# Hardware Implementation of Link Aggregation in Networks-on-Chip

Ievgen Korotkyi, Oleksandr Lysenko
Department of Design of Electronic Digital Equipment
National Technical University of Ukraine "Kyiv Polytechnic Institute"
Kyiv, Ukraine
korotkiy.eugene@ieee.org, o.lysenko@kpi.ua

***Abstract*—The link aggregation (LAG) technique for networks-on-chip (NoC) is described and investigated in the paper. It is shown that LAG permits to improve considerably the NoC saturation threshold due to connection of neighboring routers with the aid of multiple physical links. The proposed work has three main contributions. The first is the description of a structure and principle of operation of a NoC with LAG. The second is the comparative analysis of the synthesis results for Stratix IV FPGA. It is shown that hardware costs of LAG and virtual channel (VC) routers are comparable. The third is the evaluation of average latency and saturation threshold in LAG NoC (8x8 mesh). The simulation of System Verilog models indicates that saturation threshold in proposed approach increases by 152% compared to VC NoC.**

***Keywords: network on chip; link aggregation; router; synthesis;***

## I. INTRODUCTION

One of distinctive features inherent in present-day systems on chip is tendency to a continuous increase in the number of IP cores and in growing requirements to the amount and capacity of intermodular connections [1]. The former ways to organization of the communication subsystem, such as "common bus" or "fully linked architecture", gradually loose their effectiveness because of complication of SoC layout [2], [3]. To cope with the problem of data exchange within VLSI circuits, the concept of network-on-chip (NoC) has been suggested [4]. This approach incorporates all the advantages of scalability, parallelism, and high clock frequency [1]. The principles of NoC's are well described in reviews [5]-[8].

The most important performance metrics of NoCs are latency and throughput. Latency it is the time, which elapsed from the moment of packet creation until the moment of it receiving at the destination node, including the queuing time at the source. Throughput is the maximum traffic accepted by the network, where traffic accepted is the amount of information delivered per time unit. Throughput is limited by the saturation threshold – the value of accepted traffic when NoC become congested and latency increases in tenth of times.

Short input queues in NoC routers make it unreasonable to buffer the whole packet before sending it to the destination port [1]. Instead, the wormhole flow control is widely used, when the packet is partitioned into "atomic" flow control units (flits), transferred continuously one after another [9]. The flits are advancing as far as possible, without waiting for arrival of followers. It permits to soften requirements to the buffer space volume, but increases dramatically the probability of head of line blocking (HOLB). As shown in [10], this phenomenon may reduce NoC throughput to 50% of the network capacity.

An efficient technique permitting to lower HOLB probability is the use of virtual channels (VC) [10]. Particularly, a physical link is associated with several VCs, i.e. queues working in parallel. Blocking of one VC does not affect the transfer of packets over other logical channels, thus permitting to avoid HOLB and the intrinsic increase of latency. Unfortunately, this approach is not free from its own drawbacks. The first one is its poor scalability. Since the throughput of a physical link is distributed between virtual channels involved, the period of logical channel flits transmission grows together with the number of VC used. The second disadvantage is related to the need of VC allocation operation, which leads to an increasing of VC router's transfer latency by at least one clock cycle compared to it wormhole counterpart [11].

The purpose of this work is description and investigation of a link aggregation (LAG) method for NoCs (suggested by the authors), when for connection of routers, instead of a single physical link, we use several ones, each being employed for transmitting flits from different packets. This approach makes it possible to raise the NoC saturation threshold by 152% compared to networks based on VC routers. The results of a synthesis shows that hardware costs for the routers with LAG are comparable to their VC counterparts.

The paper is organized as follows. Section II is devoted to comparative analysis of architectural decisions used in creation of routers for NoC in the past. In Section III we consider the suggested structure of a router with LAG. In section IV we present results of a synthesis of different modifications of a LAG and VC routers. In Section V we use computer-aided simulation to evaluate dependence of the NoC saturation threshold on the number of physical links in aggregated logical channels. Also, here we perform comparative analysis of the obtained results against the background of already existing NoC solutions. The last section contains conclusions and proposals for further investigations.

## II. RELATED WORK

Here we offer a brief overview of architectural and design solutions used in creating state of the art NoC routers.

