15-418: Project Milestone Report

Emily Szabo Sophia Qingyang Cao | Apr 15, 2025

Updated Project Schedule

WEEK 1.0: Start ramp up. (Emily + Sophia)

WEEK 1.5: Finish ramp up. (Emily + Sophia)

WEEK 2.0: Draw state diagram for MSI. Begin state diagrams for MESI, MOESI, and MESIF. (Emily) Plan additional exploration step past correct implementation of MSI, MESI, MOESI, and MESIF. (Sophia) Prepare the Project Milestone Report. (Emily + Sophia)

WEEK 3.0: Implement MSI. (Sophia) Finish diagrams for MESI, MOESI, and MESIF. (Emily)

WEEK 3.5: Implement MESI, MOESI, and MESIF. (Sophia) Draw diagram for HRT-MESI. (Emily)

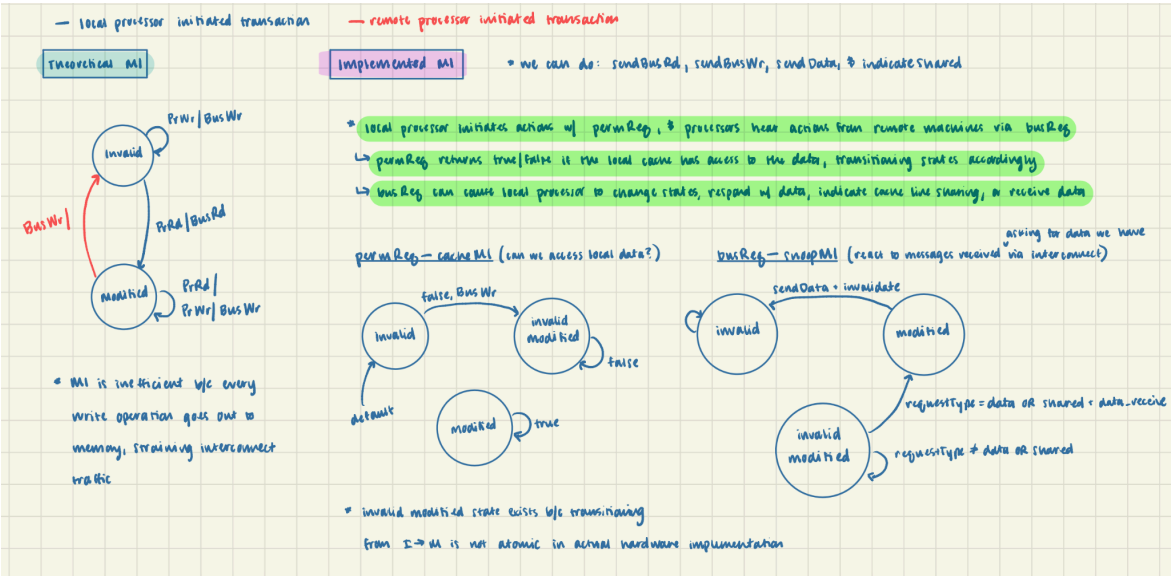
WEEK 4.0: Write trace generators. (Sophia) Implement HRT-MESI. (Emily)

WEEK 4.5: Prepare materials for poster session. (Emily + Sophia)

