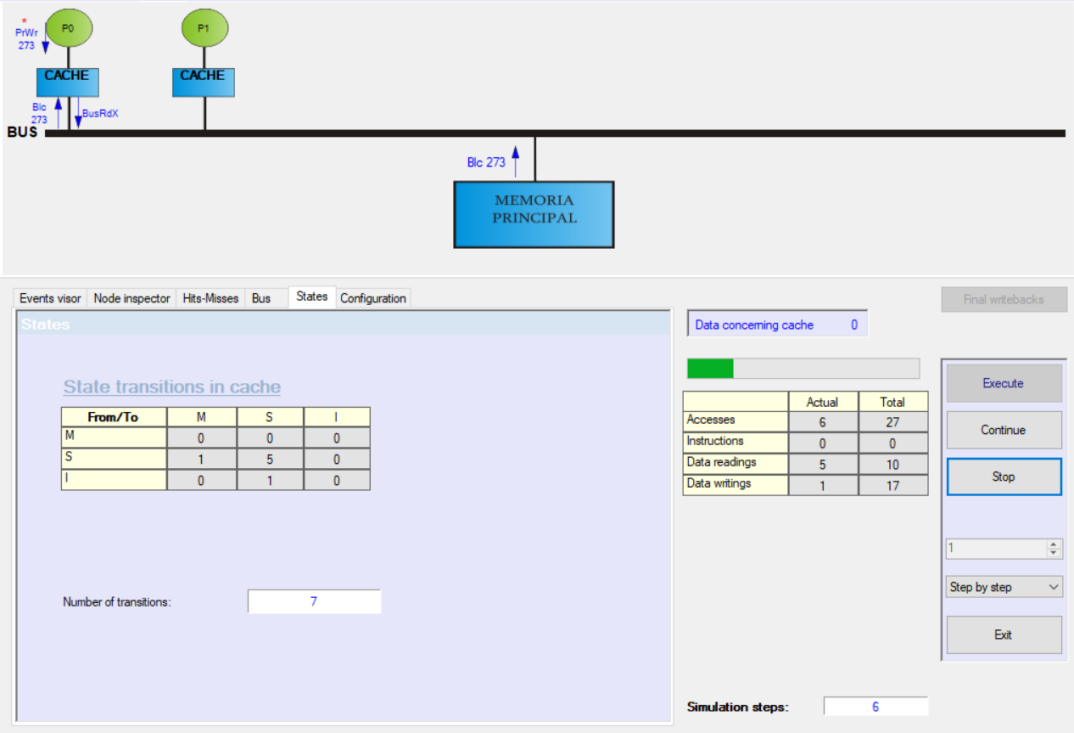
1.- Reading request (PrRd)

2.- Hit in cache

STATE TRANSITION IN CACHE:

SHARED → SHARED

ACCESS NUMBER 6:

