NIRGAM: A Simulator for NoC Interconnect Routing and Applications' Modeling[1]

M.S.Gaur[+],B.M.Al-Hashimi[*], V.Laxmi[+], Navaneeth R[+], Naveen Choudhary[+] ,
Lavina Jain[*], Mushtaq Ahmed[+], K. K. Paliwal[+], Varsha[+], Rekha[+], Vineetha[+]

Recent research indicates that it will become difficult, for the bus-based interconnection techniques to cater to needs of on-chip communication for future System-on-Chip (SoC) as design complexity increases and physical layout approaches its limit. Network-on-Chip (NoC) has emerged as a complementary solution wherein communication network is a set of routers (nodes) connected in a regular or irregular topology. This simplifies design wherein an SoC consists of regularly spaced tiles; each tile consisting of a processing core, a router and interconnection network (channels) connecting these routers.

NoC is described by topology (spatial placement of tiles and intra-tile links), switching technique (how are messages routed from source node to destination) and routing mechanism (router architecture and algorithm to determine actual path from one router to another). Topologies such as mesh, torus, butterflies, etc. and different switching mechanisms such as wormhole and virtual cut-through have been proposed. To reduce buffering capacity needed at any node, packets are divided into flits. The switching objective is to utilize channels with minimum possible buffer space. Buffering at any point can lead to blocking resources (channel and/or node buffers) and may lead to deadlock situations. Virtual-channel router architecture lets a packet progress bypassing a blocked resource by providing multiple buffers (virtual channels) instead of a single buffer at each port of NoC. Various routing algorithms including XY, odd-even[1], and more recently CAIS[2], DyAD, have been proposed. In the context of NoC, latency and throughput are considered to be important performance measures. To consider multiple options available at every stage of NoC design: topology, switching, buffer size, number of virtual channels, routing arbitration, routing algorithm and router architecture; a cycle-accurate simulator which can output performance metric(s) of a given set of choices is much needed.

Our proposal describes an extensible and modular SystemC based simulator (NIRGAM), which let the user plug-in and experiment with different applications and routing algorithms. It allows the user to analyse the performance (currently latency, throughput) of a NoC design for a user specified application and a user specified routing algorithm. At present, NIRGAM (NoC Interconnect RoutinG and Applications' Modeling) simulator supports mesh, torus, mesh with link failures and irregular topologies with wormhole switching mechanism. Users can select one of the in-built routing algorithms: deterministic XY, source, deadlock free odd-even (OE), Q-routing, minimally adaptive XY, DyAd, PROM, table-based routing, intelligent Slack Time Aware Routing (iSTAR) and fault-tolerant routing. Quality of Service (QoS) is supported where the user can reserve a specified amount of band-width for Guaranteed Throughput (GT) traffic. ORION power model has been integrated to calculate router power consumption for any specified number of clock cycles. We allow the users to specify which of the available applications is to be attached to which core in the topology. The available applications are: source (sender) application, sink (reciever) application and synthetic traffic generators (constant bit rate (CBR), bursty, input trace based traffic). Supported traffic patterns include: butterfly, bit-reversal, bit-shuffle and bit-transpose. We also facilitate users to include their own applications and attach them to the cores.

To the best of our knowledge, most of existing simulators[3,4] are specific to NoC topology architecture and may not be able to provide extension flexibility and modularity for traffic, algorithms, applications, and functionality for existing and novel architectures. NIRGAM is aimed primarily at the NoC research community wherein it provides researchers with convenient and

1This work is funded by EPSRC (UK) grant EP/C512804/1 and is greatly acknowledged.
[*] School of Electronics and Computer Science, University of Southampton, UK, [+] Department of Computer Engineering, Malaviya National Institute of Technology, Jaipur, India.

efficient mechanism to experiment with NoC design in terms of routing algorithms and applications on various topologies. Users can easily plug-in their own routers and applications. The simulator is capable of dynamically loading a router and attaching any user-specified application library to any core. A number of demonstrations showing the various capabilities of NIRGAM will be available during the poster session.

1. Sending CBR traffic from a core to a specified or randomly chosen destination core.
2. Sending Bursty traffic from a core to a specified or randomly chosed destination core.
3. Sending Bursty and CBR traffic after reserving a chosen band-width for Bursty traffic.
4. Sending Bursty and CBR traffic in a mesh topology with link failures.
5. Router power estimation for any specified number of clock cycles.

Each of the above applications can be simulated for one of the chosen routing algorithms: XY, OE, fault-tolerant routing, minimally adaptive XY and PROM . It is expected that the workshop will be the appropriate setting to NIRGAM in seeking input from the research community on the simulator current capabilities and its future development. An illustrative flow diagram of NIRGAM is depicted in Figure 1.

