IJICTR

International Journal of Information
&
Communication Technology Research

ITRC

IJICTR

International Journal of Information
&
Communication Technology Research

ITRC

Volume 9 - Number 2- Spring 2017

Topics of Interest

Information Technology

Information Systems

IT Applications & Services

IT Platforms: Software & Hardware Technology

IT Strategies & Frameworks

Communication Technology

Communication Devices

Communication Theory

Mobile Communications

Optical Communications

Satellite Communications

Signal / Image / Video Processing

Network Technology

Computer & Communication Networks

Wireless Networks

Network Management

Network Security

NGN Technology

Security Management

IJICTR International Journal of Information & Communication Technology Research

ITRC

Volume 9 - Number 2 - Spring 2017

IJICTR International Journal of Information & Communication Technology Research ITRC

Volume 9 - Number 2 - Spring 2017(1-9)

ARTEMIS: A Simulator Tool for Heterogeneous Network-on-Chip

Fatemeh Vardi
Department of Computer Engineering Science and
Research Branch Islamic Azad University
(SRBIAU)
Tehran, Iran
f.vardi@srbiau.ac.ir

Ahmad Khadem-Zadeh
Iran Telecom Research Center
(ITRC)
Tehran, Iran
zadeh@itrc.ac.ir

Midia Reshadi
Department of Computer Engineering Science and
Research Branch Islamic Azad University
(SRBIAU)
Tehran, Iran
reshadi@srbiau.ac.ir

Received: September 24, 2016 - Accepted: February 27, 2017

***Abstract*—Complex homogeneous network-on-chip or heterogeneous network-on-chip increases the need of determining and developing simulation tools for designer to evaluate and comparison network performance. Towards this end, ARTEMIS tool, a matlab based simulator environment is developed. This simulator offers some collections of network configuration regarding to the topology graph, routing algorithm and switching strategy, including allocation scheme for a target application. Consequently, designers can choose the number and depth of virtual channels and the capacity of each link by applying an efficient allocation scheme, which is provided by this tool. Average latency and throughput are evaluation performance metrics that are measured with proposed simulator tool.**

Keywords-interconnection network; homogeneous NoC; heterogeneous NoC; simulator; performance

I. INTRODUCTION

Network on Chip (NoC) is a promising communication paradigm for designing System on Chip (SoC) and Chip Multi Processor (CMP) [1]. In this interconnection network, communication resources such as routers and links have a tremendous impact on performance and power. Furthermore, these resources almost transfer different number of information flows with different data rate for a given application. Since the traffic pattern varies significantly based on the application characteristics, this is obvious that, allocating uniform number of Virtual Channels (VCs) to each router port and also uniform capacity to each link do not bring efficient design in the term of performance and power. Therefore, better handling traffic requirements depend upon to organized and careful assignment of network recourses based on the specific communication pattern of each application [2] to meet design constraints (such as latency and throughput). While poor performance and power due to unacceptable assignment is the result of inefficient design.

Developing simulation tools help designers to make true decisions before implementing the network architecture. To this end, the simulation tools should be capable to explore and assess the network architecture under performance criteria for a given application regard to its traffic characteristic. The ARTEMIS tool with a Graphical User Interface (GUI) written in matlab helps NoC designer to find their solutions. This simulator allows designer to configure and evaluate the homogeneous network or heterogeneous network under various performance parameters with studying different details of network design, such as topology graph, routing algorithm and switching strategy, including allocation scheme. To confirm ARTEMIS, a simple case study imports to this tool. Then, the homogeneous network configuration and heterogeneous network configuration of the synthetic example extracted and the results compared.

The remainder of the paper is as follows: Section II considers previous proposed software simulations. Modeling of NoC in ARTEMIS is discussed in Section III. Then, the flow diagram of the simulator is investigated in Section IV. Section V explains an allocating scheme for discovering the efficient number of VCs and links capacity in the network. The performance evaluation parameters are described in the next Section. Beside, validation of ARTEMIS simulator is provided in Section VI. Finally, Section VII includes the conclusion and future works.

II. Related works

Until now, many papers have developed software simulations with supporting particular design parameters for configuring NoCs [3, 4, 5, 6, 7, 8 and 9]. The cycle-accurate modeling of TOPAZ in [3] is a suitable tool for modeling supercomputer interconnection networks using application traces. In [4] the authors tried to develop a detailed cycle accurate interconnection network model for full system framework. In [5], Jiong presents BookSim for simulating NoCs. This flexible simulator supports many design parameters for configuring large set of networks. NoCTweek [6] and Noxim [7] are another simulators implemented in SystemC. These tools explore the performance and energy efficiency of the NoC. The software simulator presented in [8] targeted at multiprocessor systems. The HNoCs simulator presented in [9] is based on OMNeT++. This tool capable to modeling of heterogeneous NoCs with variable links capacities and number of VCs per unidirectional port. HNOCS provides some performance statistical measurements.