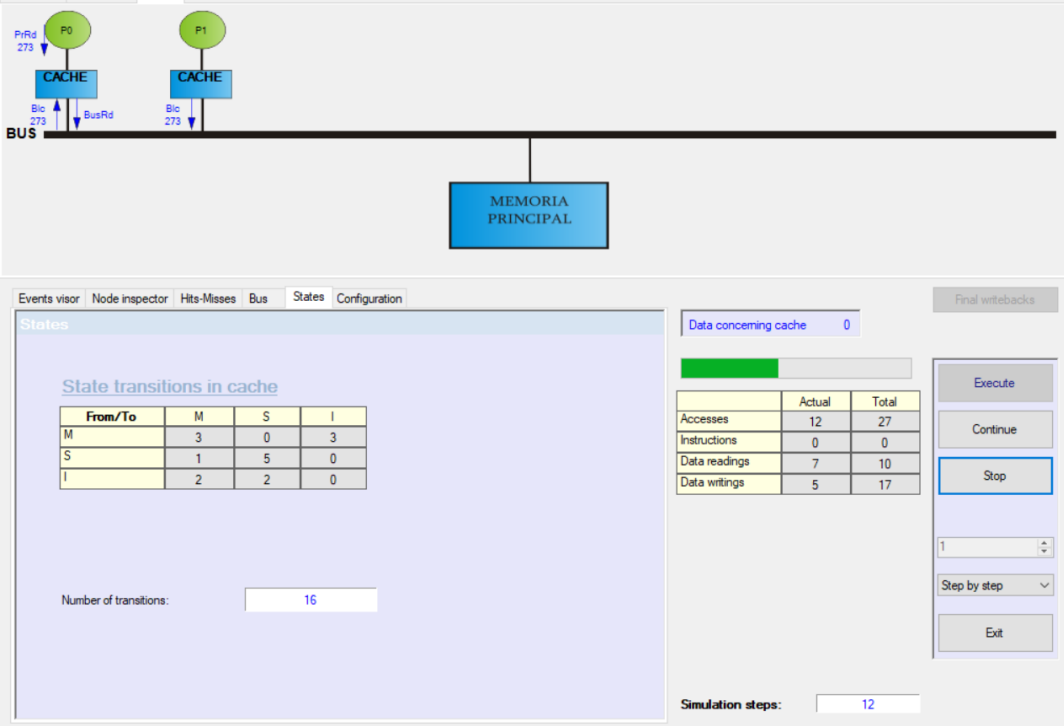


Node P[0] Seek block -> 273
1.- Writing request (PrWr)
2.- Hit in cache
3.- Bus exclusive reading request (BusRdX)
4.- The bus arbiter grants bus to node P0
5.- Transfer of block 273 from main memory

STATE TRANSITION IN CACHE:

SHARED → MODIFIED

ACCESS NUMBER 12:



Node P[0] Seek block -> 273

1.- Reading request (PrRd)

2.- Miss in cache

3.- Bus reading request (BusRd)

4.- The bus arbiter grants bus to node P0

5.- Transfer of block 273 from node P1

6.- Writeback in node P1 block 273

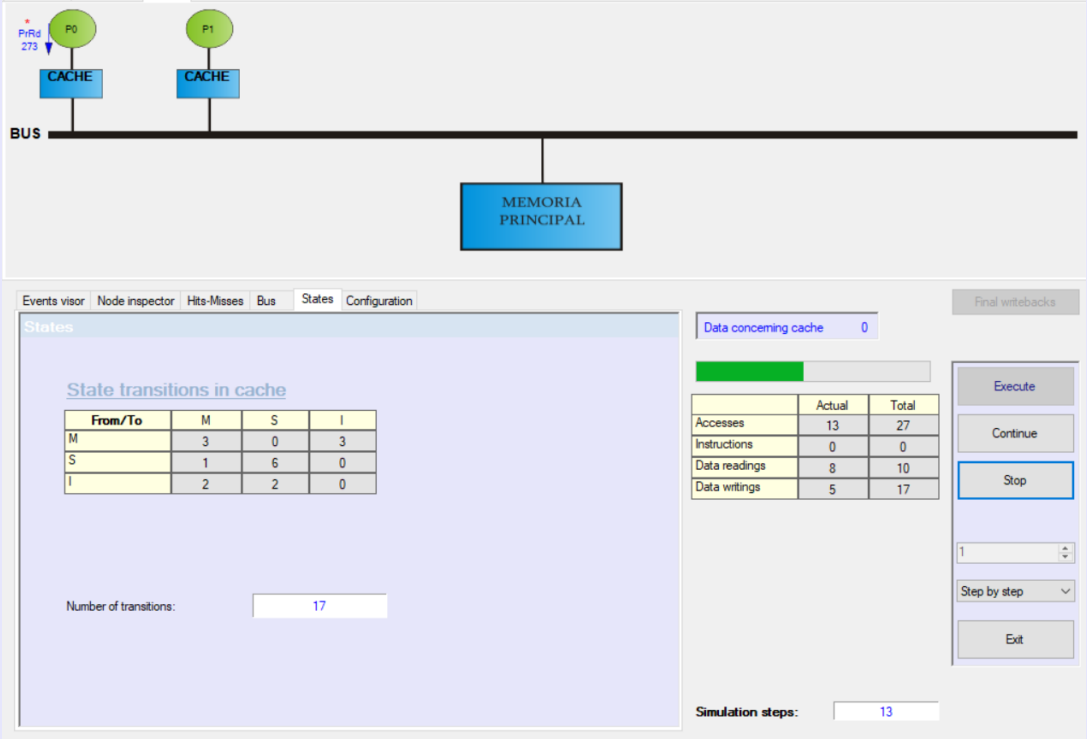
6a.- Writeback request (BusWb)

