Enhancing virtual switching system of on-chip networks

Rony Kassam, Talal Al-Aateky, Radwan Dandah

Abstract— the huge datacenters in the world - used in the social networks, researching and computing centers, and storage devices - have faced fundamental problem reflected in the need for high throughput to switch between the large numbers of virtual machines installed in these datacenters. The increased use of cloud applications will increase the need for greater numbers of virtual machines. In order to meet the requirements of this increase, it must be increased the hardware capabilities in the datacenters, through increasing the number of cores in the servers. Therefore, there is a need to provide network architecture that can connect all of these cores and chips very quickly. Consequently, it is necessary to enhance the virtual switching system to achieve the maximum integration and efficiency between the network segments, and analysis the scheduling and virtual queuing algorithms depending on the decisions from the experiences implemented using high-level source code and hardware description scheme with Xilinx platform.

Index Terms— Datacenters, Hardware description scheme, Network on Chip, Network architecture, Virtual switching system, MPSoC, Xilinx, SDF^3, MAMPSx.

1 INTRODUCTION

The Computer architecture of the datacenters consists of multiple processors and memories connected through interconnection networks, which have multiple schemes like mesh, torus, tree, and others. These schemes have built physically and logically on the chip, through advanced technologies such as NoC (Network on Chip). By using this network, it can balance the load of the computing and the data exchange among the installed processing elements in the datacenter. Especially with the increasing of the number of these elements, and the need to achieve the accurate level of the effective work of the datacenter, so there is not any additional processing elements consumption if there is no need. In the case of defining the requirements through the application installed in the datacenter, there will be proper mapping between the application and hardware architecture. It will be effective mapping between the VMs (virtual machines), the NoC routers, and processing elements, which will be responsible for operating these machines, switching, and routing among them, taking into the consideration throughput and power consumption. In this research, enhancing and optimization have performed on NoC to achieve the maximum integration and efficiency between the network segments, and analysis the scheduling and virtual queuing algorithms depending on the decisions from the experiences implemented using C source code and hardware description scheme with Xilinx platform.

- *Eng. Rony Kassam – MSc Student – Department of Computer Systems and Networks – Faculty of Information Engineering – Tishreen University – Lattakia – Syria, PH-00963988225669. E-mail: rony.kassam@live.com*
- *Dr. Talal Al-Aateky – Assistant Professor – Department of Computer Systems and Networks – Faculty of Information Engineering – Tishreen University – Lattakia – Syria, PH-00963470858. E-mail: talal.alaateky@tishreen.edu.sy*
- *Dr. Radwan Dandah – Professor – Department of Computer Systems and Networks – Faculty of Information Engineering – Tishreen University – Lattakia – Syria, PH-00963944535235. E-mail: radwan.dandah@tishreen.edu.sy*

Research steps have been set to find the correct practical environment to carry out scenarios that give the desired results. The application and load installed on the environment were descripted by using the resources from Netmap[1], in order to test the application and identify the source file that must be relied upon. It was also analyzed the link between VMs in the virtual switching system, by implementing nSwitch[2], and then come with the necessary to use MPSoC (Multiprocessor System on Chip) architecture, in order to reach a valid and workable results. Then, a theoretical and practical evaluation has been studied on routers, tiles, and network schemes of the NoC, to choose the description scheme, simulator, and development tool in this research [3]. It has been identified SDF3 (Synchronous Data Flow For Free) as a description scheme, MAMPSx (Multi-Application Multi-Processor Synthesis) tool to generate the mapped MPSoC architecture, Xilinx platform in order to get the results, and Matlab to build the diagrams.

2 BACKGROUND AND INITIAL EXPERIMENTS

2.1 Application and load

To define the application and the load installed on the practical environment, it has been relied on the Netmap[1], which is Ethernet switch design for VMs, it is provide high-speed connections between these machines depending only on the software. There is a problem faced by the network ports, especially in real systems and hardware, it is the latency of system calls and memory allocations for each packet. That is, because using specific API (Application Programming Interface) for the socket, more specifically the use of "libpcap" package. Therefore, many systems calls and memory operations will execute only to access to the kernel level. The solution used in Netmap[1] greatly facilitate the process, by forming data paths between the wire and the application at the user level; making the application very close to the hardware, this principle is the goal that we seek in this research.