Except HNoCs simulator, all mentioned techniques, configure and evaluate the homogeneous network under various metrics. While, the proposed simulator in this paper offers some collections of both homogeneous network configuration and heterogeneous network configuration regarding to the allocation scheme for defining the number of VCs and links capacity in each router port.

One important notable feature of any simulator is the simulating execution time. Compare to HNoCs, the execution speed of the proposed simulator in this paper is significantly sufficient for rapid network design. Additionally, ARTEMIS tool is driven by actual application traffic traces based on the proposed allocation scheme.

III. Modeling of NoC in ARTEMIS

Efficient communication in general purpose CMPs or application specific multi node SoCs is obtained by the NoC design. Simply state, in NoCs the Intellectual Property (IP) blocks transfer data through the network via routers instead of traditional bus. Each router connected to its adjacent routers and IP blocks with the specific number of ports. Each port consists of input buffers, switch and arbitration. The arbitration determines the output port. Designers can select different types of traffic configuration for reaching flits to its dedicated destination.

The main NoC design challenge is defining network architecture regarding to the topology graph, routing algorithm and switching strategy for a target application.

Until now, the proposed tool in this paper performs a limited set of configuration as follows:

Topology: m×n mesh thanks to better scalability, regularity and ease of implementation in silicon

Traffic distribution: true traffic characteristic of the target application based on different mapping algorithms

Routing algorithm: deterministic routing

Switching strategy: wormhole

Wormhole routing is well known for efficient communication in NoCs designs. It is operates at flit level and reduces latency requirements. This flow control strategy without using virtual channels is prone to head of line blocking and significantly diminishes network performance. Consequently, with this tool, one can choose the number and depth of each virtual channel for preventing mentioned problem. Virtual channels thanks to providing alternative paths for incoming flits boost the network performance [10, 11 and 12].

IV. Flow Diagram of the Simulator

This Section provides details of all required steps for ARTEMIS tool (Fig. 1). Besides, Fig. 2 presents the main page of this tool.

These steps are as follows:

1- An application characteristic and an architecture characteristic are read from input files.

2- The task graph and topology graph are created.

3- The placement of application tasks are determined by different mapping algorithms. Then, best mapping in term of communication cost and energy consumption is discovered.

4- All sets of flows between each source and destination node are drawn based on the deterministic routing algorithm.

5- The percentage of each link and router usage are measured and the network links capacity and the number of VCs per router port are initialized.

6- All mandatory definitions in Table 1 are read from input files.

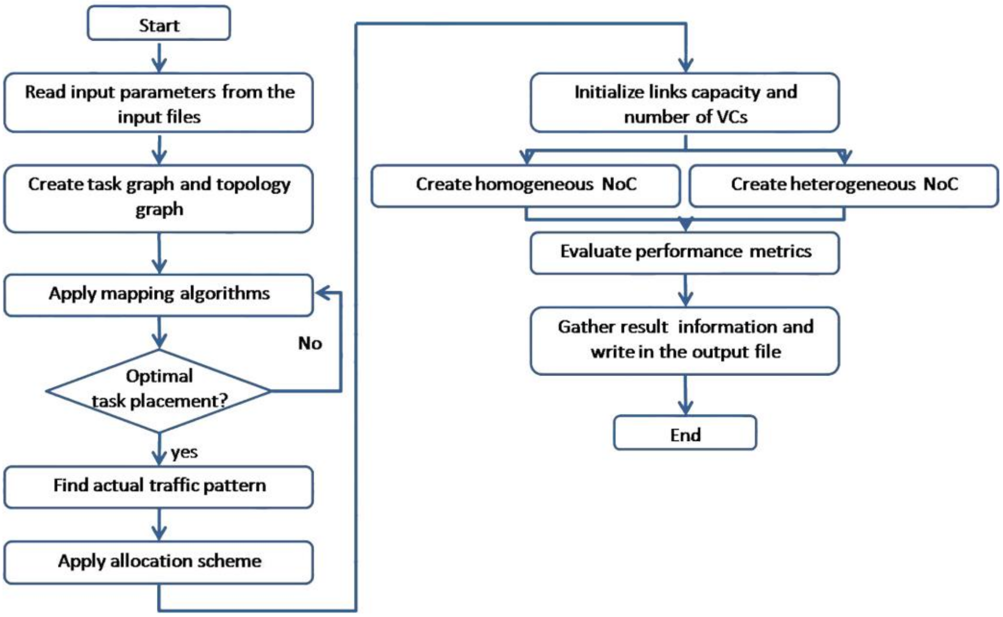
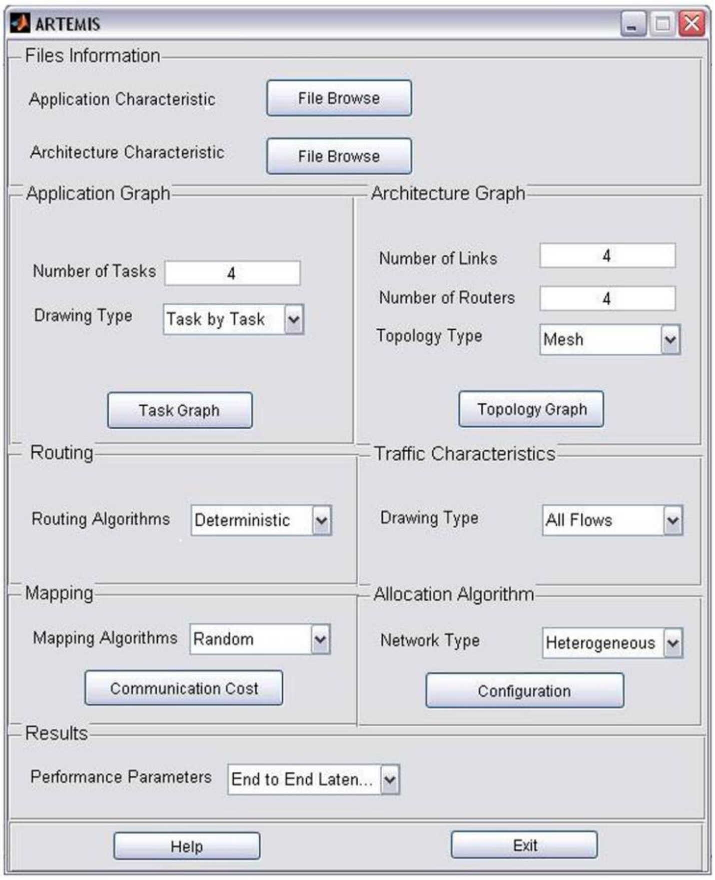

