Cache coherence protocols solve the problem of data inconsistency that arise from updating the same memory address by individual caches of multiple cores. Cache coherence protocols ensure the updated data version is propagated in memory for multi-core systems. Cache coherence thus ensures that changes in all operations are propagated in a timely manner across the system. The consistency of stored and shared data is called cache coherency.

The design modules are: 1) CPU: There are two CPUs in this project. These design units essentially demand peruse or keep in touch with the cache. 2) Cache: There are two Caches in the logic. The caches are two-way associative. There are 8 entries in the cache, and each cache section has 11 bits, which incorporates information such as data, tag, update, dirty and valid bits. These modules are considered in detail in the accompanying sections. 3) Memory-Mapping Unit: This unit is for changing the virtual location over to the actual location. The design incorporates the virtual location line as 7 bits and the actual location line is 5 bits. This is done essentially by cutting the main two bits of the virtual location. 4) Memory Bus Controller: Because we are utilizing various modules which can get to the transport simultaneously, the memory transport regulator is required. 5) Memory: The memory has 32 entries; every transaction of data within cache and moesi states.

The main methodology in this project was to implement the additional state “Owned” in the state machine. In case if the cache controller could never see a miss on the read operation on the bus, at the time of snooping, then FSM will never go to owned state. The module includes a FSM which would transfer all caches between these varying states: Modified, Exclusive, Shared, Invalid. The design was built using Verilog HDL, and verified on the testbench.

Following are the metrics using which performance can be evaluated.

- Snooping Bus Transactions - The number of bus transactions that the protocol requires for the program.
- Memory Requests - The number of cache line loads that have to be served by main memory (and not another cache).
- Memory Write-Backs - The number of times any cache line has to be written back to main memory.
- Cache to Cache Transfers - The number of cache line loads that are served by another cache.

The image below shows the functionality of the two-way caching logic. Here, initially the cache storages are filled with zeros. Thus, the “tag” bit which is compared across this data would not equate and there will be read miss. We can observe in the image below that CPU A is requesting for address 0011101, whereas CPU b is requesting for address 1110011. Since cache has just data and address =0’s, read miss takes place, and bus requests are sent out. Since we used simple arbitration, bus grants are given in order of requests. After the grants, data is taken into cache from main memory and leads to read hit then.

As seen from the simulations above, the state flow of MOESI Protocol works properly for all the read-write operations within any multiprocessor architecture. The table below shows how this MOESI protocol is advantageous over the earlier version: MESI protocol in context to writing operations. In case of reading operations, both yield similar results but indeed solve the problem of cache coherency. These results were obtained after performing a huge number of read-write operations through the testbench.

<table>
<thead>
<tr>
<th></th>
<th>Read accuracy</th>
<th>Write accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>MESI</td>
<td>4865</td>
<td>4760</td>
</tr>
<tr>
<td>MOESI</td>
<td>45912</td>
<td>5670</td>
</tr>
</tbody>
</table>

Acknowledgements

We would like to express our sincere gratitude to Prof. Lili He for constantly guiding us throughout the project, and for giving timely suggestions which helped us in successfully completing the project. We would also like to thank Prof. Morris Jones for providing us with necessary inputs regarding the design and also pointing us towards the efficient approach. Furthermore, we would like to thank the Electrical Department at San Jose State University for providing us with valuable resources.