During the installation of Netmap on Ubuntu 14.10, according to the installation guide, it appears a problem about the need for the kernel source and header files, and the need for the driver of the network interface. Since, it is necessary to install the source packages, because it will rebuild the driver of the network interface, in order to insert the "netmap.ko" module into the system. After that, a new problem appears when rebuilding the driver, because Netmap supports only specific network interface. Because it is not available, it was set a parameter "no-drivers" to generate virtual network interface. All of that because the need to identify the applications that deal with Netmap, which are very the research. It is chosen packet generator, the source file "pkt-gen.c", because, this application can send the packets in certain sizes from sender to receiver on the same environment through physical layer. Because, this generator is independent on the processors scheme, but its dependence on the number of existing processors or cores, and also its dependence on the network interface, therefore, it has been modified to deal with the processing elements and routers within the NoC. Taking into consideration the use of appropriate tools in order to coordinate the work of this generator to be as a valid input for the hardware description scheme and the other tools.

2.2 VMs networking technology

About the technologies that connect VMs with the virtual switching system, many enhancements of the SR-IOV (Single Root I/O Virtualization) have been done. SR-IOV is unable to support switching between two VMs on the same computer only through software switching. nSwitch is one of These enhancements to support hardware switching between VMs. nSwitch technique was compared with software switching vSwitch, and IEEE 802.1Qbg and 802.1Qbh pSwitch techniques[2]. As a result, using the vSwitch, which implemented by Citrix XenServer 6.5 and OpenVSwitch. Due to the heavy load on the CPU and the I/O queues to transfer between two VMs, the maximum throughput was 744 Mbps, and this value will be affected by increasing the VMs. About the IEEE 802.1Qbg and 802.1Qbh pSwitch technologies, and because the traffic was transferred through the physical network interface, so the throughput will be like the network interface throughput; 1 Gbps, and this value will be affected by the external transfer. The nSwitch technology has exceeded the delay caused by the network interface in the previous case, and therefore the throughput does not suffer from any limitations, only as to the PCIe 3.0 32-bit bandwidth, which is eight GigaTransfer/s.

Starting with nSwitch experience in this research, by reforming the network interface to enable two PFs (Physical Function), which are PF0 and PF1, one for each processing unit. Each function can take number of VFs (Virtual Function). Installing this interface using Xilinx Vivado 2014.2, then building the design depending on the operating guide and experience within this platform [5]. It has been used Xilinx Virtex-7 FPGA, then set two PFs and six VFs with PCIe X8 Gen3, also set interface AXI4-stream 256-bit, memory 4KB 32bit, and finally use read and write "Dword" 32bit transactions for memory mapping.

During the work, it shows the necessary for a physical testing interface, but because of the difficulty to provide it, it has been resorting to implement this interface virtually within the ".bin" file. Taking the advantage of Xilinx ability to run specific Linux operating system on the virtual interface, but it did not succeed in this experience, because it works only with some virtualized boards like MPSoC boards. Therefore, it has to do the work on MPSoC architecture to prepare the NoC within. This architecture can load the operating system, and at the same time can be modified to perfectly fit what we want without having to provide PCIe interface, only integrate the work into the board directly.

2.3 NoC architecture

This architecture benefited from the OSI (Open Systems Interconnection) model, in order to transfer between the specific components IPs (Intellectual property) [6], which can be processors, memories, and others. The layers of the NoC are NI (Network Interface), PL (Physical link), R (Router), and IP [6]. Where, NI (data-link layer in the OSI) tasks' are messages encapsulating and caching, PL (physical layer in the OSI) tasks' are Signals and Phits (Physical Unit) transferring, R (network layer in the OSI) tasks' are network scheming, routing and switching, and IP (transport layer in the OSI) is responsible for data transferring between the specific components.
Within R, there are many technologies, starting from VCTlite [4], which is a new implementation of the VCT (Virtual Cut Through) adapted to multicore and multiprocessors on the chip. It can take advantage of the VCT, which is characterized by supporting broadcast and multicast. However, VCT requires large buffer size. This is not suitable for on-chip applications, but it is suitable for WH (Worm Hole), which deal with small buffer size. Therefore, the target is to reach the best technology that takes features of the two technologies. It is VCTlite, which uses buffer size such as the size of control message, whereas data message must be packetized [4].