### A. *Wormhole router*

The block diagram of wormhole router is shown in Fig. 1 (a), whereas idea of this flow control method is considered in [9]. As can be seen from figure, wormhole router contains P input-output ports, connected to corresponding physical links (PL). Each PL is associated with FIFO queue. The operation principle of wormhole router is as follows. Since it does not always happen that the incoming flits will be processed immediately, they have to be queued and served one after another. For each packet, located at the head of every input queue, a sequence of steps will be performed: routing, switch allocation and crossbar traversal. The above scheme permits to pipeline the data processing. The latency of this type of router is usually within 3 clock cycles. The advantage of this approach is minimum hardware necessary for its implementation. The main drawback is low throughput because of HOLB [10]. Today the wormhole routers are used in systems with severe requirements to hardware volume, for example, in NoC's on FPGA basis [12], [13].

### B. *VC router*

To decrease the probability of HOLB, Dally [10] proposed a structure of VC router, whose block diagram is shown in Fig.1(b). Here each input port of the router has several queues (VCs) corresponding to a single physical connection. In the event of blocking of some VC, the incoming packets use any of free VC, thus diminishing the probability of HOLB and raising the network saturation threshold. The pipeline of VC router includes all the stages of wormhole router and in addition it contains one new stage which is responsible for VC allocation. This leads to additional NoC latency at least by one clock cycle [11]. Compared to wormhole structure, VC routers demand for somewhat extended hardware because of presence of input port multiplexers and of the VC allocation stage. It has been shown in [14] that doubling the number of VCs results to hardware increase by 100%. Nevertheless, VC are employed in NoC routers such as Hermes VC [14], MocRes [15], Aethereal [16], Netmaker [17]-[19], Dally and Peh VC router [11].

Analysis of NoC routers presented in [11], [14]-[15], [18] permits us to conclude that application of VC raises the saturation threshold no more than by 20% of NoC capacity compared to design without VC (supposing constant buffer space per port). There is some limit, when NoC saturation begins to depend weakly on further increase in number of VCs. In [14] we can see that for realizations with 2 and 4 VCs the saturation threshold remains practically unchanged, making up 23% and 25% of network capacity. In other words, an increase in number of VC by 100% lifts the throughput only by 2%. This is definitely low efficiency. In our opinion, this phenomenon may be explained by the fact that the throughput of each physical connection is divided between VCs involved, which in turn leads to an increase in the period of flits transmission in logical connections. So we come to a conclusion: as the number of VC increases, single physical connections between routers become a bottleneck and hamper further improvement of NoC saturation threshold and throughput.

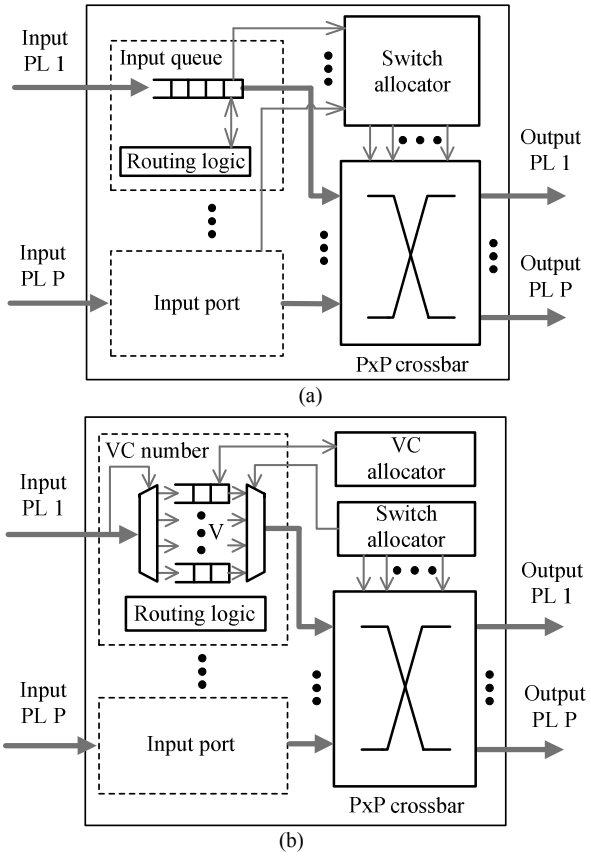


Figure 1. Block diagrams of wormhole (a) and VC router (b). Thick lines show the direction of data transfer. PL – physical link. VC – virtual channel. P – number of input-output ports. V – number of virtual channels.

