

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

Team CoheVer:

Zixin Wang, Rong Xu, Yang Jiao, Tan Bie

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 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 drawback 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, 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.

Progress so far:

1. We have designed a basic random walk routing algorithm in Booksim.
2. We inserted a fault table into Booksim. This fault table helps us introduce failure links in the network.
3. Based on the random walk algorithm that we designed and the dimension-order algorithm which the Booksim has, we build our current algorithm. This algorithm will use XY dimension order algorithm firstly and when it comes across a link failure, the algorithm will switch to random walk to bypass the fault.

Issues/showstoppers:

This algorithm is still incomplete. As you can see, we haven't added the timeout counter so our algorithm hasn't been able to clear those timeout flits. Actually, we tried to add this feature into our algorithm, but because we are not very skilled with Booksim, there is always something wrong when we tried to do it.