The basic architecture of this router equipped with input buffers, as well as five-stage pipelining: IC (Input Controller), RT (Routing), VA-SA (Virtual channel And Switch Allocator), XB (Crossbar), and LT (Link Traversal) (see Fig. 1). Each router consists of a number of input and output ports, one port communicates with the processing element, while the remaining ports communicate with neighbors routers, as to the specified network scheme [4]. About output units [7], which track the status of the receiving VC, by using a group of registers. "input_vc" records the VCs reserved for this output, "idle" indicates whether the VC has received the flit tail of last packet, and "Credits" records a certain value reflects the possibility of the port to proceed in its function; if it contain the value "1" then can transfer one flit and so on.

Each PL allocates a set of VCs in order to support the coherence protocol that is used to coordinate between caches. It has allocated one buffer to each VC with IC. In design view, the buffer is implemented by conventional shift register. Despite the fact that this structure is not effective in terms of power consumption, it does not introduce any additional circuits, which lead to increased router latency. VCs are grouped, therefore any message can use the VCs of the same group, and thus the traffic in different VCs groups cannot be mixed. This constrain is very necessary in order to avoid the case of a deadlock within coherence protocol.

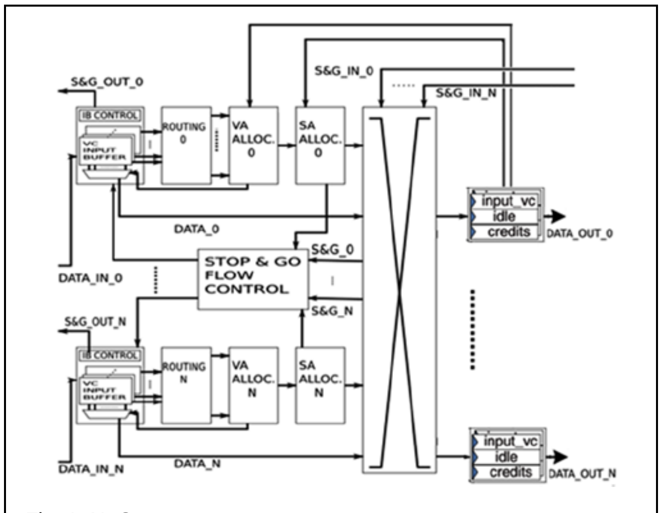

Fig. 1. NoC router components.

The parameters of this router [4], VCs number is 5, link width is 3 bytes, Flit size is 3 bytes, input buffers can store five Flits to achieve short RTT (Round Trip Time) between neighboring routers. This time is affected by propagation and flow processing delay. So small buffer size will appear bubbles in the communication stream [7].
For flow control, it was implemented Stop&Go protocol to control the Flits between neighboring routers. About routing stage on each input port, it has been achieved to support a routing algorithm DOR (Dimension Order Routing) [4]. The VA of the input port determines which VC will compete with other VCs of other input ports, in order to reach the XB. The SA to be achieved for each output port, and use round-robin arbitrator [4]. SA permissions will control for the entire packet, in order to give priority for the VCs that got permission through the switch [7]. To improve performance, VA-SA stage does not depend on Flit-level arbitration, as the case in WH, but they continue to route the Flit from the current message as long as there are Flits ready to send. However, this is a critical stage for the router, because it determines the operating frequency for this router. To reduce power consumption, clock gating is used [7].

2.4 Specific MPSoC generation

Providing a design to evaluate the throughput constrained applications on the MPSoC architecture [8]. It was merged many schemes and tools to apply the load on Xilinx development platform. It was depended on SDF, which represents the application, so it can be mapped with hardware units, and determines the throughput in the most difficult status. After the application modeling, it was used C language in order to implement the scheme depending on a certain hardware architecture. Using these inputs, it is generated MPSoC architecture designed specifically for the requirements and the schemes of the application. Then take advantage of the MAMPS tool in order to operate this architecture with Xilinx.

In order to achieve the best performance, taking into account time to market requirement, it presents MAMPSx to address these challenges. It is design flow, which takes application and architecture scheme and generates MPSoC architecture with hardware and software models suitable for Xilinx. MAMPSx includes the stages, as follow: BONES is a model, depended on an algorithm recursively improving the design and quality time. DSE (Design) is an effective way to check MPSoC design in order to achieve performance and area conditions. GEN (Generate) is a structured approach, which generates multi-processor systems with the hardware infrastructure for a specific application, suitable with Xilinx platform [8].

The target of this research is to prepare an effective laboratory environment, includes general application capable to form and modify network packets. In addition, to link the application through description languages, which map it on a specific hardware environment, and at the same time, it is not limited to a specific development platform [9]. Therefore, according to [1], packet generator has been selected as an application, but this application is dealing directly with the operating system, so it has been modified in order to be deal with the laboratory environment in this research.