### C. *VC router with speculative architecture*

The flow control based on VC application in its "classical" version implies sequential execution of operations concerning VC allocation and switch allocation. However, latency of the NoC router can be diminished if these procedures are being performed simultaneously. Parallel architecture of this type, suggested by Peh and Dally [11], makes it possible to reduce duration of packet head flits processing by 1 clock cycle. Authors of [11] called such an approach "speculative", because we speculate that a waiting packet will successfully be allocated an output VC when generating requests to switch allocator.

A disadvantage of this structure consists in lowering probability of successful completion of both operations [11]. In order to avoid reduction of throughput, the router is equipped with two modules responsible for switch allocation: the first one dealing with speculative requests (from packets without allocated VC), while the second — non-speculative requests (from packets still without access to the switch but with allocated VC). Such a decision reduces the average delay of the router but inevitably leads to an increase in hardware. Apart from [11], a similar VC router with speculative architecture has been realized within the framework of Netmaker (open source System Verilog library for NoC design and simulation)

[17]-[19]. The works [11], [18] testify that the use of speculative approach permits to lower the NoC transport delay by 5–20% against classical VC architecture, but hardly affects the saturation threshold. The reason is that the main problem of VC technology, related to competition of virtual flows for access to a single physical link, remains unresolved.

## III. NoC ROUTER WITH LINK AGGREGATION

As shown above, division of a physical communication line into logical paths, adopted in VC technology, has its pros and contras. Having applied the so-called inventive method of inversion [20], the authors of the present work suggested an "inverse" concept — joining several physical links (PL) in an aggregated logical channel (trunk). In other words, in order to link topologically adjacent routers, we offer to take several spaced channels, each being used for transmission of flits belonging to different packets. Fig.2 illustrates that aggregation of three connections eliminates HOLB at the western input of router $R_{22}$. Later we may see that even a small number of aggregated channels helps diminish HOLB and raise NoC saturation threshold considerably. Moreover, if we know spatial distribution of data flows in NoC, we can completely eliminate HOLB by properly adjusting the number of channels in each trunk.

Analysis of literature shows that aggregation of physical channels has its applications in macro-networks, and some documents [21], [22] standardize this approach for Ethernet. Since the design decisions used in macro-networks are hardly applicable to NoC because of their complexity and hardware volume, we are compelled to seek some simple and efficient mechanism for LAG in networks on chip. The authors propose one of possible solutions to this task.

The block diagram of a NoC router with the LAG is shown in Fig.3. To describe the inputs-outputs of this device we use the term "trunk", whose meaning has been considered earlier. The suggested decision includes M trunks, each containing N physical links. The connections between PLs are established by MNxMN crossbar. The flits arriving at each physical link are queued, since in the event when the number of PL is less than that of logical flows passing over the trunk, we are not sure in immediate servicing of the input request.

The data path in the suggested router takes form of a two-stage pipeline. For the head flit the pipeline latency is at least two clock cycles and for all consequent flits until the end of the packet this latency is at least one clock cycle. At the first step the operations of routing and PL allocation are performed. During the second step, flit is extracted from the queue and traversed through the crossbar and inter router connection to the next network node. Reliable data delivery is attained by "credit-based" flow control [23]. Now consider the above operations in more detail.

### A. Routing

In order to reduce the device latency we use the "look-ahead" routing [23], when the destination trunk for the current node has been determined in the previous router. This expedient makes it possible to start establishment of connection immediately after arrival of the first flit of a packet, and