Start
Read input parameters from the input files
Create task graph and topology graph
Apply mapping algorithms
Optimal task placement?
No
yes
Find actual traffic pattern
Apply allocation scheme
Initialize links capacity and number of VCs
Create homogeneous NoC
Create heterogeneous NoC
Evaluate performance metrics
Gather result information and write in the output file
End

Fig. 1: Flow diagram of ARTEMIS


ARTEMIS

Files Information

Application Characteristic

Architecture Characteristic

Application Graph

Number of Tasks 4

Drawing Type Task by Task

Architecture Graph

Number of Links 4

Number of Routers 4

Topology Type Mesh

Routing

Routing Algorithms Deterministic

Traffic Characteristics

Drawing Type All Flows

Mapping

Mapping Algorithms Random

Allocation Algorithm

Network Type Heterogeneous

Results

Performance Parameters End to End Laten...

Fig. 2: Main page of ARTEMIS tool

7- The homogeneous network and heterogeneous network are created by allocation algorithm.

8- The performance metrics are initialized and evaluated.

9- The results are written to the output file.

Briefly, the different components of ARTEMIS simulator are described below.

Application graph: the application graph component represented task graph component as well. It receives the number of tasks in the target application as input and it draws the application graph. This component also displays and determines the name of each task, the weight and direction of each arc between two tasks as the required bandwidth and the communication direction between them respectively.

Architecture graph: the architecture graph component represented topology graph component as well. It chooses the type of the topology and the number of links and routers in the specific topology and it builds the architecture graph and connects each node to one router.

Mapping: this component maps the application tasks onto the topology graph based on different mapping algorithms. The objective of this component is improving power and performance of the NoC by discovering the mapping with minimum communication cost. The user can select the mapping algorithm and see the place of each task in the topology graph.

Routing algorithm: routing path from source to destination is determined by routing algorithm. Network power and performance depend on the routing algorithm. This component provides the routing technique for reaching to destination node.

Traffic pattern: the traffic characteristics component of this tool presents the actual traffic pattern of each application by specific routing function and it shows all communication flows between each source and destination node across the entire network.

Allocation algorithm: This component parameterizes the number and depth of VCs and the capacity of each link by applying an efficient allocation scheme provided with this tool that will be explain in the next Section.

Results: this tool calculates the network latency and throughput as the performance metrics for both homogeneous network and heterogeneous network. ARTEMIS presents the comparison results by some plots.

Help: this component provides the necessary guide of using the simulator for the NoC designers.

V. VCs and Links Capacity Allocation Scheme

Since router and link both influence the network power and performance, some of the researches designed heterogeneous NoC only with different number of VCs at each router port [11, 12 and 13] or only with different link capacities [10]. Furthermore, other works used the predefined routers across the network with specific number of VCs and links capacity [14, 15 and 16]. However, this tool configures heterogeneous network by performing an allocating scheme, which formally described in this Section for discovering the efficient VCs and capacity assignment use the actual traffic pattern of the target application to enhance network performance.

More precisely, the allocating scheme decides the links capacity and VCs to each unidirectional router port according to the number of VCs and amount of links capacity budget for different flows. These flows are between communication nodes from source to destination with determined routing algorithm.

Fig. 3 presents a heterogeneous router assigning non-uniform links capacity and VCs to each input channel based on the communication pattern of the target application for maximizing network performance. For example, in this Figure, the north router port and the connected link to this port has three VCs and 17MBps capacity respectively while distributing resources to other router ports is the same.

