NNSE: Nostrum Network-on-Chip Simulation Environment

Zhonghai Lu, Rikard Thid, Mikael Millberg, Erland Nilsson and Axel Jantsch
Royal Institute of Technology, Stockholm, Sweden

Abstract: *A main challenge for Network-on-chip design is to select a network architecture that suits a particular application. NNSE allows the user to analyze the performance impact of NoC configuration parameters. It allows one to (1) configure a network with respect to topology, flow control and routing algorithm etc.; (2) configure various regular and application specific traffic patterns; (3) evaluate the network with the traffic patterns in terms of latency and throughput.*

Description:
Network-on-Chip (NoC) design faces many challenges coming from the huge architectural design space and time-to-market/time-in-market pressures. One major difficulty is to select a communication network that suits a specific application. Currently no public simulation tool exists to aid NoC designers to make the decision. Our configuration and simulation tool NNSE aims to fill this gap.

1. *Simulation kernel*

The tool logically comprises a NoC simulation kernel wrapped with a graphical user interface (GUI). The kernel provides a layered network simulation engine. Following ISO's OSI model, the kernel implements five layers, namely, the physical layer, the data link layer, the network layer, the transport layer and the application layer. The transport layer provides transaction-level communication primitives such as read() and write() to enable communication via channels between application processes. Each layer may be configured with a set of parameters that represents its characteristics. The simulation tool currently supports to configure the network layer and the application layer. The GUI provides a convenient way to specify parameters for the network and traffic configuration.

2. *Network configuration*

A network is characterized by topology, flow control scheme and routing algorithm etc. Each of them has a large design space on its own. With the tool, all of these characteristics are parameterized. So far the tool realizes only a limited set of configurations, which are as follows:

- Topology: 2D mesh and 2D torus.
- Routing/Switching: Wormhole routing and deflection routing. For wormhole routing, one can choose virtual channel (VC) parameters like the number and depth of VCs. For deflection routing, one can specify the base routing algorithm and deflection policy.

Other routing and switching techniques will be integrated into NNSE in the near future.

3. *Traffic configuration*
One important aspect in NoC design is that the chosen network should be customized for applications, i.e. application-specific. In addition to customize the network parameters, the network should be evaluated extensively with various regular and application-specific traffic patterns since the network selection is an architectural decision and thus should be relatively stable.

The regular traffic patterns consist of uniform and locality traffic. The uniform traffic is distributed over the network nodes uniformly. With locality traffic, one can specify the locality index which controls the communication probability between nodes at different distances. The application-specific traffic is based on per channel and can be used to configure application-specific, irregular traffic. One can specify explicitly the communication characteristics of each channel. The former is dynamic while the latter is static. With both types of traffic patterns, one can specify message temporal behavior and message sizes.

4. *Performance evaluation*
After configuring a network and a traffic pattern, one can evaluate the network with the traffic pattern. The evaluation is based on the kernel simulation results shown as figures. The main performance measures are latency and throughput. Typical figures include average latency vs. offered traffic, throughput vs. offered traffic etc.

With the GUI, all the network and traffic configurations can be stored and reused. To facilitate data exchange, they are stored as XML files.