A SystemC based Simulation Platform for Network-on-Chip Architectures

Thanh-Vu Le-Van, Dien-Tap Ngo, Xuan-Tu Tran

SIS Laboratory, VNU University of Engineering and Technology, Vietnam National University, Hanoi

144 Xuan Thuy road, Cau Giay district, Hanoi 10000, Vietnam. Email: vulvt@husc.edu.vn

Abstract – In recent years, the NoC architecture was developed with many approaches to solution communication request of complex SoCs. The study of NoC paradigm is complex and long time, the researchers have described all component of NoC and SoC and then they have been simulated and synthesis by dedicated tools. By using SystemC, we propose a NoC platform for simulation and performance evaluation of network. In the platform, we design router architecture with five ports to support 2D mesh topology and variable network sizes. This work supports flexible for simulation with many communication pattern. In the simulation, we use shell bash to automatic configuration network and communication pattern. Based on work's result, the platform described NoC paradigm with efficient and faster time.

Keyword NoC, Platform, SystemC

1. Introduction

The NoC paradigm has been considered an effective solution for communication infrastructure in MPSoCs. The NoC was provided higher throughput, higher scalability, lower power consumption and easy for increasing more functions [4][6].

Because difference. With distributed communication, NoC is a new paradigm on communication on chip, so on it should be presented and evaluated in detail. Some works developed NoC architecture at system level design to simulation [5] or trade-off performance and cost [8]. To study NoC architecture, we focus on NoC architecture and simulation in full platform.

In this paper the context is organized as follow: the section II present related works about architecture NoC and simulation of NoC. The proposed platform was described in section III. In the section 4, we execute simulation the platform in Linux environment and some results. The conclusion was repented in section V.

2. Related works

There are many works about NoC, works approach with difference orientations and levels. Some works used VHDL or Verilog to development NoC architecture [4][6][10]. In the architecture, the router was proposed to forwarding data with high performance and lower cost or trade-off between features. However, some works studied NoC at system level design [2][5][9] by using SystemC.

In [5], Chai and et al present a NoC platform for simulation and verification with two parameters: latency and throughput. The platform supports five traffic pattern and changing depth of buffer in router.

Thonnart et al have developed a fully framework using GALS NoC [10]. In this framework, the NoC was proposed with full components: routers, NI and protocols. The result of works was synthesized and fabricated with CMOS 65nm technology.

On the other hand, performance network was studied through some works [1] or [8]. Using test wrapper in Design for Test (DfT), NoC and router was tested by an IEEE 1500-compliant test wrapper [1]. The wrapper was built in cix cells and comparator, which are compliant with the IEEE 1500 standard. Talwar and Amrutur explored microarchitecture of NoC by using SystemC at system level design [8]. In the works, the NoC was configured as 2D array with three topologies: mesh, torus and folded torus. When the framework was simulated, the traffic consists of both Request-Response messages and One-way messages to evaluating latency and throughput of NoC.

With the NoC paradigm, the flexibility SoCs was able to exchange optimal data. But NoC is complex knowledge and need many tool for designing and implement. Maximum page of each paper is 6. If your paper is less than 6 pages, you may choose 2 or 4 pages. If your paper has odd page numbers, insert a blank page at the last.

3. Proposed architecture

NoC is communication infrastructure in SoCs, it receives data from IPcores and exchanging data to sink IPcores. In complex SoC, IPcores were connected NoC through Network Adaptor, and router have architecture which were described Figure 1.

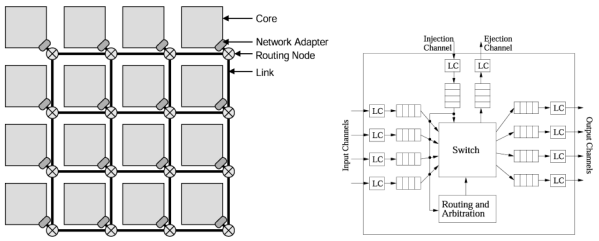

Figure 1. A Network-on-Chip architecture [4]

In paper we focus on simulation NoC for performance evaluation of NoC. So on, we propose the platform, which has a NoC 2D mesh, $m \times n$ IPcores and the wrapper for simulation in SystemC. The NoC is made up of $m \times n$ router with 2D mesh topology with bidirectional links and two virtual channels. So that, the router has five INPORT blocks; five OUTPORT blocks and two crossbars. For simulation, we created many NoC with different network size, and then using the input parameter in simulation instruction to choosing a NoC. And for simulation, we proposed IPcore, which was directly connected router through LOCAL port. All works were represented in .