By reprogramming “pkt-gen.c”, detailing its components and data streams within, and linking them with the description schemes. Then the files were initially created “sdf3PI-noc.opt”, “archgraph-noc.xml”, and “usecase.xml”, which represents the settings required by the tool SDF3. These files are modified for each scenario, in order to be examined gradually. After that, all of the files become an input for MAMPSx tool.

Beginning with the data stream analysis, a consistency of the scheme is analyzed to prevent the deadlock in the application by examining its resources, also by using the following rule: the scheme of “archgraph-noc” is consistent, if the repetition vector is not equal to the null vector. It has been planned MCM (Maximum Cycle Mean) analysis, to determine the throughput in the scheme, in order to organize tokens registering and releasing of the Actors. The scheme is analyzed in terms of repetition vector, in order to calculate all the comparisons between allocated buffers of channels identified in the scheme and the mapped maximum throughput. In order to make the resources of this model far from the unlimited, it is determined the number of edges within the Actors, where this number will be adjusted through the various scenarios.

At this stage, in order to reduce the use of resources and provide guarantees on the application throughput in MPSoC system, it is used iterative design flow consisted of four stages [10]. it automatically and frequently does all stages, but the only manual stage is to determine the application scheme “usecase.xml”, NoC scheme “archgraph-noc.xml” within the setting file. As a final step, it is planned to export several formats fit with the display, as well as fit with the input of MAMPSx tool. SDF3 output files are placed into a folder that includes “archgraph-noc.xml”, “usecase.xml”, and the source files that implement the actors of the Packet generator.

Then, it is run the tool “mamps_template” on the application, which was named within the setting files, as "RTRpktgen". The generated files are "RTRpktgen.tar.gz" and "RTRpktgen.backup.tar.gz", where they contain the full Xilinx project. In order to get the executable file on the MPSoC, it is compiled through "make" command.

Thus, the system has become ready for evaluation within the Xilinx development platform according to various scenarios. At each scenario, it will be back to the beginning of the experiment and make adjustments on the parts of the MAMPSx. This modification has named RTRNoC, which separates the formation of the NoC from the IP types within the different version of Xilinx platform. In addition, it supports Stop&Go flow control protocol. After doing all the modification, it will be started from the beginning of the experiment steps, to analysis and generate. The part that concerns us greatly in the research is the part located in the generated folder "pcores", which contains a hardware description by VHDL (VHSIC (Very High Speed Integrated Circuit) Hardware Description Language) for the hardware components of the NoC, NI, and PE which is represented by the MicroBlaze processor.

3 Evaluations and Results

Talking about the evaluation scenarios, and identifying the parameters and metrics of the experiments, to reach for the enhancement. It is implemented 218 scenario; each scenario has 40 input parameters as their categories. In the scenario (X for example), it is modified some parameters' values, according to the results from the scenario (X-1), and so on. The number of modified parameters are 18 parameters, whereas the other parameters remain constant. For system's metrics that determine the effectiveness and quality of the system are frequency, area occupied by the chip containing the whole system, latency of links, protocols and components consisting the system, power consumed. The input parameters can be classified as following:

* Fixed parameters related to the application: "executionTime" consumed by the actor in certain MicroBlaze (or number of the Actors that do the same function). "MemoryElementSize" of the data memory ".data" and code memory ".code". Tokens number of channels within the application "channelTokensNum", which expresses the relationship weight between two actors.

* Variable parameters related to network scheme: the number of nodes "tilesNum", and their topology "NetworkTopology". Certain link delay "tilesConnectionDelay", work frequency within the network "networkFrequency", and "linkWidth". After making scenarios that include variable values of these parameters, the resulting chart, see Fig. 2.

We note, with increasing the number of tiles in the network scheme, the frequency decreases, because it is inverse proportionality to the time consuming during the transition. The area increases, but have little increase in the first (more than doubled), and then become a big increase (two and more). With this increase, latency reduced slightly, but after that, it increases significantly. For power consumption, it increases significantly even up to the point where getting a bit, the reason is the use of clock isolation, there is not additional tiles participated in the work.