The problem of VCs and links capacity allocation for satisfying performance under latency constraints is formulated based on these notations as follow:

Given:

1- Application graph

2- Architecture graph

3- Mapping function and a deterministic routing algorithm

4- Total number of VCs budget

5- Sum of network links capacities budget

Determine:

1- Number of VCs for each router port

2- Amount of capacity for each network link

Such that:

The end-to-end latency is minimized under VCs and links capacity assignment.

In Table1, the general notations and definitions using during following allocation algorithm are summarized.

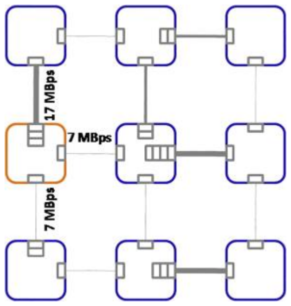


Fig. 3: Non-uniform links capacity and VCs at each router port

Table 1: Parameter definitions

Notation	Definition
F	The set of all flows from each source to destination
f	A flow from the set *F*.
m(f)	Mean packet length of flow *f* [flits].
λ(f)	Packet generation rate of flow *f* [packets/sec]
flit	Flit size (bit)
Path(f)	Path of flow *f* composes of some links and routes
F(l)	The number of competing flows over link
F(p)	The number of competing flows over ingress port
L	The set of all links in the NoC
l	A link from set *L*.
C(l)	Capacity of link [bits/sec]
N(C)	Total amount of allocated capacity to all links
MC(l)	Maximum amount of capacity for link
LCB	Links Capacities Budget.
P	The set of all router ports in the NoC.
p	A port from the set *P*.
V(p)	Number of VCs at port *p*.
N(V)	Total number of allocated VCs to the network routers
VCB	VCs Budget.
MV(P)	Maximum number of VCs per port
VCD(p)	VC depth of each port *p*
LE2E(f)	End to End Latency of flow *f*

Algorithm 1 defines the links capacity and VCs allocation scheme. At first, this algorithm initializes the capacity of each link (lines 1- 3), number of VCs and the depth of each VC (lines 4- 7) in the network. After initialization phase, the algorithm calculates the end-to-end latency by delay model in [17] for each flow from specific source to specific destination (line 9). If the obtained delay is longer than target delay, the algorithm searches a link *l* (lines 11- 13) and a port *p* (lines 14- 16) with maximum number of competing flows between all links and all ports construct the path flow respectively. After that, the algorithm adds a positive small amount of capacity only to link *l* respect to sum of the links capacities budget. If the capacity of link *l* become more than maximum amount of capacity for link *l*, the algorithm finds the next link with maximum number of competing flows in the path flow (lines 17- 22). Additionally, the algorithm adds exactly one VC only to port *p* regard to the total number of VCs budget and maximum number of competing flows over ingress port *p*. If the added VC to ingress port *p* exceeds the maximum number of VCs in port *p*, the algorithm selects the next port with maximum number of competing flows in the path flow (lines 23- 29). Then, the algorithm calculates the resulting packet delay again for this new network configuration (lines 30 and 31). The algorithm continues until a network configuration with better performance than the target delay is found (lines 10- 32).

```
1.  for each link l do
2.      C(l )=sum (λ(f). m(f).flit)    : f∈F, l∈Path(f)
3.  end
4.  for each port p do
5.      V( p) =1
6.      VCD (p) = min C(l)
7.  end
8.  for each flow  f∈F do
9.      LE2E(f) is calculated by latency model
10.     while LE2E(f)> target_LE2E(f)  do
11.         for each l∈Path(f) do
12.             l′= l with max F(l)
13.         end
14.         for each  p ∈Path(f) do
15.             p ´=  p  with max F(p)
16.         end
17.         while N(C) <= LCB do
18.             C(l')= C(l′)+ε
19.              if  C(l´)> MC(l') then
20.                  l′ =next l  with max F(l)
21.              end
22.         end
23.         while N(V) <= VCB do
24.           if (V(p')< F(p′ )  && V(p')< MV(p′ )) then
25.               V(p')= V(p')+1
26.           else
27.               p′ =next p with max F(p)
28.             end
29.           end
30.           L'E2E(f) is calculated by latency model
31.          LE2E(f)= L'E2E(f)
32.     end
33. end
```

Algorithm 1: capacity and VCs allocation algorithm

VI. Performance Evaluation and Synthetic Example

After creating a homogeneous NoC or heterogeneous NoC by allocation algorithm, designers can evaluate the network performance based on the actual traffic pattern of the target application. ARTEMIS is capable to measure end-to-end latency and network throughput as the main performance metrics and link utilization, buffer utilization, communication cost as well. At first, this Section discuses some parameters and then confirms the simulation tool with a simple synthetic example.

