

Project Title: Robust Cache Coherence Protocol Verification with Inferno

Team Name: FLY-Bee

Team Member: Zeyu Bu, Xiangfei Kong, Chenxi Lou, Yao Jiang

Problem to be addressed:

Fault-tolerant architectures are emerging to guarantee reliable functionality on vulnerable silicon devices. For a fault-tolerant architecture, the RTL design is more complex than a normal one and is more likely to introduce design bugs.

Why does this problem matter?

Fault-tolerant architectures are popular in current circuits design industry. However, the verification of such a complex RTL design is a time consuming and costly task that is likely to become a bottleneck in the release of new architectures. Thus, finding an efficient way for debugging would speed up the verification process.

Idea/Solution to be investigated by the project:

This project will focus on the debugging process of a fault-tolerant directory-based 'MSI' protocol. We will also apply Inferno++ in our debugging process and evaluate this novel tool. To start with, the robust protocol proposed in 'A Systematic Methodology to Develop Resilient Cache Coherence Protocols' can be used to implement the resilient architecture. In the process of implementing the resilient architecture, we expect to run into several bugs. Both traditional debugging methods and Inferno++ will be used to locate bugs and their efficiency will be evaluated and compared in a systematic way. In our case, the efficiency of inferno++ will be evaluated based on its ability of locating bugs. Our analysis should provide a reference for future verification process of complex designs.

Progress so far:

- We have received the core RTL code from Xiaoming & Sijia's group, who did a snoopy bus based dual-core design in their EECS 470 project. The design was claimed to have some bugs in its LSQ unit, which can be a good point for us to investigate after design integration using Inferno. We have separated their single-thread core design from the common bus and the core will later be connected to a 2x2 mesh network within this week.
- We also have downloaded the mesh router RTL model and are currently looking at its RTL code. Two group members (Chenxi & Xiangfei) are responsible for the router - router and router - core connection in this stage and will complete the initial integration before 10/28.
- The other two group members (Zeyu & Yao) are responsible for the directory (memory controller) development as well as the modification to the core's cache controller. They are currently

working on the directory RTL and we are scheduled to assemble two parts together and do the high-level simulation & debugging next week.

- All team members have read the Inferno paper and the Inferno source code has been downloaded. Chenxi and Xiangfei will check if we can use the tool without any license issue in this week. We will report to Doowon if we have any questions/issues with Inferno.
- We have created a google form which will be used to record the debugging process with and without Inferno. We will document detailed date, module related and debugging effort involved in the log for future analysis.

Issues/Showstoppers:

- Inferno
Since we haven't involved the Inferno tool into the project yet, we are not sure at which stage we should bring it in. So far we are considering about using the Inferno tool after initial integration so that the debugging log with and without Inferno assistance can be used as a comparison in the final report.
- Directory-based Design
Xiaoming's EECS 470 project is a snoopy bus based dual core system while for our project we need to move the design into a directory-based one. Only one team member has taken EECS 570 (EECS 470 focused more on a common bus based protocol introduction) so we are not sure how much design effort we will put to change into a directory based protocol.
- Time Conflicts
Two of the team members have on-site interviews scheduled and will fly out of town, which makes it hard for the group to discuss the project together. It is the reason we decided to separate the team into two and we can still work on the project in parallel. Starting from next week, we plan to do the project together.

Comments:

As stated in our project outline, we expect the baseline integration stage will take two or three weeks, which is scheduled to finish around 25th October. However, because of the absence of one of our teammate during the weekend, we would like to delay the integration deadline to 10/28 and start to customize the baseline protocol from then.

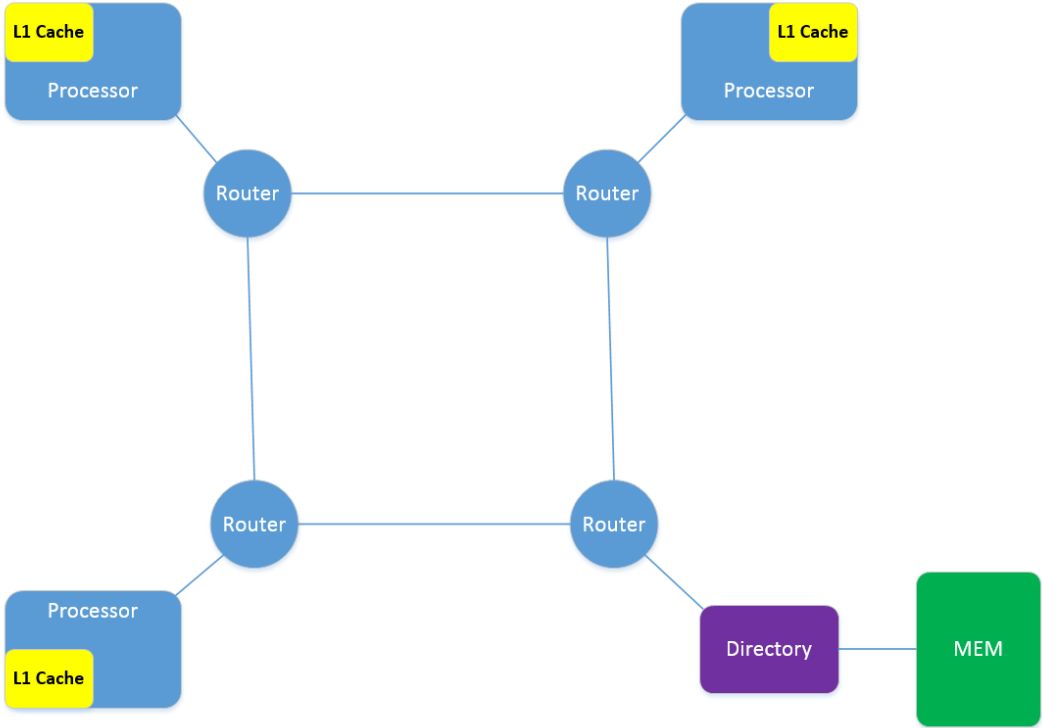


Figure.1. Proposed Multi-core Network for Design