Fault-tolerant & Adaptive Stochastic Routing Algorithm for Network-on-Chip

Team CoheVer:
Zixin Wang, Rong Xu, Yang Jiao, Tan Bie

Problem to be addressed

On-chip circuits are vulnerable to errors due to transistor geometric shrinking and performance improvement, leading to serious reliability issues. As a consequence, several network-on-chip (NoC) routing algorithms emerge to address such issues, such as stochastic algorithms. However, stochastic routing algorithms are only intended for dealing with errors in NoC without taking traffic performance into consideration (prone to packet loss). Thus, in our EECS 578 project, we plan to improve the traffic performance of existing stochastic routing without losing its merit in dealing with reliability issues in NoC.

Why does this problem matter?

With intense integration of high performance processing elements on chip, we have a more severe requirement in throughput and latency of NoC, which cannot be achieved by the existing stochastic routing algorithms. Even though it lives up to our requirement for reliability, it's still not a qualified option for current NoC.

Idea & solution to be investigated by the project

There are some options available for the implementation of stochastic routing algorithm, including probabilistic flood, directed flood, random walk, etc. Firstly, we need to analyze the algorithm essence and choose one of them that can be suitably adapted into the NoC system. In order to accommodate adaptive features, the basic stochastic routing algorithm cannot be too much complex, otherwise large area or power overhead will emerge. Thus, at current stage, we prefer to utilize random walk based stochastic algorithm due to its advantage in low complexity and relatively small area overhead.

Even though stochastic routing algorithm gains advantage in fault-tolerance, its inherent defect in performance (due to its oblivious routing strategy) restricts its popularity in NoC system. So we plan to add some adaptive features to our implementation of stochastic routing algorithm for high performance. Specifically speaking, some modules that are responsible for collecting and analyzing real-time data of NoC will be added, by comparing probability of each route, the optimal route can be determined. Of course, adaptive features cannot do harm to fault-tolerance, some evaluations will be conducted not only for the performance improvement, but for its reliability.

In all, we plan to develop a fault-tolerant & adaptive stochastic routing algorithm for NoC in our project.

Plans to develop the project

The project is based on BookSim[1] Interconnection Network Simulator.

(1) Design and implement the stochastic routing algorithm based on random walk in BookSim. (Specifically, adding random-walk-based routing function in 'routefunc.cpp' within BookSim)

(2) Add reliability measurement functions to BookSim and verify the proposed algorithm in a transient / permanent faulty environment.

(3) Add adaptive features such as self-learning routing table to the stochastic algorithm. Thus, the system could utilize acknowledge signal from destination to adjust the real-time probability of each route and further take advantage of such probability data to determine routers' output port. In this way, stochastic routing algorithm becomes adaptive to real-time status and run-time errors.

(4) Simulation process: compare the original stochastic routing algorithm with our adaptive one from several aspects, including performance (overall latency), fault tolerance, power overhead, etc.

Project evaluation

Firstly, BookSim equips us with measurement in latency and throughput, which can be applied to the result evaluation. Besides, BookSim provides us a set of 32nm ITRS prediction module for power consumption, we can take advantage of such tools to evaluate the power overhead of our adaptive features. At last, as there is no existing module to conduct reliability evaluation, we plan to add some reliability measurement functions within the source code for reliability simulation and comparison.

Timeline

Checkpoint 1 (10/23/2015):
Design and realize the stochastic routing algorithm in BookSim.

Checkpoint 2 (11/13/2015):
Add reliability measurement modules to BookSim. Start adding adaptive features to the stochastic algorithm.

Checkpoint 3 (12/04/2015):
Finish adding adaptive features to stochastic algorithm. Complete evaluation process in performance, power and realiability.

Checkpoint 4 (12/10/2015):
Final optimization of the project.

[1] BookSim: A cycle-accurate interconnection network simulator. *Nan Jiang, Daniel U. Becker, George Michelogiannakis, James Balfour, Brian Towles, John Kim and William J. Dally. A Detailed and Flexible Cycle-Accurate Network-on-Chip Simulator. In Proceedings of the 2013 IEEE International Symposium on Performance Analysis of Systems and Software, 2013.*