Quinn Magendanz

6.823 Lab 3

1. A description of your protocol.

Summary: My protocol relied on a system of requests and acknowledgements to maintain consistency between the directory and the processors. The while the processor is in any of its stable states (MSI), it can initiate a transition request using the defined rules. The directory is prompted by these transition requests and shifts to ensure processors’ requests are provided while maintaining cache coherence. A series of transitionary states are used in both the directory and processors. A processor waits in a transitionary state while it waits for the directory to either grant it additional permissions or to writeback data. The directory waits in a transitionary state while it waits for processors to acknowledge the change in state it wants to make, and downgrade/invalidate in order to make it happen.

Message Traffic: I adjusted the NetMax for my protocol to be 2\*(N-1) + 1 for my protocol since there can be at most 2\*(N-1) messages queued in the HomeNode at once (and the number of messages must always be less than NetMax). This max number can happen when all processors are sharers, one requests exlusive access, and the directory issues (N-1) invalidations, but at the same time the other (N-1) processors also each issue exclusive requests. This results in 2(N-1) messages.

Coherence Challenges: The most difficult challenge was when there were simultaneous invalidation requests sent by the directory and writeback requests sent by the processor. This resulted in the processor completing the action desired by the directory, but without receiving the invalidation request. If not careful, this could result in an extra invalidation request or writeback acknowledgement floating around. To solve this, I ensured that in the directory’s transitionary states, I was careful to not issue writeback acknowledgements to a writeback request if an invalidation request was already sent out.

3-Hop Implementation: The changes involved in the 3-hop implementation were focused around downgrading an owner while in the directory’s exclusive state. This involved passing the share-requesting processor into the aux field of the messages while issuing a downgrade request to the owner. When the owner finished downgrading, it would send both the downgrade acknowledgement to the directory and the share acknowledgement to the requesting processor.

1. State transition diagram documenting the complete state machine for your protocol.

Key:

SR - ShareRequest

SA - ShareAck

ER - ExclusiveRequest

EA - ExclusiveAck

WR - WritebackRequest  
WA - WritebackAck

IR - InvalidateRequest

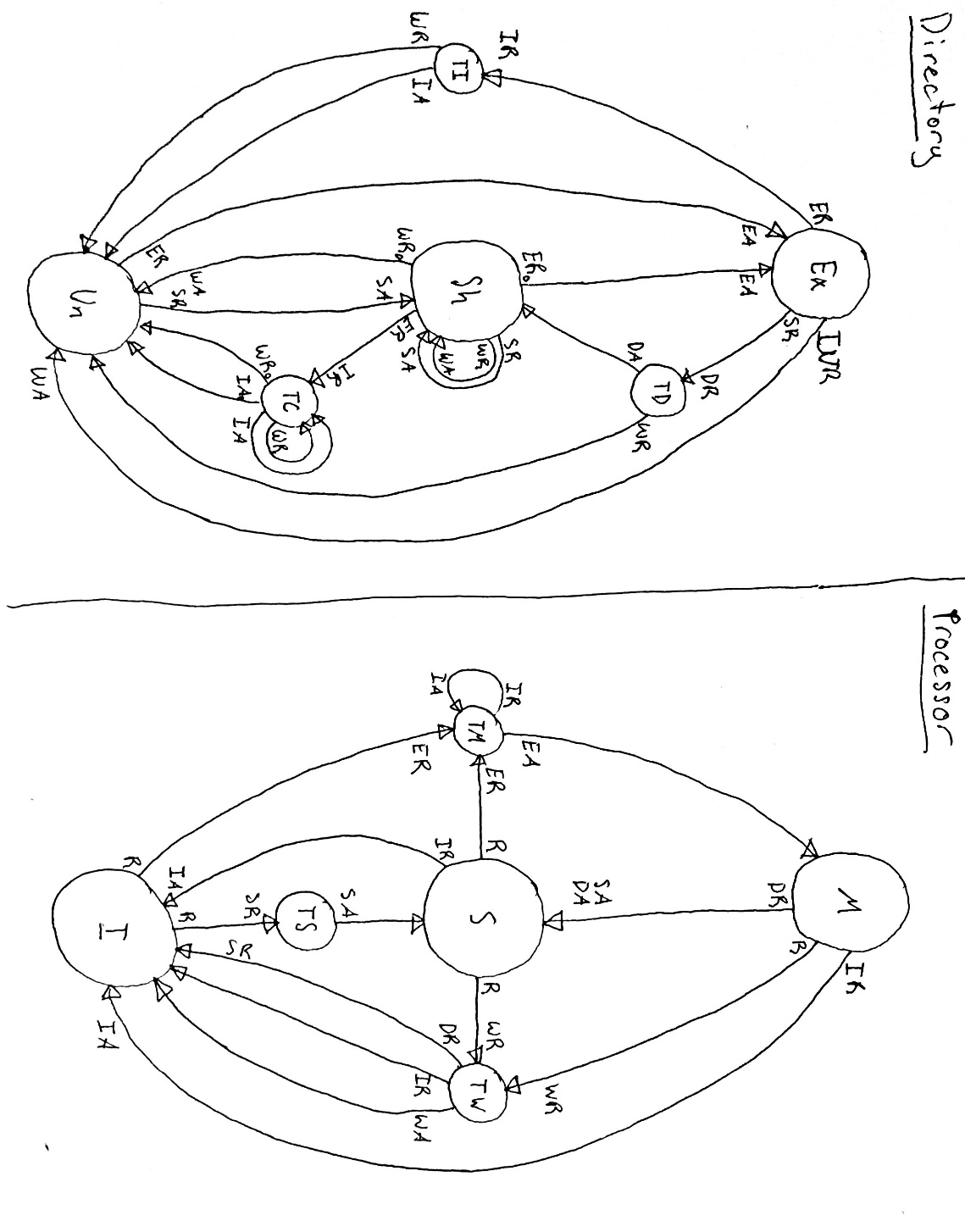
IA - InvalidateAck

DR - DowngradeRequest

DA - DowngradeAck

R - Processor rule

Each arrow starts with the message/action that prompted the transition and ends with the messages sent out during the transaction. A subscript of ‘0’ indicates that the sharer list must also be empty. Stable states are drawn in the large circles while transitionary states are drawn in smaller circles.



SA

1. The output from Murphi showing that no errors were found, the number of states explored and running time.