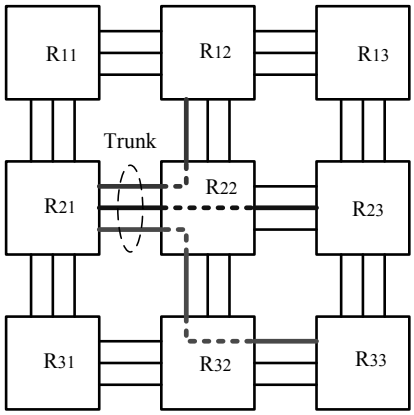


Figure 2. Eliminating HOLB in NoC by aggregation of 3 physical links

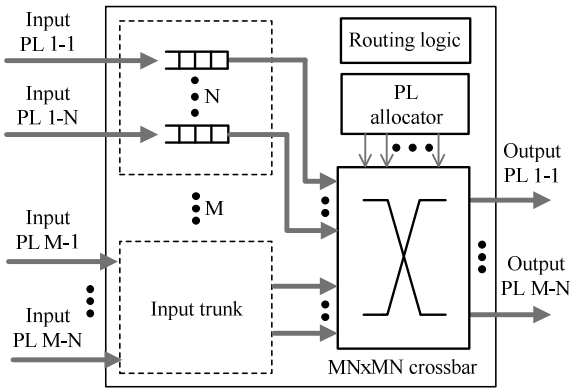


Figure 3. Block diagram of NoC router with link aggregation. PL - physical link. M - number of input-output trunks. N - number of physical links per trunk.

simultaneously determines the destination trunk for next router. The term "destination trunk" means aggregated logical connection used for accepting the flits of our packet. The routing information for the current router is confined in the head flit of the packet.

### B. PL allocation and switching

To advance the data over the router we have to link PL at the input with one of PL at the output. In our device the connection is set up upon arrival of the head flit and disappears after sending the tail part of the packet. At the cutoff instant the output physical channel becomes free and again ready to accept any other input PL, that winning arbitration. For each incoming packet is made an attempt to allocate an unused PL in the destination trunk by sending request to the PL allocator (Fig.4). In a case of successful PL allocation, mentioned request is cancelled and connection of the respective physical links is performed by the crossbar. The number of request inputs in the PL allocator equals to quantity of input PLs in router and each such an input consists of request line and destination trunk number bus. As can be seen from Fig.4, PL allocator includes two levels of arbitration. For each input PL, the correspondent first level arbiter performs preliminary selection of one free PL in the destination trunk. The status of output PL (busy/free) clears to 0 (busy), when this link is allocated to any input PL and sets up to 1 (free) when the tail flit leaves the router by busy output PL. The information about status of PLs in required output trunk delivered to correspondent first stage

arbiter by dedicated multiplexer (one per arbiter) according to destination trunk number for this request input. The outputs of the first level arbiters are connected with the aid of demultiplexers to the inputs of the second level arbiters. The k-th output of the first level arbiter corresponds to the p-th input of the q-th second level arbiter. The values of p and q can be obtained with the help of the following expressions:

$$p = N \cdot i + j\,, \quad q = N \cdot t + k\,,$$

Where values of *i* and *j* denote the numbers of input trunk and input PL of the corresponding first level arbiter. *N* characterizes the count of PLs per trunk. Values of *t* and *k* correspond to the destination trunk number and output PL number (which is previously allocated by the first level arbiter). It is possible that two or more first level arbiters choose the same output PL and output link will be contested by several inputs. Therefore, the final selection of input PL, which will be connected to the particular output, is made at the second level of arbitration, consisting of one arbiter per output PL. The active output of second level arbiter indicates the number of input PL to which particular output PL is allocated. All the arbiters are realized based on the matrix method, providing a better throughput compared to round-robin solution [24].

### C. *Flit advancing*

At the second stage of pipeline, all of the per-packet processing is complete and all remaining control is the flit-by-flit crossbar traversal and operations with credits. The head flit starts this process, but is handled no differently than any other flit. In order to pass a flit over the established route, two conditions are to be satisfied: the data available in the respective queue at the input and a non-zero number of credits at the router output. Then the flit will be extracted from the queue and, by the established connection, comes to the downstream router. At the same time a credit is directed to the preceding node of the network, meaning that in the input queue a new free place has appeared. Simultaneously, the number of credits for physical links with next router is decreased by one.

## IV. SYNTHESIS RESULTS

To obtain information about hardware resources needed to implement a NoC with LAG, we synthesized the several configurations of a router described in the previous section. As a synthesis tool we choose the Quartus 9.1 software and as the target FPGA we use Altera's Stratix IV chip. System Verilog was our hardware description language.