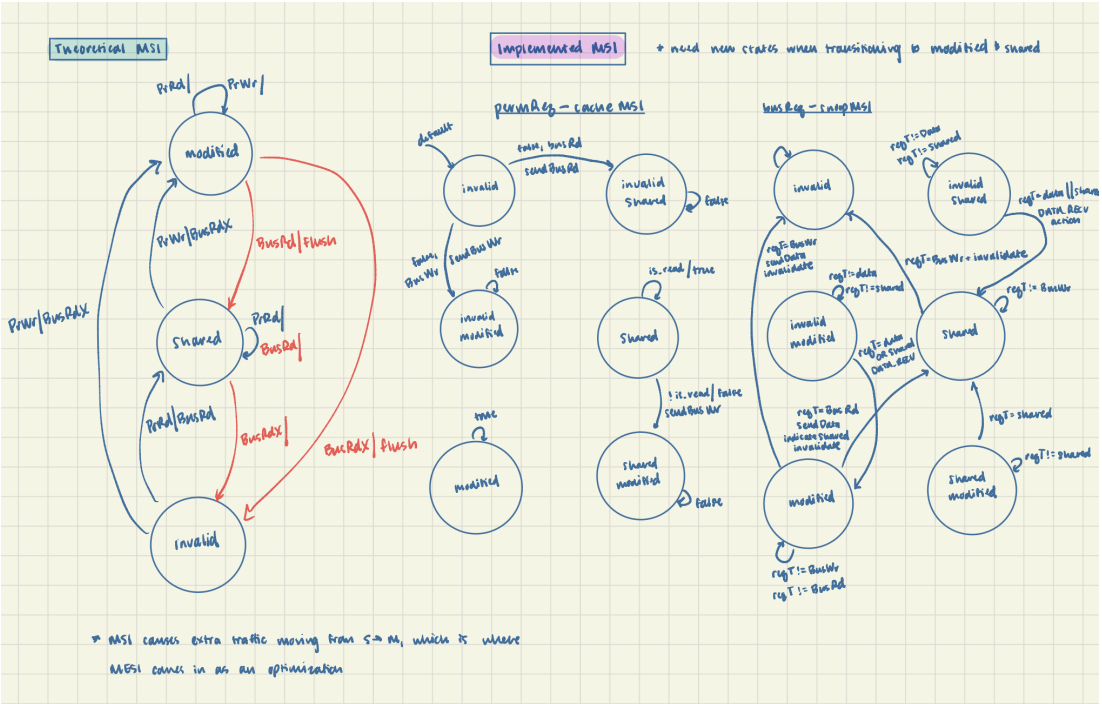
FINALS WEEK: Poster session!

Work Summary So Far

Since the start of the project, we've ramped up on the CADSS codebase from the 15-346 cache coherence lab. We have been particularly focused on understanding the implementation of the MI protocol, especially why it diverges from the textbook two-state version. Through examining how permReg and busReg affect local vs. remote transitions, we identified non-atomic state transition behaviors in real implementations that necessitate intermediate or compound states.



From there, we designed a detailed state machine and transition table for MSI, accounting for local processor requests and remote snooping events. We are in the process of MSI protocol implementation and validation.



Must-Have and Stretch Goal Alignment

We are behind in our project progress because Emily has been sick for the past two weeks and Sophia has been handling PhD travel and decisions; however, we are very close to a completed MSI implementation, so we are only slightly behind from our initial schedule.

Even though we are behind, we are still confident we will be able to produce all of our deliverables because Emily is almost fully functioning and Sophia is wrapping up PhD logistics handling, and we have figured out what needs to be done implementation-wise.

Our goals have changed slightly and are as follows:

- ☐ Necessary
    - ☐ MSI state diagram + implementation.
    - ☐ MESI state diagram + implementation.
    - ☐ MESIF state diagram + implementation.
    - ☐ MOESI state diagram + implementation.
    - ☐ HRT-MESI state diagram + implementation.
- ☐ Nice-Haves
    - ☐ Trace generation. Build a trace generator to generate patterns like high sharing, frequent writes.
    - ☐ Extended protocols, e.g. HRT-ST-MESI

Deliverables at Poster Session

A comparison of protocol behavior and performance via graphs (e.g., coherence traffic, latency, invalidations per access)

Potentially a live demo of the simulator running trace workloads and printing protocol stats in real-time.

Current Issues

Designing correct and deterministic transitions: the things we need to dive deeper into are fully being able to write out the theoretical state transition diagrams for MOESI, MOESIF, and HRT-MESI so we can

transform those into hardware state diagrams (with intermediate states to account for lack of atomic state transitions in the real world).

Available trace characterization: We need to understand the available traces provided by the course 15-346 on afs. (/afs/cs.cmu.edu/academic/class/15346-f22/public/traces/coher/)

Generated Trace validation: if we implement the trace generator, we want to ensure the traces adequately test corner cases and common scenarios of cache.