Bing Li, 1004191910
Tapasvi Patel, 1004176606

1.0 Pre Lab Answers

1. In real hardware, state transitions and their associated actions do not happen instantly, transient states offer a time period where the controller can safely process the request in case another request comes in during that transition period. Without transient states, when you are switching from one state to another, this operation does not happen atomic or automatically. It takes some time to switch from one state to another. Within this time, there could be another operation that happens that may affect one of these state values. Therefore, it is better to have transient states so that as a cache block is transitioning and another operation that may affect it occurs, it will stall.
2. If a request comes in during a transient state, it needs to be stalled to be process later after the transition is complete - the controller can't allow the request to go through or it may mess up the single-writer, multiple-reader invariant or the data values, and the request can't just be discarded because it could cause the computation of incorrect values.
3. A deadlock is when no progress can be made in a system due to 2 inter-dependent events (e.g. A must complete before B occurs, but B depends on A completing). This can be avoided through the use of virtual networks – networks with their own set of input/output buffers for different types of requests.
4. Put-Ack messages send an acknowledgement to a cache controller, to indicate that a PUT message has been received.
5. The sender of a data reply is determined by who currently owns the data, the possible options are:
    a. DATA_FROM_OWNER: data from a remote owner (L1 cache)
    b. DATA_FROM_DIR: data from the directory
    c. DATA: data from a cache going to a directory
6. A Put-Ack is sent as a response to a PUT message, and an Inv-Ack is sent as a response to an INV message. Put-Ack occurs when a successful Put has happened and the data has been written to the directory. Inv-Ack occurs when a request has not been successful due to a number of reasons (ex. Data is currently in a transient state and cannot be modified).

2.0 Section 5 Answers

1. The FSM will wait in a transient state until it receives the Inv-Ack, after which it will move to the Invalid state.
2. A PUTM could come from a NonOwner source. This could happen if the NonOwner core wants to write data to a cache line but a cache miss occurs. First, the NonOwner core requests a write to a particular memory location. A cache miss occurs. Then, the NonOwner core will request ownership of that address from the directory controller by issuing a PUTM message.
3. We need to differentiate in case there are two cores that are writing to a data block at the same time. Therefore, the message requests have to be stalled and thus, they will have different request types.

Bing Li, 1004191910
Tapasvi Patel, 1004176606

4. It is possible to receive a PUTS Last request because you can have multiple cores writing a message to a data block, therefore, you have to stall each request when the block is transitioning.
5. A cache block in a Modified state will be the only existing instance of it outside of memory, and therefore would never be sending an Invalidation request. The request would be sent from the same core if it was in a Shared state, or another core if that core needs to have the cache block in a modified state
6. This is not possible because if a block in a cache is in the process of transitioning from the shared state to the invalid state, this could have only happened if the directory was also given notice that this block will no longer be shared. Therefore, this block is first removed from the shared list in the directory, then it is transitioned from shared to invalied. Therefore, if another core sends a Fwd-GetS request to the directory, the directory no longer has the current cache in it's shared list, thus it will not forward the message to it.
7. We were not able to finish the lab. However, from a high level point of view, we went through all the different combinations of states and state transitions that were possible and all the events that could happen in each state. We drew a state transition diagram like the one given in lecture, and our logic made sense.

3.0 Protocol Modifications

Cache Controller

Transient State Added	**Description**
IS_D	I->S wait for data
SM_BI	S->M broadcast a bus read, invalidate in other proc
MS_W	M->S write to dir, then change to share
MI_W	M->I write to mem/send data, then invalidate in curr proc, wait for WB_ACK to go to I
IM_BW	I->M send bus write, the write data

Directory Controller

Transient State Added	**Description**
SM_BI	S->M, invalidate sharers, wait for ACK
MS_R	M->S, get modified value, send data

4.0 Statement of Work

Bing Li, 1004191910
Tapasvi Patel, 1004176606

Bing: report, code (50%)
Tapasvi: report, code (50%)