A. Network Latency

The average latency for a given set of flows is defined as the time required for complete transfer of a packet from the generation of the packet until its arrival to the destination [17]. Flits of the specific flow in the source node transfer from dedicated network links and routers and reach to destination. The position of the source and destination nodes in the network and routing algorithm determine message latency [18].

The latency of the homogeneous NoC configuration or heterogeneous NoC configuration creating by allocation scheme is one of the performance metrics, which is explored with

ARTEMIS tool. The average latency is measured based on delay model in [17].

B. Network Throughput

Throughput is another performance metric exploring by proposed tool. This metric indicates the maximum accepted traffic from the homogeneous NoC configuration or heterogeneous NoC configuration. A model for calculating the Throughput has been proposed in [18] (Eq.1). ARTEMIS defines the network throughput based on [18] as well.

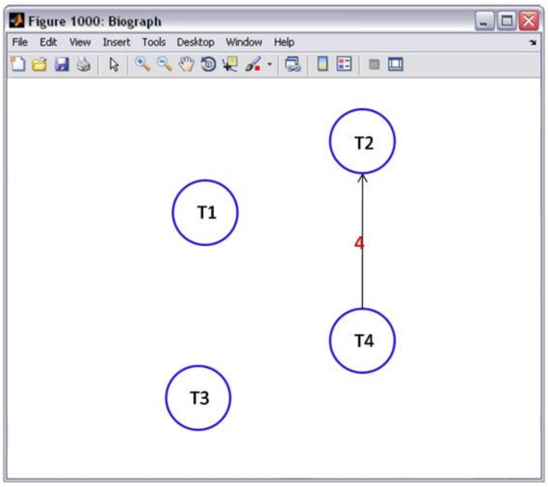
TP= (Total message completed). (Message length)/ (Number of IPblocks). (Total time) ***(Eq.1)***

C. Synthetic Example

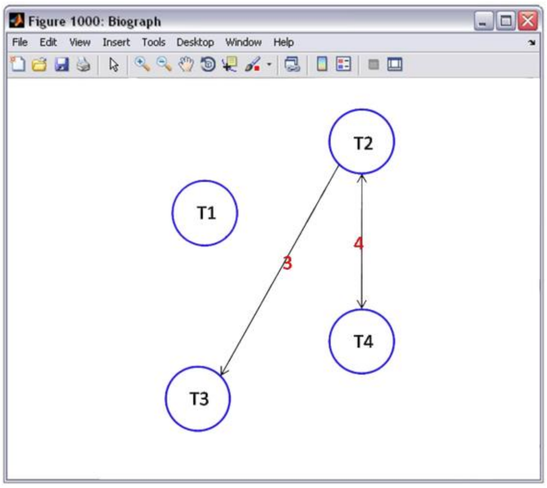
In order to verify the simulator, a simple synthetic example is used. The simulation performs based on the real traffic pattern and the same packet injection rate for all flows of the case study. The ARTEMIS tool evaluates the performance parameters for a set of flows in the NoC.

ARTEMIS in Fig. 4 shows the name of each task and requires bandwidth sent from each source task to each destination task as the task graph for the synthetic example. Fig. 4 (a, b, c and d) indicate the process of drawing the specific application tasks with four tasks and six weighted connections.

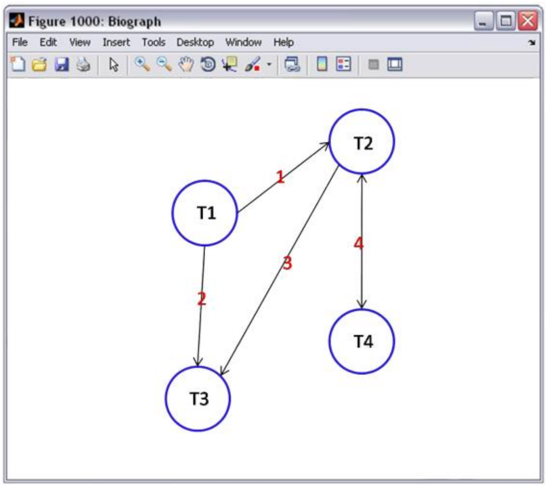
According to the task graph as shown in Fig. 4, application tasks map into 2×2 mesh by various mapping algorithms. The mapping with minimum communication cost and energy consumption is the best result. After finding the final mapping, this tool creates the homogeneous NoC and heterogeneous NoC by applying efficient allocation scheme respect to actual traffic of target application.