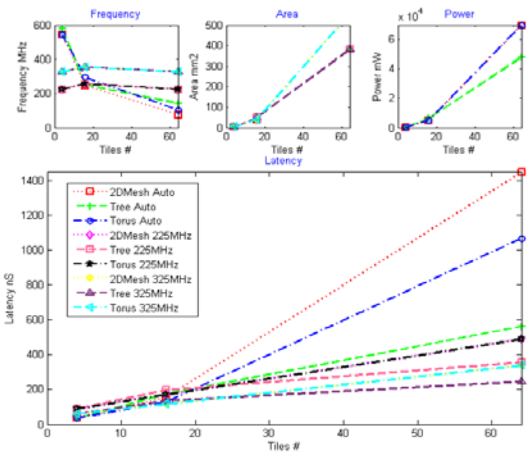

Fig. 2. System output related to network scheme and its parameters.

The chip area increases in the case of 2D-Mesh and torus linearly with the increase in the number of tiles. The latency decreases at first, but with the increase in area, it increases significantly. Power consumption and latency in torus are less than the case of the 2D-Mesh, the reason is the low maximum distance between tiles, because of the additional links on the edges in the network scheme. Area dramatically increased in the case of the tree, the reason is the design of a multi-level tree. For the latency and power consumption in the tree, the reason is bandwidth increasing in the main branches of the tree.

* Fixed and variable parameters related to network interface: the fixed includes, identify network interface model "niModel" that communicates with the PE. The variable includes, determine the number of input and output channels of the interface, "nrInputConnection" and "nrOutputConnection", as well as determine the bandwidth of input and output, "InBandwidth" and "outBandwidth". These parameters have been introduced into the scenarios, after choosing four schemes from the Fig. 2, these schemes are 2D-Mesh Auto, 2D-Mesh 325MHz, Tree Auto, and Torus 325MHz. And, it is given series of compound values for the previous parameters respectively (4,4,48,48 - 8,8,96,96 - 16,16,192,192 - 32,32,384,384); the interface bandwidth is proportionate with the number of input and output channels. Notice that, if the number of tiles is "4" as shown in Fig. 3, increasing in frequency, decreasing of latency and stabling in power consumption and area, with the increasing in input and output bandwidth of network interface. However, increased frequency stands at a value close to 600MHz when bandwidth close to 200 Mbps, as well as decreased latency stands at a value close to 30ns.

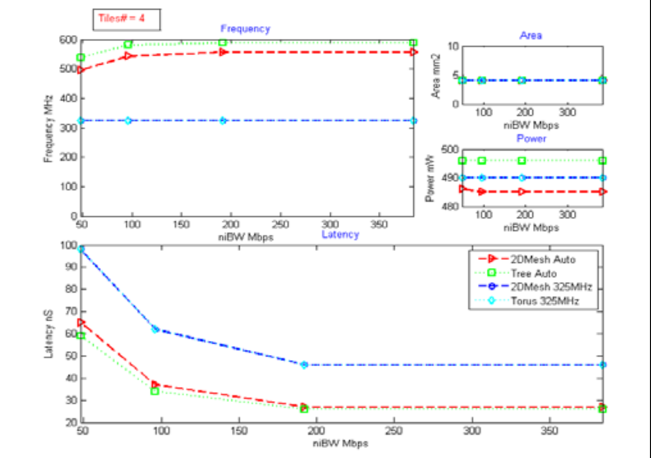

Fig. 3. System output related to network interface parameters, tiles=4

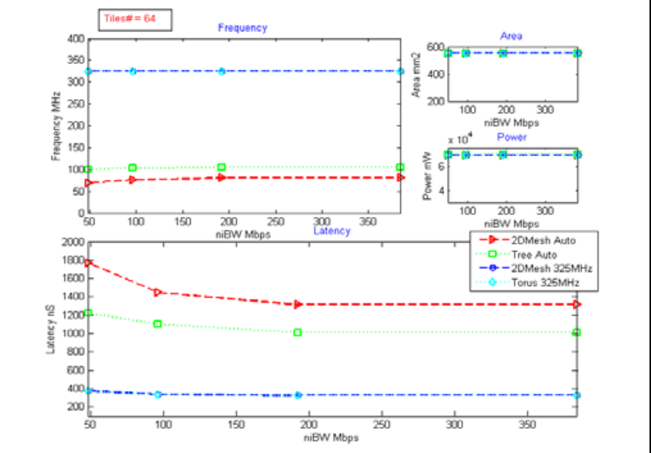

Fig. 5. System output related to network interface parameters, tiles=64

If the number of tiles is "16", as in Fig. 4, increasing in frequency, decreasing of latency and stabling in power consumption and area, with the increasing in input and output bandwidth of network interface. However, increased frequency stands at a value close to 200MHz when bandwidth close to 200 Mbps, as well as decreased latency stands at a value close to 150ns.