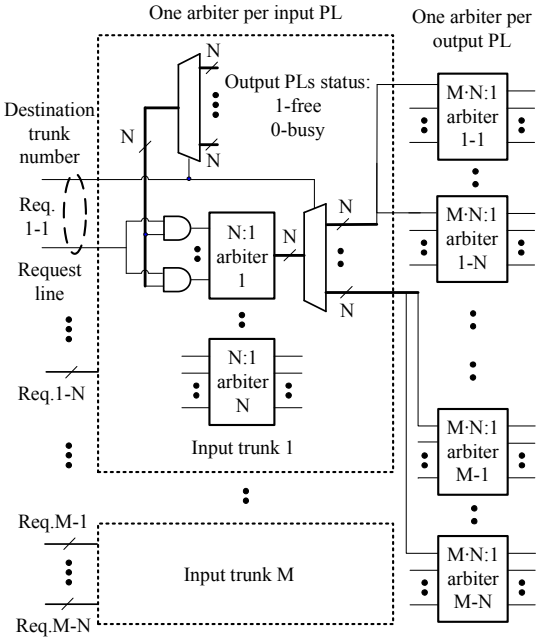


Figure 4. Block diagram of the physical links allocation module. PL – physical link. N – number of PLs per trunk. M – number of input-output trunks.

The results of the synthesis are present in the table 1. To compare our router with the counterparts we include into the table information about several VC routers. First of them is HERMES VC, described in [14]. Second is the VC router from Netmaker framework [17]. As the Netmaker has an open source code, we adopt it to be synthesized in the Quartus. In mentioned router the unrestricted VC allocation and switch allocation performed consequently in one clock cycle. More information about implementation of the Netmaker VC router can be found in [18]. In our router, the PL allocation is also performed in one clock cycle and if we'll pipeline this operation, maximum clock frequency grows up by 60 MHz. All the mentioned VC routers have 5 input-output ports and our LAG router has 5 input-output trunks.

For the purposes of the fair comparison, the bisection width of VC and LAG NoC should be equal. Hence the port width of VC router should be the same as the width of the trunk in a router with LAG. The width of PL inside the trunk will be determined by the following formula:

$$\text{PLwidth} = \frac{\text{Flit width in the comparable VC router}}{\text{The number of PLs within a trunk}}$$

TABLE I. THE SYNTHESIS RESULTS FOR DIFFERENT MODIFICATIONS OF LAG AND VC ROUTERS

| Router Type | FPGA device | Number of VC's | Number of PL's | PL width, bits | Port or trunk width, bits | Buffer length, flits | LUT's | REG's | Fmax, MHz |
|---|---|---|---|---|---|---|---|---|---|
| HERMES, VC | XC2V6000 | 2 | – | – | 16 | 8 | 1377 | 327 | – |
| Our design, LAG | EP4SGX230KF40C2 | – | 2 | 11 | 22 | 4 | 1900 | 894 | 200 |
| Netmaker, VC | | 2 | – | – | 22 | | 2179 | 1822 | 135 |
| Our design, LAG | | – | 2 | 16 | 32 | | 2167 | 1074 | 190 |
| Netmaker, VC | | 2 | – | – | 32 | | 2400 | 2232 | 130 |
| Our design, LAG | | – | 4 | 16 | 64 | | 7513 | 2571 | 140 |
| Netmaker, VC | | 4 | – | – | 64 | | 6971 | 6794 | 95 |

According to design of our router, the flit width can not be less than 11 bits, so the trunk width in the router with 2 PL's will be 22 bits. The flit width for HERMES VC is 16 bit and can not be changed, because this NoC is not open source. This fact is to our regret violates the requirement for equality of port and trunk width (but for the Netmaker there is no problem).

From the table 1 it can be seen that the hardware costs of our and VC routers are comparable. Let's analyze, why the growing of PLs number in the LAG router do not increase very much it hardware resources versus VC approach. First, the absence of input VC multiplexers in our router contributes to reducing the fraction of hardware costs. Second, in LAG router there is no need in storage of VC numbers within flits, which permits to diminish the volume of service information transmitted. Thus we can shorten the width of inter-router links and, hence, also decrease the hardware costs. Third, to establish a connection between input and output in a typical VC router, we have to perform actions of virtual channel and switch allocation. In our device there is no need in switch allocation, which consequently decreases the hardware costs. And the last, but not least, the flit width in NoC with LAG will be less than in analogous VC NoC due to requirement for equal bisection bandwidth. This circumstance is also reduces the crossbar size and ensures that wire capacitance of interconnections between LAG routers will be the same as for VC NoC.