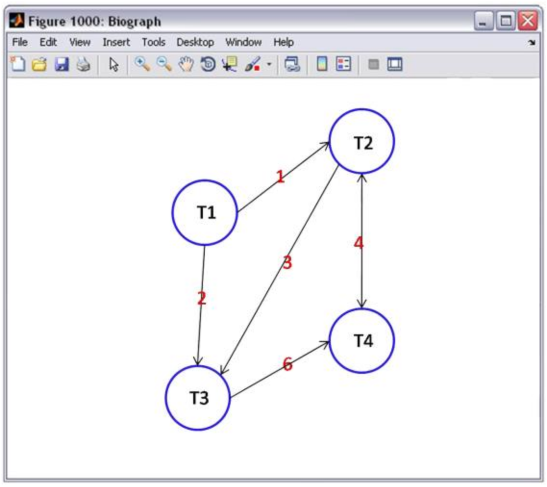
(a) Step 1



(b) Step 2



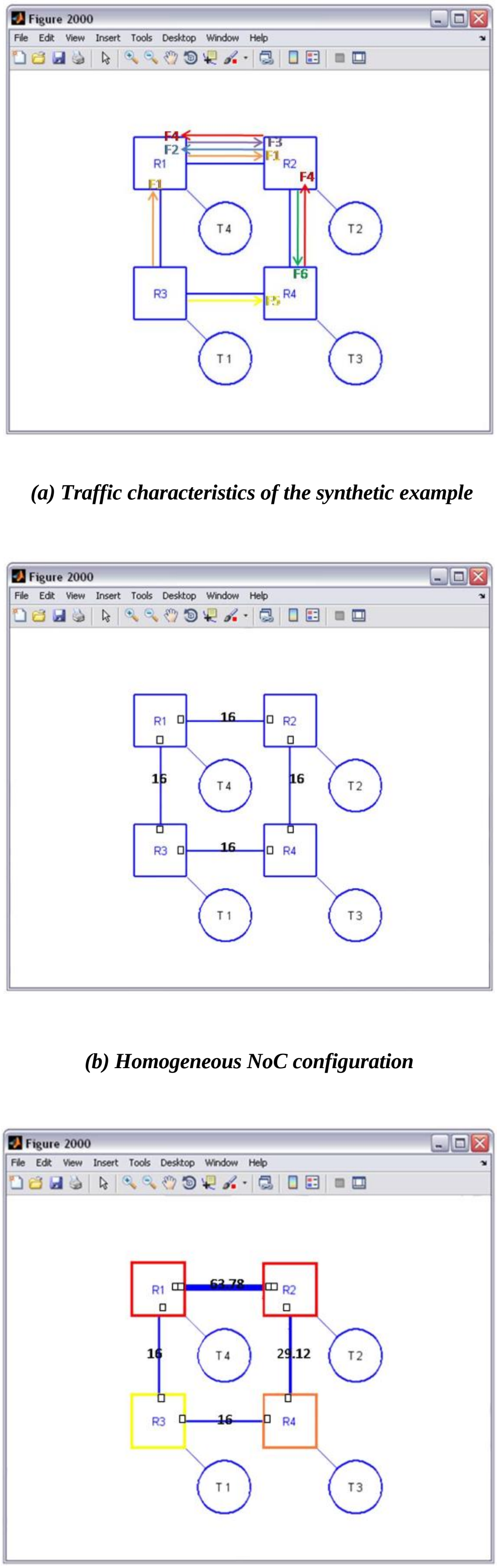
(c) Step 3



(d) Step 4

Fig. 4 (a, b, c and d): Process of drawing task graph of the synthetic example

Fig.5, Fig.6 and Fig.7 are extracted from ARTEMIS tool as well. The following part explains these figures in detail respectively.



(a) Traffic characteristics of the synthetic example

(b) Homogeneous NoC configuration

(c) Heterogeneous NoC configuration

Fig. 5 (a, b and c): Process of the network design for target example

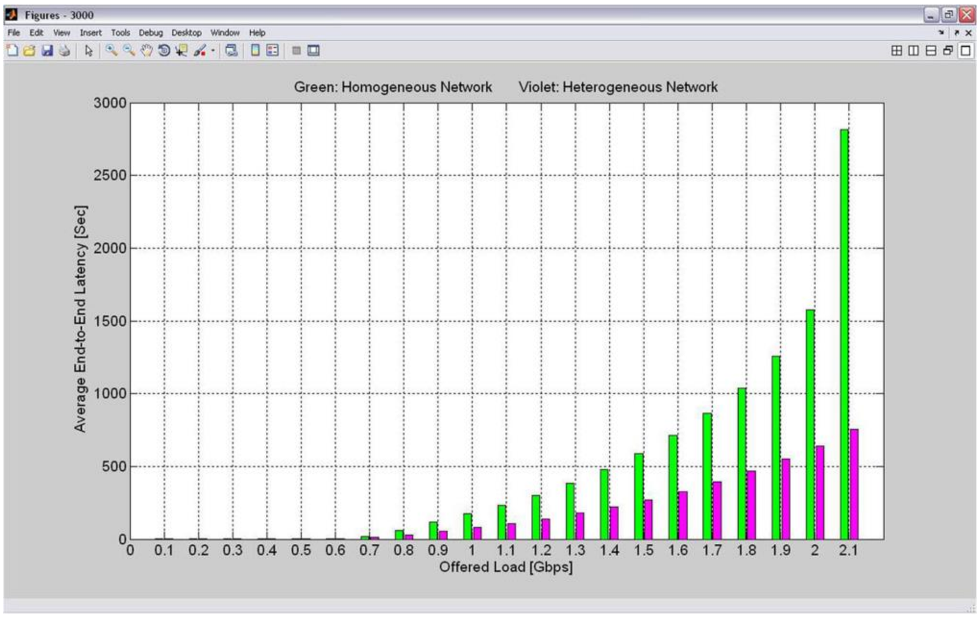


Fig. 6: Comparison of the average end-to-end latency for the homogeneous NoC and heterogeneous NoC configurations in different offered load