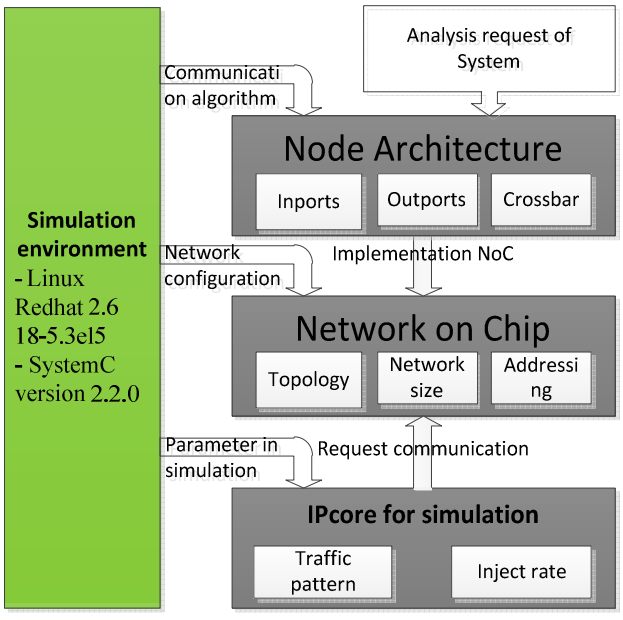

Figure 2. Platform for simulation

The NoC has array architecture, so that it support DOR routing algorithm, such as: XY, WF, NL or NF. The router was addressed with two axis: X axis and Y axis, and the IPcore was connect with router which has a same address.

3.1. The router architecture

The router is key element of NoC, it implement exchanging data between IPcores through routing and forwarding. It received data flit at INPORT block, and then routing and switching data to the corresponding OUTPORT. The OUTPORT process data forwarding, it decides which INPORT was allowed forwarding. The CROSSBAR is two matrix signals, it is used connection INPORT and OUTPORT for two virtual channel. Therefore, the router was design with blocks in Figure 3. By using block design, we need build basic blocks, and then we synthesis blocks into component.

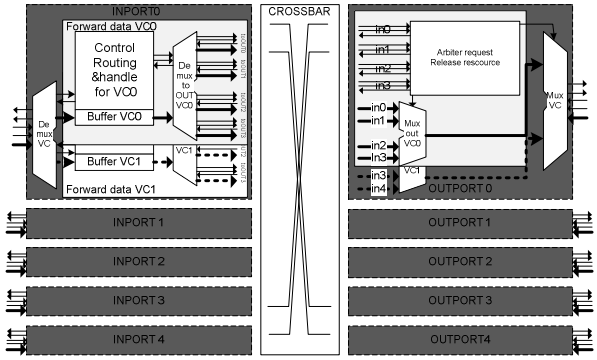

Figure 3. Detail of router in NoC

3.2. Micro architecture of INPORT block

INPORT block has functions: receiving data flit, buffering and routing data to corresponding OUTPORT. Because we used credit-based flow control, the INPORT have two handshake signals to the previous router: send signal and accept signal for a virtual channel. The proposed architecture of INPORT was represented in Figure 4. This architecture uses two least bits in PTT field of header flit to routing data (Figure 6), so that the IPcores must create routing information before injecting header flit of data packets into network when the platform was simulated.

The *VC_Demux* sub-block was used receiving data and control the others sub-block of INPORT. The *sc_fifo* primitive was used to implementation buffer in our routers, so that we only read data from buffer after checking the ready of OUTPORT. At *Pro_routeVCx*[1], when *enablex* signal active, it request to OUTPORT through *Enablex_toOUTy*[2] signal. The INPORT block

[1] *x*:The number of virtual channel

[2] *y*: The number of port: 0,1,2,3 or 4

will forward data only when the OUTPORT block accepts by actively *Res*x*_frOUT*y.