## V. SIMULATION RESULTS

In this section we estimate the latency-throughput characteristics of the NoC with LAG by performing it RTL simulation. RTL simulation versus gate-level one was chosen due to better speed of operation. Based on System Verilog model of the router with LAG we create 8x8 mesh network. Every node in this network, with the aid of four bilateral trunks, was connected to neighboring routers in conformity with selected topology. Particularly, each router had 5 input-output trunks. The fifth pair of trunks was used for coupling to IP-core of traffic generator-receiver, imitating the functionality of the computational module. Simulation was conducted in ModelSim 6.5 medium on a PC with Windows XP installed.

For the standardization purposes is chosen a simple dimension ordered XY routing, because almost all NoC benchmarks uses it. During one simulation run, the traffic generators create packets consisting of flits and insert them into the network with intensity of λ (offered traffic). Time intervals between flits insertion are roughly equal to 1/λ and distributed by exponential law. In order to preserve the time distribution characteristics, the packets, prior to their insertion, form a queue at the source [23]. To describe the spatial distribution of messages we took the uniformly random law, when each source is equally likely to send data to each destination (except sending to itself). For each next packet its own new destination point will be generated. After starting the simulation, all nodes in the network generate 1100 packets each, and pick-up of the results begins after receiving the first hundred of messages. This scheme is adopted to put NoC in steady-state condition [23]. The simulation is considered terminated after registration in destination points all messages sent.

Fig. 5 shows the dependencies of NoC latency-throughput characteristics from the offered traffic and router architecture parameters. In order to plot any of the curves depicted, we have to perform several times the simulation of NoC functioning. To automate the process, we have written a script in Cygwin 1.7.7