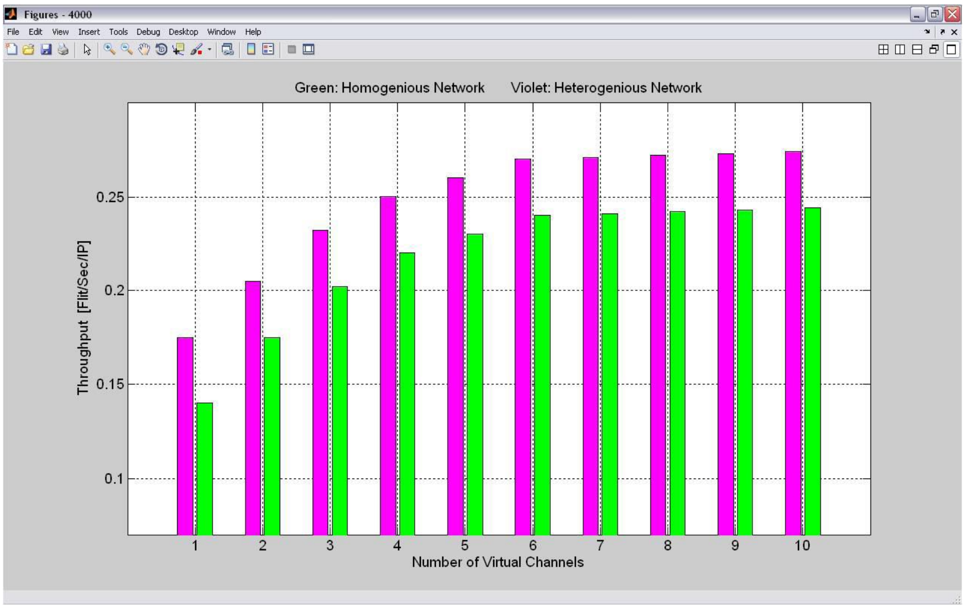


Fig. 7: Comparison of the throughput for the homogeneous NoC and heterogeneous NoC configurations in different offered load

Fig. 5(a) shows the actual traffic of the synthetic example extracted from ARTEMIS tool. Fig. 5 (b) and Fig. 5(c) present the homogeneous NoC design and heterogeneous NoC design in the target example respectively. In the homogeneous network design, the number of VCs per router port is one. Furthermore, the capacities of all links are 16 Gbps. The capacity and the number of VCs between each two routers in the heterogeneous network design are shown in the Fig. 5(c) as well.

Fig. 6 and Fig. 7 are extracted from ARTEMIS tool. Figure 6 compares the average end-to-end latency versus offered load in the homogeneous NoC design and heterogeneous NoC design by the analytical delay model in [17]. Fig. 7 presents throughput results for homogeneous NoC and heterogeneous NoC design based on the equation in [18].

The comparison results between the homogeneous NoC design and the heterogeneous NoC design for the synthetic example in Fig. 6 and Fig.7 demonstrate the performance boosting by heterogeneous NoC design.

VII. CONCLUSION

This paper has presented the ARTEMIS tool to model the complex homogeneous NoC and heterogeneous NoC. This simulator is capable to construct the network configuration regarding to the topology graph, routing algorithm and switching strategy for a target application. Moreover, the major purpose of this tool is careful non-uniformity assignment for enhancing the performance under defined criteria constraints. An efficient VC and link capacity allocation schemes with this tool come up this goal. Beside, ARTEMIS evaluates the performance of the designed network under actual traffic pattern of the target application. Although the tool provides only the limited set of homogeneous network configuration and heterogeneous network configuration, suggested ideas such as implementing other types of the traffic distribution, applying other types of routing algorithm and configuring other types of topology graph for the next version of ARTEMIS are recommended.

REFERENCES

[1] Y. Ben-Itzhak, I. Cidon, and A. Kolodny, Optimizing heterogeneous NoC design, Proceedings of the International Workshop on System Level Interconnect Prediction (2012), 32-39.

[2] J. Hu and R. Marculescu, Application-specific buffer space allocation for networks-on-chip router design, IEEE/ACM International Conference on Computer Aided Design (2004), 354-361.

[3] P. Abad, P. Prieto, L. Menezo, A. Colaso, V. Puente, and J. A. Gregorio, Topaz: An open-source interconnection network simulator for chip multiprocessors and supercomputers, Sixth IEEE/ACM International Symposium on In Networks on Chip (2012), 99-106.