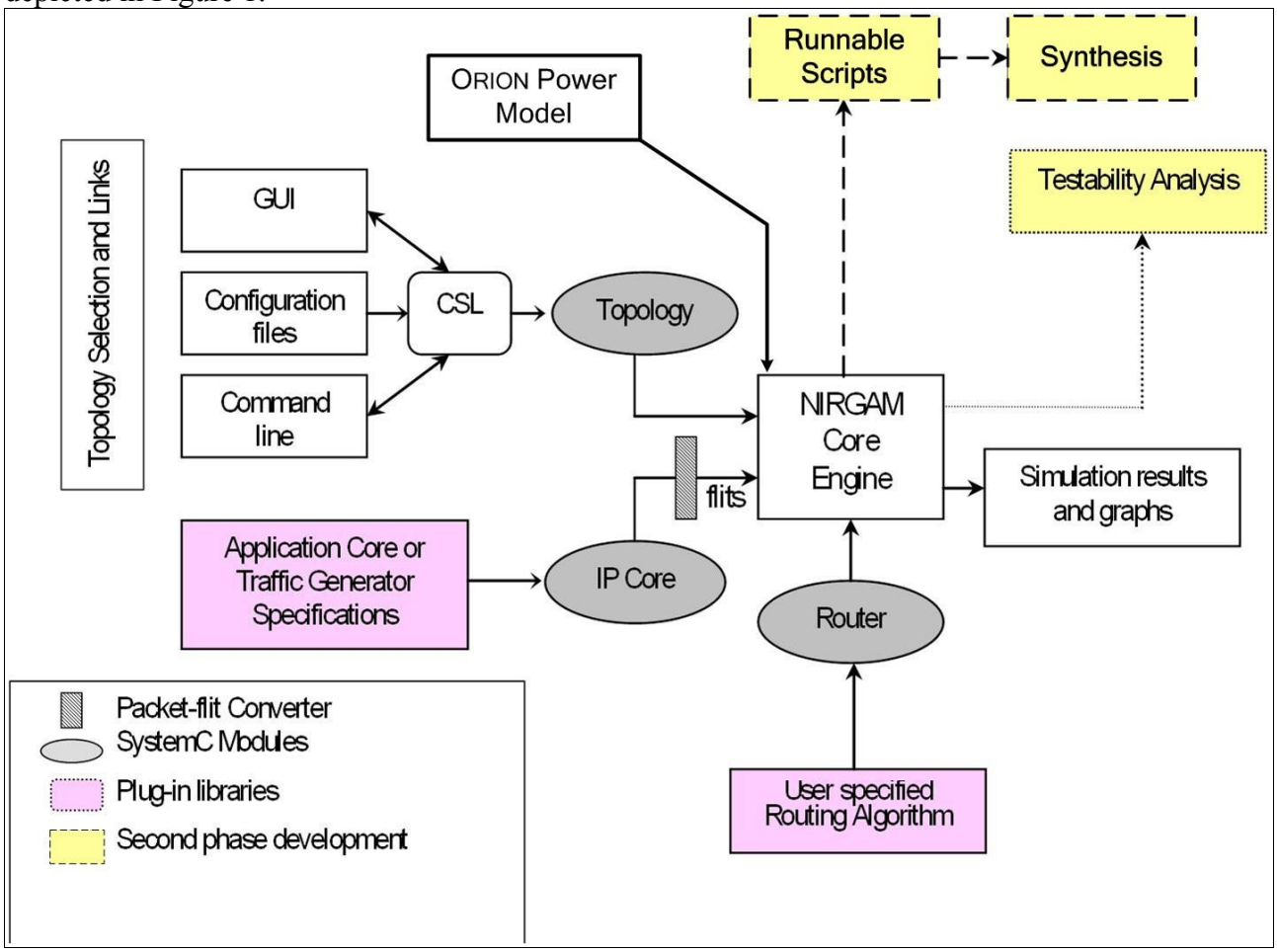

Figure 1: NIRGAM simulator

References:

[1] G.-M.Chiu. The Odd-Even Turn Model for Adaptive Routing. *IEEE transactions on Parallel and Distributed Systems*. 11(7), pp. 729-738, 2000.

[2] D. Wu, Bashir M. Al-Hashimi, Marcus T. Schmitz. Improving Routing Efficiency for Network-on-Chip through Contention-Aware Input Selection. *Proceedings of Asia South Pacific Design Automation Conference (ASPDAC'06)*. Jan. 24-27, Yokohama, Japan, pp. 36-41, (2006).

[3] NNSE: Nostrum Network Simulation Environment. *http://www.imit.kth.se/*

[4] TuD: The MANGO clock less network-on-chip: Concepts and implementation. *http://www2.imm.dtu.dk/*