Network Interface Buffer Elimination

Presented by Freelock
Jiabo Li Jiong Xue Qilu Guo Jing Ji

Problem

In a chip multiprocessor, the network interface between core caches and routers needs buffers to store data flits. If the system uses an ack-based protocol, only after the acknowledgement from the other core is received, the value inside the buffer can be invalidated. The handshaking protocol is important for data communication correctness. However, these data buffers in the network interface take up significant silicon area.

Motivaction

Storage of body flits in network interface will cause an area overhead and therefore consume more power. Instead, we could store the data in the cache and eliminate the extra storage in network interface. As the transistor scaling trend goes, low power design becomes more desirable. Thus, our group believes that it is a good idea to save both area and power to remove the extra storage at the network interface.

Proposed Solution

To remove the extra storage at the network interface, we added several cache lines in the processor to store the packet and the network interface will only store the head and tail of the packet. The high-level design is shown in the Figure 1.

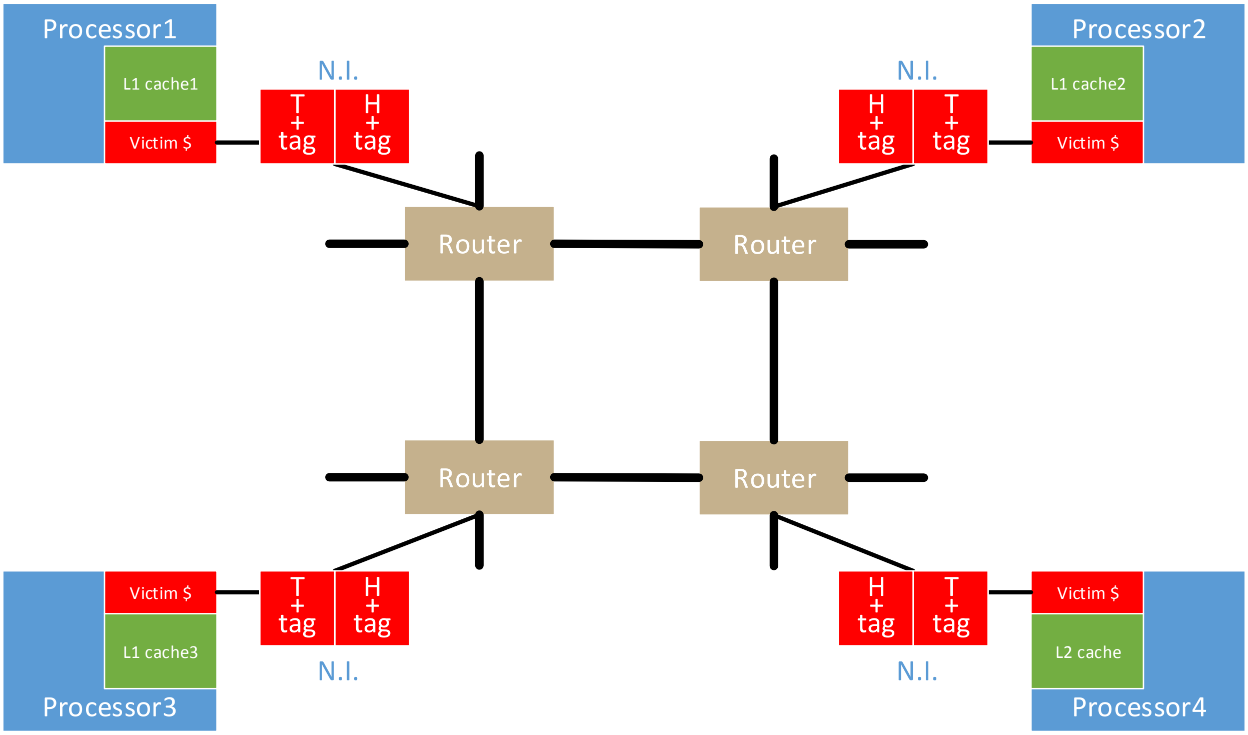

Figure 1. High-level Architecture

The added cache lines will serve as the victim cache to improve the cache hit rate when there are extra lines. However, if there is no line to store packet because lines are used as victim cache, a victim cache line will be removed to give space to the packet.

Developing Plan

- Finish components and baseline design using Verilog
- Complete current design
- Run test cases and do the evaluation on area and performance

Evaluation Plan

Upon finishing our design, we will first evaluate on the total amount of area we have saved comparing with the baseline design of a four-core multiprocessor. The software tools for evaluating storage area as suggested by GSI is CACTI. We will also compare the performance penalty in execution time, which will be mostly caused by the extra time for the packet to stay in the cache. However, we don’t expect to see a large penalty because the packet sent from one core heading to the L2 cache is already in victim cache.

Timeline

Checkpoint 1 (10/23):
By checkpoint 1, the group will complete most of the baseline design and will be working on components including victim cache, network interface, and routers.

Checkpoint 2 (11/13):
By checkpoint 2, the group will complete the design of the 4-core processor for this project and start testing on the correctness of the design. The group will optimize and improve on the design considering area and time penalty.

Checkpoint 3 (12/4):
By checkpoint 3, the group will complete several testing and evaluation mostly on area. All the first-stage optimizations will be finished at this point. The group plan to work on the final paper and present the project to both the professor and GSI for further advice and last-time modifications on group help around that time..

Checkpoint 4 (12/10):
By the last checkpoint, the group will complete a draft final paper and finish up all the testing and optimizations.