GLB - Efficient Global Load Balancing Method for Moderating Congestion in On-Chip Networks

Masoud Daneshtalab, Masoumeh Ebrahimi, Juha Plosila
Department of Information Technology, University of Turku, Finland
{ masdan, masebr, juplos }@utu.fi

Abstract— Network Congestion can limit the performance of NoC due to increased transmission latency and power consumption. In this paper, to reduce the network congestion, we present an efficient congestion-aware routing method, named Global Load Balancing (GLB). In GLB, packets aggregate and carry congestion information along a path they route. Consequently, this information contains a global view of the path from where the packet is routed. This global congestion information is used as a metric in the arbitration unit in order to give more priority to the packets arriving from congested area. GLB can efficiently balance the load across the links by steering traffic from highly congested regions to low congested areas. Experimental results with synthetic test cases demonstrate that the GLB method provides significant improvement in average network latency (>19%).

I. INTRODUCTION

As technology geometries have shrunk to the deep submicron regime, communication delay and power consumption of global interconnections in high performance Multi-Processor Systems-on-Chip (MPSoCs) are becoming a major bottleneck [1][2]. The Network-on-Chip (NoC) architecture paradigm, based on a modular packet-switched mechanism, can address many of the on-chip communication design issues such as performance limitations of long interconnects, and integration of high number of Intellectual Property (IP) cores on a chip [2-6].

In network-based multiprocessor architectures, network congestion affects the system performance considerably [7-10]. Several adaptive routing algorithms (output selection) [11-17] and arbitration techniques (input selection) [18-21] have been presented to deal with the network congestion problem. In adaptive routing algorithms, the path between a source and a destination is determined at each router depending on the network status as packets move toward the destination; this can improve the load balance across the links. The input selection chooses one of input channels to get access to the output channel, done by an arbitration process. The arbiter could follow either non-priority or priority scheme. In the priority method when there are multiple input port requests for the same available output port, the arbiter grants access to the input port having the highest priority level. The priority scheme can also flatten the network congestion by giving higher priority level to traffic coming from congested areas [19-22].

In this paper, we propose an efficient congestion-aware routing method for multiprocessor architectures, named Global Load Balancing (GLB). GLB method can significantly improve the performance of NoCs by employing the global congestion information of upstream routers in the arbitration unit of routers. The paper is organized as follows. In Section II, the related work is discussed. In Section III, the proposed Global Load Balancing (GLB) method is explained. The results are reported in Section IV, while the summary and conclusion are given in the last section.

II. BACKGROUND AND RELATED WORK

2D-mesh topology is a popular architecture for NoC design because of its simple structure, ease of implementation, and support for reuse [23]. The performance and efficiency of NoCs largely depend on the underlying routing methodology. Adaptive routing algorithms can be decomposed into routing and selection functions. The routing function supplies a set of output channels based on the current and destination nodes. The selection function selects an output channel from the set of channels supplied by the routing function [7]. The selection function can be classified as either congestion-oblivious or congestion-aware schemes [8][14]. In congestion-oblivious algorithms, such as Zigzag [24] and random [25], routing decisions are independent of the congestion condition of the network. This policy may disrupt the load balance since the network status is not considered. Unlike congestion-oblivious methods, in congestion-aware algorithms, such as DyXY [11], CATRA [14], odd-even [12], GOAL [9] and GAL [10], the selection is usually performed using the congestion status of the network [7][25]. Most of congestion-aware algorithms consider local traffic condition in which each router analyses the congestion condition of itself and adjacent routers to choose the output channel. D. Wo et al. presented the CAIS method in [20]. CAIS investigates the impact of input selection on the routing efficiency by taking into consideration the local congestion information of the upstream routers. Routing decisions based on local congestion information may lead to an unbalanced distribution of traffic load. Therefore, they are efficient when the traffic is mostly local, i.e. cores communicate with other ones close to them [27], but they are unable to solve the global load balance problem via making local decisions [8].