6b.- Transfer of block 273 to main memory

STATE TRANSITION IN CACHE:

INVALID → SHARED

ACCESS NUMBER 13:

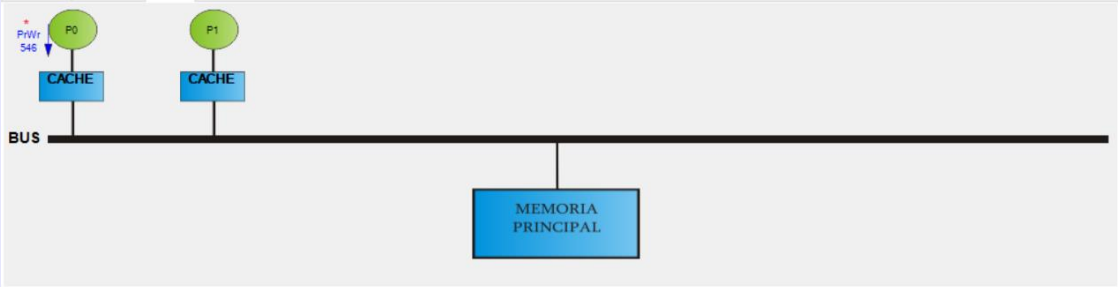


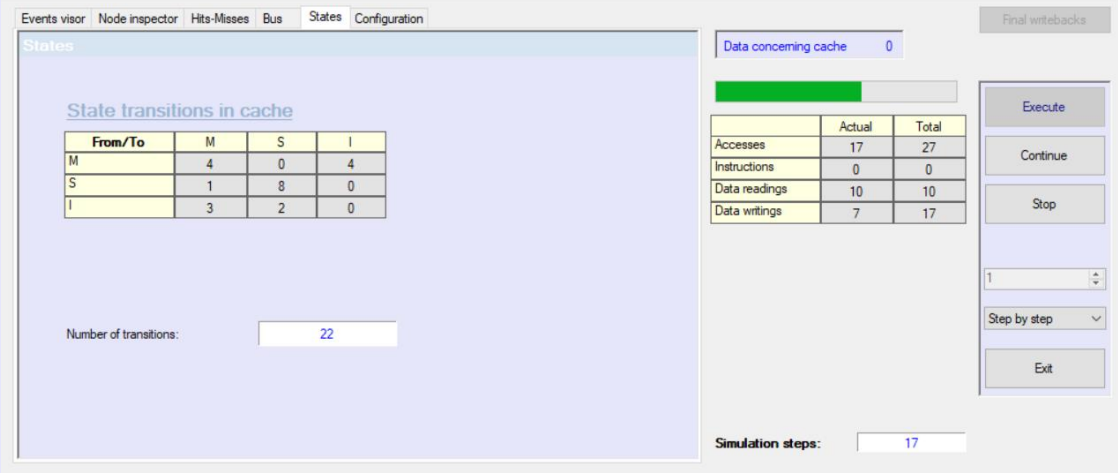
Node P[0] Seek block -> 273
1.- Reading request (PrRd)
2.- Hit in cache