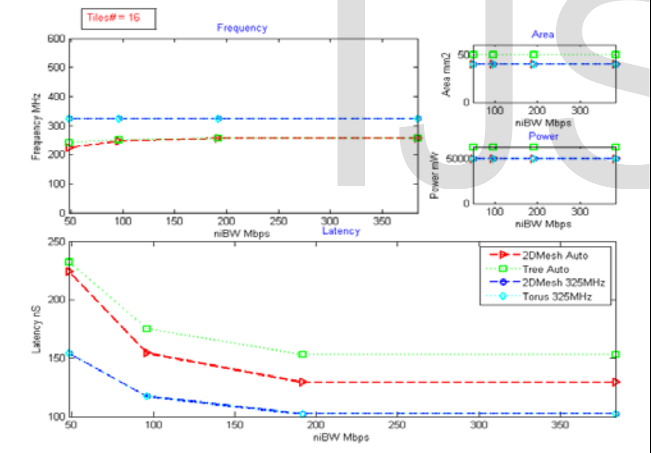

Fig. 4. System output related to network interface parameters, tiles=16

When tiles number is "64" as in Fig. 5, increasing in frequency, decreasing of latency and stabling in power consumption and area, with the increasing in input and output bandwidth of network interface. However, increased frequency stands at a value close to 50MHz when bandwidth close to 200 Mbps, as well as decreased latency stands at a value close to 1400ns.

* Fixed and variable parameters related to MicroBlaze PE: identify the arbitration type of the scheduler "arbitrationMcroBlz" inside Microbalze. The variable parameters are determining the "wheelsize", and the size of the data and instruction memories, "dmemSize" and "imemSize".

These parameters have been introduced into the scenarios, after choosing four schemes from the Fig. 3, these schemes are 2D-Mesh Auto, 2D-Mesh 325MHz, Tree Auto, Torus 325MHz. And, it is given compound values for network interface (8,8,96,96), because the charts in Fig. 3. 4. 5 show that these values are what make the system outputs in case of changing. It has been given multiple values for the parameters of the MicroBlaze and after noticing the results, it selects "wheelsize=1000" and the size "64KB" for the memories in the next scenarios.

* Fixed and variable parameters related to the router: The type of coherence protocol "coherencyProType", the design of input buffers and registers "inpuBufferDesginType" "registerType", determine the protocol to avoid a deadlock "deadlockAvoidPro", flow control protocol "flowControlPro". Routing algorithm "routingAlgo", the switching mechanism within the router "switchAllocate", the type of arbitration within the router "routerArbitration", a mechanism to reduce power consumption "powerSaving".

The variable parameters are the number of ports of the router "routerPorts", the number of virtual channels for each physical link "VCsNum", "bufferSize", "flitSize", "linkWidth", and the size of the main packet "packetSize". The number of router's ports relies on the network scheme, in the case of the tree, the number is four, while in the 2D-Mesh and the torus, the number is five. To determine the number of VCs, many experiments have been done using different values of VCs (2, 5, and 10). Notice from the results, the use of values (2, 5) more effective. The last four parameters have been given series of compound values for respectively (6,3,3,6 - 15,3,3,15- 15,5,5,15 - 9,3,3,64). Each of previews compound values is considered as a model of router architecture. The model (15, 3, 3, 15) represents VCTlite [4].

Notice that in [4], it was planned to use 2D-Mesh network scheme only, while this research includes an implementation of VCTlite architecture and protocols and install it in our laboratory environment, in addition to input new parameters

and schemes. The other models are contributions of the research after doing many experiments.
If the number of tiles is “4” as in Fig. 6, notice that the model RTR155515 that implemented in tree scheme and two VCs achieves an improvement in frequency while maintaining the area and latency, but with a small increasing in power consumption. In the three models, which implemented in tours scheme with frequency lock and five VCs, notice improving the latency and small increasing in power consumption.