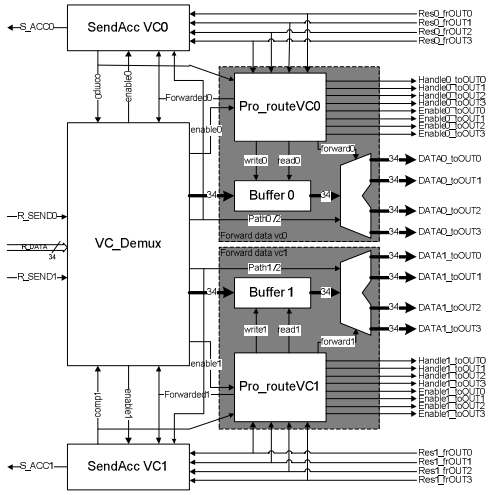

Figure 4. Architecture of INPORT

The handshake signals between routers/IPcores are SEND and ACC signals (**Error! Reference source not found.** and **Error! Reference source not found.**). At INPORT block, *SendAcc VC*x sub-blocks permit ACC as soon as the OUTPORT response for request of INPORT.

3.3. Micro architecture of OUTPORT block

The important of OUTPORT block is arbitration requests form INPORT blocks. When having the request form INPORT block, the sub-block *Pro_ArbiterVC*x combines state of next router and current state of OUTPORT to response INPORT.

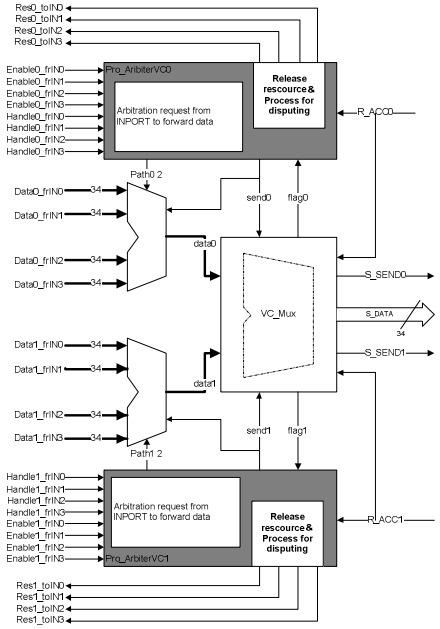

Figure 5. Sub-blocks in OUTPORT module

If the next router cannot receive data, the *Pro_ArbiterVC*x disable all response signals.

If current state of OUTPORT is responding INPORT*y*, *Pro_ArbiterVC*x sub-block disable all response of other INPORT.

If OUTPORT is idle state and next router is ready, the OUTPORT block is active response signal to corresponding INPORT.

After accepting INPORT, the OUTPORT receives data from INPORT and switch data to next router in the same cycle of clock. The *Pro_ArbiterVC*x control multiplex to collecting data of INPORT blocks.

4. Simulation platform

4.1. Configuration NoC

The platform has a NoC, which was made from $m \times n$ routers on 2D mesh topology. The network was addressed by two axes: X axis and Y axis. So that, each router has a coordination (x,y), and the IPcore was connected router also uses the same coordination.

In this work, we have two network sizes: 2×2 and 4×4. With corner router or gap router, the interface of port was connected the wrapper for simulation. When the platform was simulated, the network configuration was setup through input parameters. The input parameters are used to determine the routing algorithm, network size.

The IPcore was designed to injecting data into network through LOCAL port of router. The platform supports difference communication pattern by changing simulation parameter of IPcore, such as: load (used in per cent), packet size and time for simulation.

4.2. Executing simulation

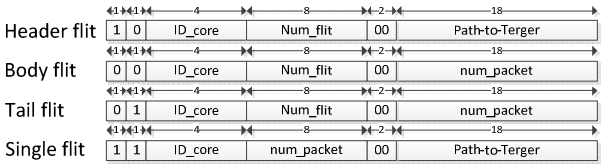

Figure 6. Format of data

The platform was designed by OSCI SystemC version 2.2.0 on Linux 2.6.18-5.3el5. We use input parameters at the command to simulation. First parameter is the network size, the second use to determine routing algorithm. The communication pattern was identified three parameters: the load, packet size and time for simulation.

In the platform, we choice source routing method to

creating routing information at sink IPcore. If packet has one flit, the IPcore creates data flit which has five parts such as Figure 6. The *num_packet* field is number of packet, it is used to compare the data flit was stored at sink. In other case, a packet has a header flit, body flit and tail flit which are shown in Figure 6.

