Protocol Description: The protocol is a request/acknowledgement style message passing. Any request made transitions to an intermediate state where the corresponding ACK is expected.

Read Request: A processor sends a read request, and the home responds back with a ReadAck if that line was in a non-exclusive state. If the line was in an exclusive state then the home sends a DownReq to the processor who has it with the auxiliary information as the source processor. The other processor receives the DownReq and sends a ReadAck to the original request processor, downgrades to a shared state and responds back to home.

Write Requests/Upgrade Requests: A WriteReq sends Invalidation requests to all the other processors which are in shared or exclusive state for that data. After receiving responds from all of them, home sends a WriteAck back to the original processor.

Writeback Requests: A WBReq is generated whenever the processor wants to voluntarily invalidate itself. The home responds back with a WBAck. However, WBReq processing involves a lot more things. WBReq serves as a InvResp or DownResp whenever the processor is expecting a response. This is simply because the WBReq serves the same purpose as the other two i.e. downgrades the current state. On the processor side, A InvReq or DownReq servers as a WBResp also for the same reason as above. Therefore, again every message has been appropriately acknowledged by some means.

An interesting case occurs for DownReq serving as WBResp. This happens when home sent a DownReq to the processor but it sent a WBReq before it received a DownReq. This WBReq has already invalidated the local data, which means it can no longer serve the original ReadReq which triggered this DownReq. So in this case, when processor receives a DownReq as an ack for WBReq, it sends a spoofed ReadReq to home. Now WBReq has a higher priority and will get processed before this ReadReq. Then after that, the ReadReq is served as a normal read request by the home.