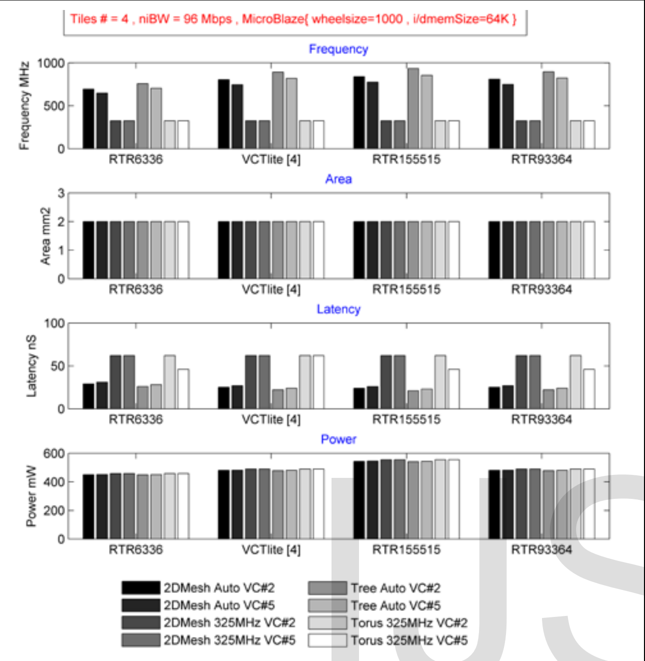

Fig. 6. System output related to PE and router parameters, tiles=4

Note if the number of tiles is “16” as in Fig. 7. The three models, which implemented in 2D-Mesh scheme with frequency lock and five VCs, achieve an improvement in latency while maintaining of area and power consumption. However, there is little increase in power consumption only in RTR155515 model.

Note if the number of tiles is “64” as in Fig. 9, the model RTR6336 that implemented in tree scheme and two VCs achieves an improvement in frequency, area, latency, and power consumption. In addition, the same model implemented in torus scheme with frequency lock with different VCs’ number achieves an improvement in area and power consumption while maintaining the frequency and latency.
.

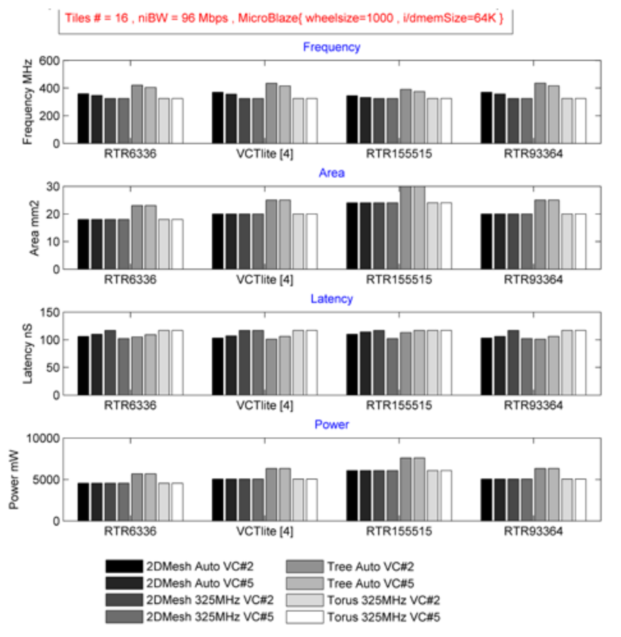

Fig. 7. System output related to PE and router parameters, tiles=16

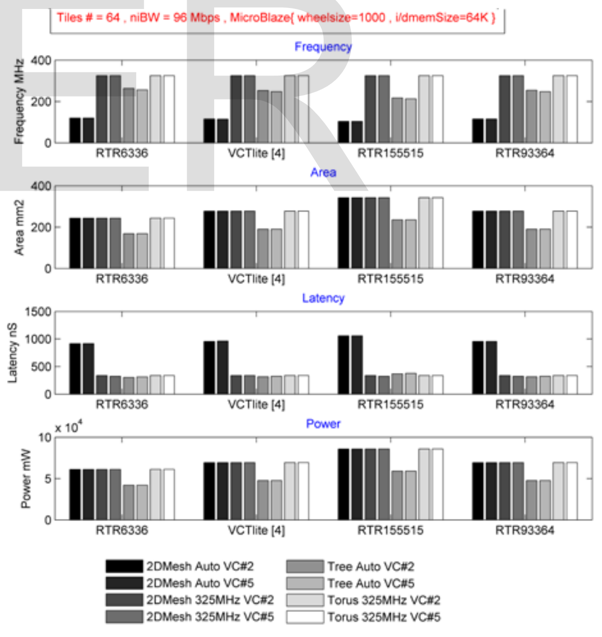

Fig. 8. System output related to PE and router parameters, tiles=64

4 Conclusion

It is prepared laboratory environment, which enables to build any hardware emulator. By using the tools MAMPS, SDF3, and RTRNoC, we can provide all the necessary sources and descriptions files that reflect certain hardware for analyzing through any development platform, such as Xilinx. In addition, we provide a design and an implementation for an application "RTRpktgen", which generates packets, and designed to fit MPSoC architecture that supports NoC.