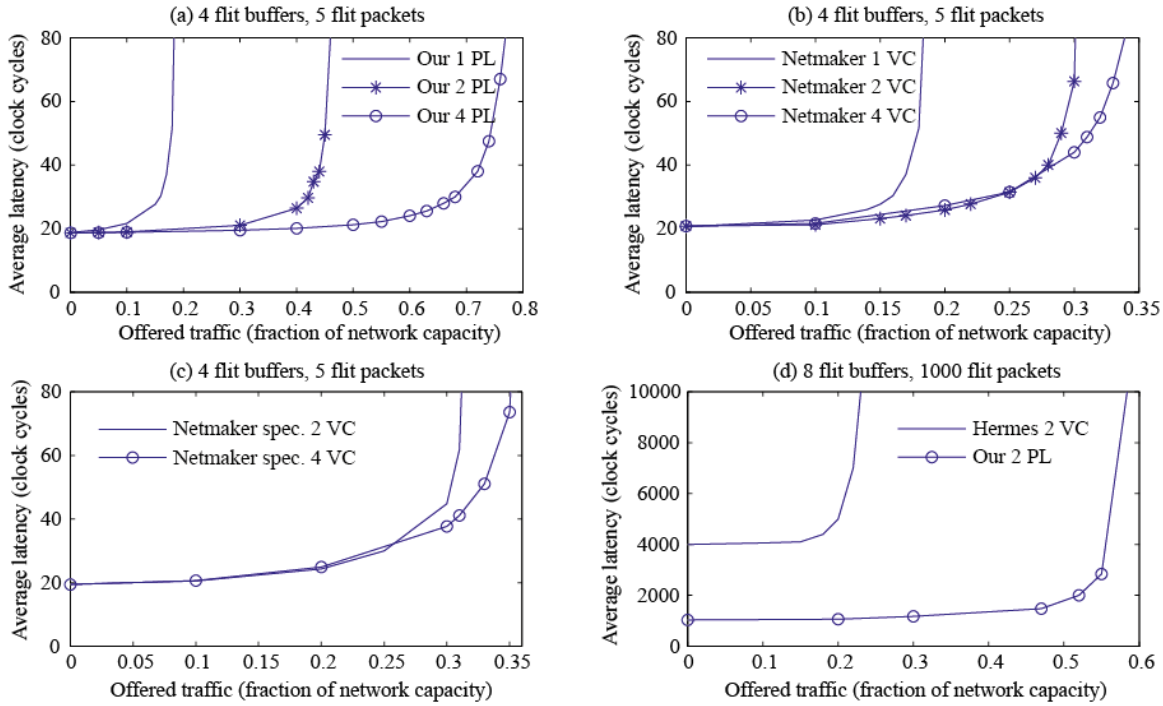


Figure 5. The dependency of NoC latency from offered traffic. All results are obtained for NoC with 8x8 mesh topology. (a) Proposed NoC with link aggregation. (b) Netmaker NoC with baseline VC routers. (c) Netmaker NoC with speculative VC routers. (d) The comparison between VC Hermes and our NoC with link aggregation. VC – virtual channel. PL – physical link.

medium permitting to execute several simulation runs for different values of the controlled parameter.

Fig.5(a) illustrates variations of saturation threshold for NoC with LAG depending of the number of physical links. Simulation was conducted for a network with 4 flit queues and 5 flit packets. As can be seen, doubling the number of PLs results in doubling throughput, while the maximum of saturation threshold is attained for 4 PL and equals 0.77 network capacity. For comparison sake, Fig.5(b) depicts the same dependence for a NoC, which based on Netmaker v0.82 baseline VC routers [17]. Since Netmaker is open source, the datasets for plotted curves were obtained by the authors themselves. Here we can see that maximum value of saturation threshold in this case makes up 0.34 of the network capacity, which is by 126% less compared to NoC with LAG. Application of Netmaker router with speculative architecture [Fig. 5(c)] makes it possible to reduce the average latency by 5% and increase the network saturation threshold by 1% against the previous case. Again, in this situation the use of speculative architecture does not lead to considerable improvement of NoC saturation threshold.

Fig.5(d) illustrates the difference in latency-throughput characteristics between NoC with LAG and Hermes VC network on chip. The data for comparison and simulation conditions have been taken from [14]. The size of input queues is 8 flits, and the data packet length — 1000 flits. For such "heavy" load the network with LAG exhibits an increase in saturation threshold equal to 152% against its VC counterpart. In addition, the average value of network latency in author's version is 1034 clock cycles while for Hermes VC this parameter equals 4045. As can be seen, the LAG gives a four times decrease in NoC traffic delay.

As noted in the previous section, the bisection width of VC and LAG NoC should be equal. This rule is satisfied during RTL simulation. For example, if the flit width in the router with 4 VC is 64 bits, than the PL width in LAG router with 4 PLs per trunk is 16 bit.

## VI. CONCLUSIONS

Analysis of state of the art NoC VC routers shows that the bottleneck in their structure, limiting growth of throughput, consists in using a single physical channel for connection of network nodes. To overcome this difficulty, the authors of the present work suggest an approach with so-called link aggregation (LAG), when, instead of one, we use several physical lines for linkage of topologically neighboring routers, so that each line takes part in advancing flits of different packets. Thus we can decrease substantially and, in the event of a sufficient number of PLs, almost eliminate the head-of-line blocking phenomena. Computer-aided simulation shows an increase in saturation threshold for NoC with LAG by 125÷152% against it virtual channel counterparts. At the same time, the results of the synthesis shows that hardware costs of LAG and VC routers are comparable.

To reduce the hardware costs of the proposed approach, it seems expedient to investigate the case of uneven distribution of PLs quantity per trunk, when allocation of physical links to aggregated connections depends on traffic load.

## REFERENCES


[1] T. Bjerregaard, S. Mahadevan, "A survey of research and practices of network-on-chip," *ACM Comp. Surveys*, vol. 38, no. 1, pp. 1-51, 2006.

[2] F. Angiolini, P. Meloni, L.Benini "A layout-aware analysis of networks-on-chip and traditional interconnects for mpsocs," *IEEE Trans. on CAD of Integr. Circ. and Syst,* vol. 26, no. 3, pp. 421-434, 2007.

[3] H.G. Lee, N. Chang, U.Y. Ogras, R. Marculescu, "On-chip communication architecture exploration: A quantitative evaluation of point-to-point, bus and network-on-chip approaches," *ACM Trans. on Design Automation of Electronic Systems*, vol. 12, no. 3, pp. 1-20, 2007.