[4] N. Agarwal, T. Krishna, L. S. Peh, and N. Jha, Garnet: A detailed on-chip network model inside a full system simulator, IEEE International Symposium on Performance Analysis of Systems and Software (2009), 33- 42.

[5] N. Jiang, D. Becker, G. Michelogiannakis, J. Balfour, B. Towles, D. Shaw, J. Kim, and W. Dally, A detailed and flexible cycle-accurate network-on-chip simulator, In IEEE International Symposium on Performance Analysis of Systems and Software (2013), 86- 96.

[6] A. T. Tran and B. Baas, NoCTweak: a highly parameterizable simulator for early exploration of performance and energy of networks on-chip, Technical Report, VLSI Computation Lab, ECE Department, UC, University of California, Davis, [online] http://www.ece.ucdavis.edu/vcl/pubs/2012.07.techreport.noctweak/ (2012).

[7] [7] M. Palesi, D. Patti and F. Fazzino, NOXIM released software (2005- 2010). [online] http://noxim.sourceforge.net.

[8] V. Puente, J. A. Gregorio, and R. Beivide, SICOSYS: an integrated framework for studying interconnection network performance in multiprocessor multiprocessor systems. Proceedings of the 10th Euromicro conference on Parallel, distributed and network-based processing (2002), 15-22.

[9] Y. Ben-Itzhak, E. Zahavi, I. Cidon, and A. Kolodny, HNOCS: Modular Open-Source Simulator for Heterogeneous NoCs, International Conference on Embedded Computer Systems (2012), 51-57.

[10] Z. Guz, I. Walter, E. Bolotin, I. Cidon, R. Ginosar, and A. Kolodny, Network delays and link capacities in application-specific wormhole NoCs, VLSI Design(2007).

[11] T. Huang, U. Ogras, and R. Marculescu, Virtual channels planning for networks-on-chip, the 8th International Symposium on Quality Electronic Design (2007), 879- 884.

[12] A. Kahng, B. Lin, K. Samadi, and R. Ramanujam, Trace-driven optimization of networks-on-chip configurations, In Design Automation Conference (DAC), 47th ACM/IEEE (2010), 437- 442.

[13] M. Al Faruque and J. Henkel, Minimizing Virtual Channel Buffer for Routers in On-Chip Communication Architectures, Proceedings Design, Automation and Test in Europe (2008), 1238-1243.

[14] M. Kreutz, C. Marcon, L. Carro, F. Wagner, and A. Susin, Design space exploration comparing homogeneous and heterogeneous network-on-chip architectures, Proceedings of the 18th annual symposium on Integrated circuits and system design(2005), 190- 195.

[15] A. Mishra, N. Vijaykrishnan, and C. Das, A case for heterogeneous on-chip interconnects for CMPs, Proceedings of the 38th annual international symposium on Computer architecture (2011), 389- 400.

[16] A. Bakhoda, J. Kim, and T. Aamodt, Throughput-effective on-chip networks for many node accelerators, Proceedings of the 43rd Annual IEEE/ACM International Symposium on Micro architecture (2010), 421- 432.

[17] Y. Ben-Itzhak, I. Cidon, and A. Kolodny, Delay analysis of wormhole based heterogeneous NoC, Proceedings of the 5th ACM/IEEE international symposium on Networks-on- Chip (2011), 16- 168.

[18] P. P. Panda, C. Grecu, M. Jones, and A. Ivanov, Performance Evaluation and Design Trade-Offs for Network-on-Chip Interconnect Architectures, IEEE Transactions on Computers (2005), Vol. 54, No. 8, 1025-1040.

Fatemeh Vardi received M.Sc. degree in Computer Architecture from Science and Research Branch of Islamic Azad University (SRBIAU), Tehran, Iran in 2009. She is currently a Ph.D. student in Computer System Architecture in Islamic Azad University, Science and Research Branch, Tehran, Iran. She is also a Faculty member of Computer Engineering Department Parand Islamic Azad University (PIAU). Her research interests is Energy Management in Embedded System with emphasis on Application Mapping and Network Topology in Multi-core Architectures.

Ahmad Khadem-Zadeh received M.Sc. and Ph.D. degrees in Digital Communication and Information Theory and Error Control Coding from the University of Kent, Canterbury, UK respectively. He is currently the Head of Education and National Scientific and International Scientific Cooperation Department at Iran Telecom Research Center (ITRC). He was the head of Test Engineering Group and the director of Computer and Communication Department at ITRC. He is also a lecturer at Tehran Universities and he is a committee member of the Iranian Electrical Engineering Conference Permanent Committee.

Midia Reshadi received his M.Sc. degree in Computer Architecture from Science and Research Branch of Islamic Azad University (SRBIAU), Tehran, Iran in 2005. He also received his Ph.D. degree in Computer Architecture from SRBIAU, Tehran, Iran in 2010. He is currently Assistant Professor in Faculty of Electrical and Computer Engineering of SRBIAU.