Through this environment, it has been working on a wide range of scenarios, which depend on each other in order to achieve a better evaluation and optimization for the parameters of the virtual switching system of on-chip networks.

A set of conclusions has been reached. It can improve the frequency in a few tiles network scheme by increasing the buffer size within the router, being equal to the size of the main packet, being greater than twice of the size of the Flit, and using this combination with the tree unlocked frequency and a small number of VCs. For improving the delay, we can use any combination of router parameters with the torus locked frequency.
By increasing the number of tiles to average limit, we must go to the 2D-Mesh unlocked frequency with average number of VCs, to achieve improvement on the frequency, latency, and power consumption. By dramatically increasing the number of tiles, it is necessary to reduce the size of the buffer, to be equal to the size of the main packet, also to be equal to twice of the size of the Flit, and to use this combination with the tree unlocked frequency and a small number of VCs. To achieve improvement in area and power consumption also follow the previews structure but with the torus locked frequency.

ACKNOWLEDGMENT

This work was supported by Tishreen University, faculty of Information Engineering, department of Computer Systems and Networks Engineering. Also thanks to Dr. Akash Kumar, PhD (TU Eindhoven/NUS); 2009 for online helping.

REFERENCES

[1] L. Rizzo, M. Landi, Netmap: Memory mapped access to network devices, *SIGCOMM Comput. Commun. Rev.* Vol. *41*, n. 4, pp. 422-423, 2011.

[2] J. Bardgett, C. C. Zou, nswitching: *Virtual machine aware relay hardware switching to improve intra-nic virtual machine traffic*, Proceedings of *IEEE International Conference on Communications* (Pages: 2700-2705 Year of Publication: 2012 ISBN: 978-1-4577-2052-9).

[3] A. B. Achballah, S. B. Saoud, A survey of network-on-chip tools, *International Journal of Advanced Computer Science and Applications IJACSA*, Vol. *abs/1312.2976*, pp. 4-9, 2013.

[4] A. Roca, J. Flich, F. Silla, J. Duato, *Vctlite: Towards an efficient implementation of virtual cut-through switching in on-chip networks,* Proceedings of *International Conference on High Performance Computing, HiPC* (Pages: 1-12 Year of Publication: 2010).

[5] V. Surabhi, *Designing with sr-iov capability of xilinx virtex-7 pci express gen3 integrated block,* Report in Xilinx, Inc, 2013. URL: http://www.xilinx.com/support/documentation/application_notes/xapp1177-pcie-gen3-sriov.pdf

[6] A. B. Achballah, S. B. Saoud, The design of a network-on-chip architecture based on an avionic protocol, *International Journal of Advanced Computer Science and Applications IJACSA*. Vol. *abs/1401.4891*, 2014.

[7] S. Ma, Z. Wang, Z. Liu, N. D. E. Jerger, Leaving one slot empty: Flit120 bubble flow control for torus cache-coherent nocs, *IEEE Trans. Computers.* Vol. *64*, n. 3, pp. 763–777, 2015.

[8] R. Jordans, F. Siyoum, S. Stuijk, A. Kumar, H. Corporaal, *An automated flow to map throughput constrained applications to a mpsoc,* Proceedings of *Bringing Theory to Practice: Predictability and Performance in Embedded Systems, DATE Workshop PPES.* (Pages: 47–58 Year of Publication: 2011).

[9] S. Fernando, A. Kumar, H. Corporaal, *Mampsx: A design methodology for rapid system-level exploration, synthesis of heterogeneous soc on fpga.* In Eindhoven University of Technology and National University of Singapore, 2012.

[10] A. H. Ghamarian, M. C. W. Geilen, S. Stuijk, T. Basten, B. D. Theelen, M. R. Mousavi, A. J. M. Moonen, M. J. G. Bekooij, *Throughput analysis of synchronous data flow graphs*, Proceedings of *the Sixth International Conference on Application of Concurrency to System Design, ACSD, IEEE Computer Society* (Pages: 25–36 Year of Publication: 2006 ISBN: 0-7695-2556-3).