STATE TRANSITION IN CACHE:

SHARED → SHARED

ACCESS NUMBER 17:



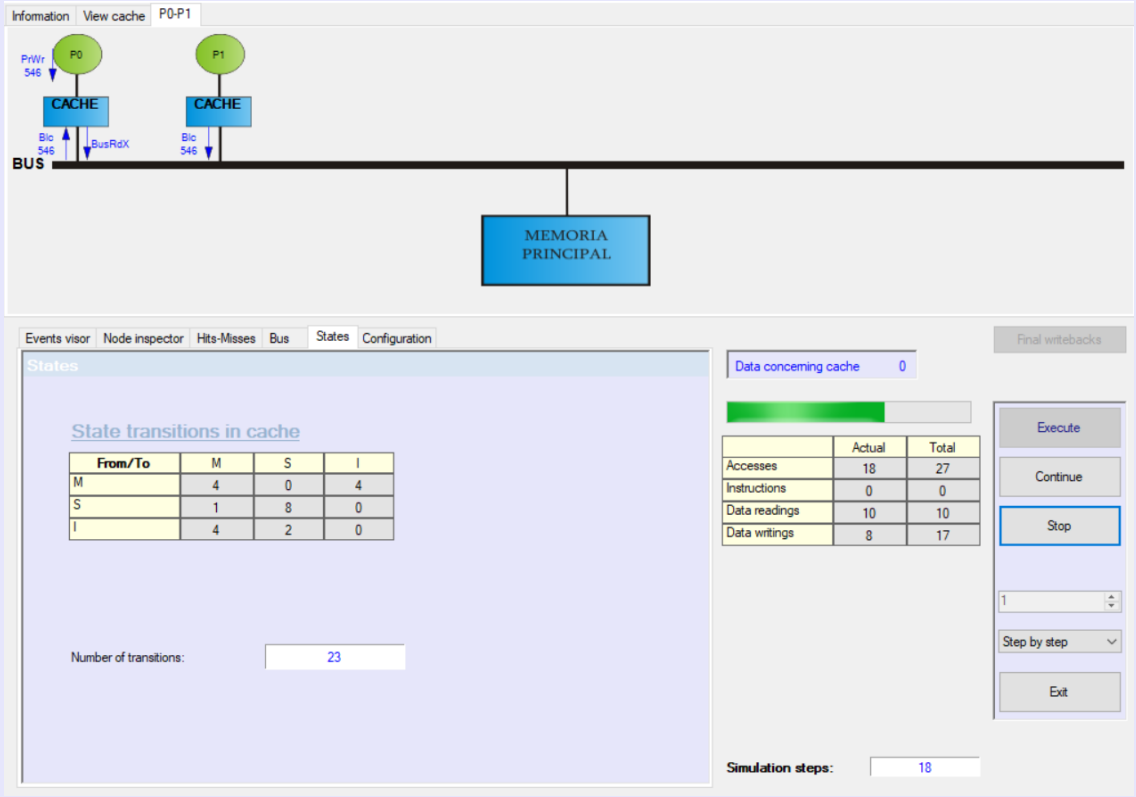


Node P[0] Seek block -> 546
1.- Writing request (PrWr)
2.- Hit in cache

STATE TRANSITION IN CACHE:

MODIFIED → MODIFIED
AFTER MODIFICATION OF P[1] ON B.546
MODIFIED → INVALID

ACCESS NUMBER 18:



Node P[0] Seek block -> 546
1.- Writing request (PrWr)
2.- Miss in cache
3.- Bus exclusive reading request (BusRdX)
4.- The bus arbiter grants bus to node P0
5.- Transfer of block 546 from node P1

STATE TRANSITION IN CACHE:

INVALID → MODIFIED

THANK YOU

https://github.com/Nouran-Alaa/CacheCoherence_CA