[4] W. Dally, B. Towles, "Route packets, not wires: on-chip interconnection networks," *in Proceedings of the 38th annual Design Automation Conference*, Las Vegas, USA, June 2001, pp. 684-689.

[5] D. Atienza, F. Angiolini, S. Murali, A. Pullini, L. Benini, G.D. Micheli, "Network-on-chip design and synthesis outlook," *Integration The VLSI journal,* vol. 41, no. 3, pp. 340-359, 2008.

[6] R. Marculescu, P. Bogdan, "The chip is the network: toward a science of network-on-chip design," *Foundations and Trends in Electronic Design Automation*, vol. 2, no. 4, pp. 371-461, 2009.

[7] R. Marculescu, U.Y. Ogras, L.S. Peh, N.E. Jerger, Y. Hoskote, "Outstanding research problems in NoC design: system, microarchitecture, and circuit perspectives," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 28, no. 1, pp. 3-21, 2009.

[8] Haiyun Gu, "Survey of dynamically reconfigurable Network-on-chip," *in Proc. of Intern. Conf. on Future Computer Sciences and Application,* Hong Kong, China, Jun. 2011, pp. 200-203.

[9] W.J. Dally, "Performance analysis of k-ary n-cube interconnection networks," *IEEE Trans. Comput.*, vol. 39, no. 6, pp. 775-785, 1990.

[10] W.J. Dally, "Virtual-channel flow control," *IEEE Trans. on Parallel Distrib. Syst.*, vol. 3, no. 2, pp. 194-205, 1992.

[11] L.S. Peh, W. J. Dally, "A delay model and speculative architecture for pipelined routers," *In International Symposium on High-Performance Computer Architecture*, Nuevo Leone, Mexico, Jan. 2001, pp. 255–266.

[12] F. Moraes, N. Calazans, A. Mello, L. Moller, L. Ost, "HERMES: an infrastructure for low area overhead packet-switching networks on chip," *Integration, the VLSI Journal*, vol. 38, no. 1, pp. 69-93, 2004.

[13] A. Ehliar, D. Liu, "An FPGA based open source network-on-chip architecture," *in Proc. of Fields Program. Logic and App. Intern. Conf.*, Amsterdam, Netherlands, Aug. 2007, pp. 800-803.

[14] A. Mello, L. Tedesco, N. Calazans, F. Moraes, "Virtual channels in networks on chip: implementation and evaluation on Hermes NoC," *in Proceedings of 18th Symposium Integrated Circuits and System Design*, New York, USA, 2005, pp. 178-183.

[15] A. Janarthanan, "Networks-on-chip based high performance communication architectures for FPGAs," Ph.D. dissertation, Dept. Elect. Comp. Eng. and Comp. Science, Univ. of Cincinnati, Ohio, 2008.

[16] E. Rijpkema, K. Goossens, A. Radulescu, "Trade-offs in the design of a router with both guaranteed and best-effort services for networks on chip," *in Proc. of DATE conf.*,Munich,Germany, Mar. 2003, pp.350-355.

[17] Netmaker, http://www-dyn.cl.cam.ac.uk/~rdm34/wiki.

[18] R. Mullins, A. West, S. Moore, "Low-latency virtual-channel routers for on-chip networks," *in Proceedings of 31-th International Symposium on Computer Architecture,* Munich, Germany, June 2004, pp. 188-197.

[19] R. Mullins, A. West, S. Moore, "The design and implementation of low-latency on-chip network," *in Proc. of 11-th Asia and South Pacific Des. Autom. Conf.,* Yokohama, Japan, Jan. 2006, pp. 164-169.

[20] G. Altshuller, *The innovation algorithm. TRIZ, systematic innovation and technical creativity*. Worcester, MA: Technical innovation center, 1999.

[21] *Link aggregation,* IEEE Standart 802.3ad, 2000.

[22] *IEEE Standart for Local and Metropolitan Area Networks – Link Aggregation*, IEEE Standart 802.1ax, 2008.

[23] W.J. Dally, B.P. Towles, *Principles and Practices of Interconnection Networks*. San Francisco: Morgan Kaufmann Publishers, 2004.

[24] Z. Fu, X. Ling, "The design and implementation of arbiters for networks-on-chip," *in Proceedings 2nd International Conference on Industrial and Automation Systems*, Dalian, China, Jul. 2010, pp. 292-295.