Table 1. Characteristics of environment for simulation

Features	Description
HDL	OSCI SystemC 2.2.0
OS	Linux 2.6.18-5.3el5
RAM	1024Mbyte
CPU	Dual Core E2160 1.8GHz

The environment of this work was shown at Table 1.

In Linux OS, we use shell bash to write a script for automatic simulation the platform. The script call proposed platform in loops with set of parameters follow simulation strategies. So that, while the simulation was automatic executed, nobody cannot change the simulation parameters. The result of simulation is ANSI file, which are able to use in MS EXCEL, Origin or other software to plot and data analysis.

Table 2. Parameters of simulation

Parameters	***Value***
Network configuration	4×4 2D mesh Credit-based, wormhole
Routing	Source – XY routing
Communication pattern	Uniform, complement
Range of packet size	1-256 flit/packet
Number of packet	219510
Number of flit	12462290
Packet error	0
Time for simulation	240508 seconds

5. Results and conclusion

The platform was simulated in shell bash with file script, it have list of input parameters. We use the loop increment the network load and packet size, in the loop the script proposed NoC was simulated with a network load, packet size and time for simulation. So that all the simulation was executed automatic and the result is the final data file in ANSI format. In the result data file contains latency parameter and throughput corresponding network load and packet size.

The results of simulation was shown in Table 2~~Table 2~~. Using trace_vcd file, we take the waveform of signals in router and network to validate operation of proposed architecture. From the waveforms show at the maximum throughput, each cycle clock has a transferred flit at link between two routers.

The proposed platform support simulation and performance in SystemC and C++, by using input parameter we could configurable flexibility network configurable and communication pattern. Simulation was executed automatically so that it was faster and reliability. The result is data file in ANSI format, which can be used many tools to analyze and graph plot.

6. References

[1] Alexandre M. Amory, Eduardo Brião, Érika Cota Marcelo Lubaszewski, and Fernando G. Moraes. "A scalable test strategy for Network-on-Chip routers." International Test Conference. 2005.

[2] Antoni Portero, Ramon Pla, Jordi Carrabina. "SystemC Implementation of a NoC." Industrial Technology, 2005. ICIT 2005. IEEE International Conference on. 2005.

[3] Beigné, E.; Clermidy, F.; Vivet, P.; Clouard, A. & Renaudin, M. "An Asynchronous NoC Architecture Providing Low Latency Service and its Multi-level Design Framework." Proceedings of the 11th IEEE International Symposium on Asynchronous Circuits and Systems (ASYNC'05), 2005, 54 - 63

[4] Bjerregaard, Tobias, and Shankar Mahadevan. "A Survey of Research and Practices of Network-on-Chip." ACM Computing Surveys, 38. 2006.

[5] Chai, Song, Chang Wu, Yubai Li, and Zhongming Yang. "A NoC Simulation and Verification Platform Based on SystemC." 2008. 423-426.

[6] Guerrier, Pierre, and Alain Greiner. "A Generic Architecture for On-Chip Packet-Switched Interconnections." Proc. Design, Automation and Test in Europe Conference and Exhibition (DATE). Paris, 2000. 250-256.

[7] Suman K. Mandal, Nikhil Gupta, Ayan Mandal Javier Malave Jason D. Lee, and Rabi N. Mahapatra. "NoCBench: A Benchmarking Platform for Network on Chip." Proceedings of Workshop on Unique Chips and Systems. 2009.

[8] Talwar, Basavaraj, and Bharadwaj Amrutur. "A System-C based Microarchitectural Exploration Framework for Latency, Power and Performance Trade-offs of On-Chip Interconnection Networks." 41st Annual IEEE/ACM International Symposium on Microarchitecture (MICRO-41), 2008.

[9] Tim Kogel, Malte Doerper, Andreas Wieferink Rainer Leupers Gerd Ascheid Heinrich Meyr, and Serge Goossens. "A Modular Simulation Framework for Architectural Exploration of On-Chip Interconnection Networks." Proceedings of the 1st IEEE/ACM/IFIP international conference on Hardware/software codesign and system synthesis. 2003. 7-12.

[10] Yvain Thonnart, Pascal Vivet, Fabien Clermidy. "A Fully Asynchronous Low Power Framework for GALS NoC Integration" DATE '10 Proceedings of the Conference on Design, Automation and Test in Europe. 2010. 33-38.