P. Lotfi-Kamran et al. introduced EDXY [28] where the congestion information of a router is propagated to its row and column nodes via separate wires. Non-local information provided by these wires is used only when the packet is one row or column away from destination. In NoP [29] the routing decision is performed based on the congestion information of the nodes within two hops of the current node that are located in the minimal path to the destination. A well-known method named Regional Congestion Awareness (RCA) is proposed in [8] to utilize non-local congestion information in routing decision. In the RCA method, in order to prepare global congestion value in routers, the locally computed congestion value of a router is combined with those global signals propagated from downstream routers and the newly-aggregated value is transmitted to the upstream routers and so on. However, all congestion-aware methods aim to solve global load balancing problem by improving routing decisions utilizing local or non-local congestion information. The algorithms mainly focus on routing packets through less congested paths and avoiding additional traffic to the congested area and thus balancing the distribution of traffic load among the network nodes. However, they cannot utilize the congestion information in the scheduling process of routers. In this paper, we utilize the global congestion information of upstream routers in the arbitration process of routers. In this way, traffic in a highly congested area can be smoothed by steering packets to the less-congested regions.

III. GLOBAL LOAD BALANCING (GLB) METHOD

A. Using congestion information in intermediate routers

The major parts of research works target to balance the traffic load by routing packets around the congested areas using either local or non-local congestion information while they do not consider the impact of the router arbitration in distributing the traffic. These methods cannot efficiently distribute the traffic over the network, for instance when some nodes are highly congested and the others are not, the traditional methods cannot help packets to leave the congested areas and steer traffic to the less congested ones. The idea behind the GLB method is to allow congested nodes to forward their buffered packets rapidly, which greatly diminish the overall blocking probability. Consider the example in Fig. 1(a) where a 4×4 mesh network with three congested nodes 5, 6 and 10 is illustrated. If a fair arbitration mechanism is used at the router 9, arriving packets from the congested routers 5 and 10 will have the same chance to win the arbitration as the packets from the routers 8 and 13. Accordingly, the router 9 can be a bottleneck for the packets leaving the congested area. This bottleneck problem can be resolved by giving higher priority to the arrived packets from the routers 5 and 10 to access the output ports at the router 9. In contrast, packets arriving from the routers 8 and 13 should wait in the input buffers of the router 9 before accessing the output channel which increases the congestion condition at the router 9 slightly. In other words, the traffic of highly congested areas is distributed over low congested ones, and thus alleviating the traffic load in the congested regions. Fig. 1(b) depicts the spread of traffic congestion over the network where packets arriving from congested nodes (i.e. nodes 5, 6 and 10) get more chance to win the arbitration among the neighboring nodes (i.e. nodes 1, 2, 7, 11, 14, 9 and 4), and similarly, the priority based arbitration is performed in the rest of the routers. If we assume that the congestion value of a router is determined by the congestion condition of the router as well as its neighbors, and this information is carried by packets, then packets are able to collect the congestion information of routers and their neighbors on the path from the source to destination. Since this value contains a global view of the routing path, it can be used as the priority parameter in routers to recognize the congested areas in the network. Thus, the router arbitration is performed based on the global congestion information carried by packets.

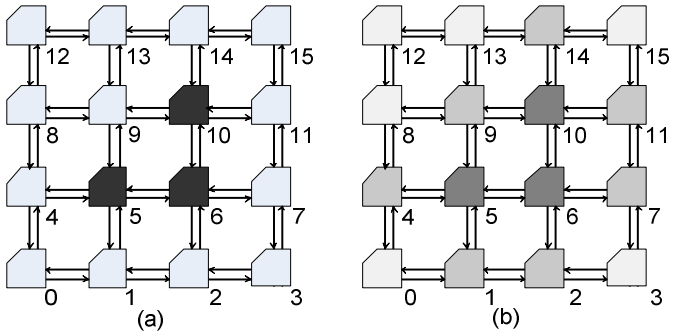

Fig. 1. Traffic distribution (a) without (b) with applying the idea of GLB.

B. Implementation of GLB

1) Adaptive Output Selection

As the routing function returns a set of admissible channels to send a packet, the selection function chooses one of them based on the local or non-local congestion information. In this paper, we employ Dynamic XY (DyXY) [11] routing algorithm which is an adaptive routing model without requiring virtual channels. The selection is made locally according to the congestion condition of the neighboring nodes. The number of occupied buffer cells at the corresponding input buffers of the neighboring nodes is considered as the congestion metric. Therefore, if the occupied space of input buffer is larger than a threshold value, then the congestion flag of the input port becomes '1', otherwise '0'. Note that for simplicity, we do not consider the non-local congestion information in routing decisions.

2) Adaptive Input Selection

We reserve a 4-bit field, named Congestion Status, in the header of each packet to store the congestion information of the path being traversed by the packet. Therefore, in each intermediate router, the local congestion status is combined with non-local information stored in the header of a packet, and then the header of the packet is updated with the new

congestion information. The congestion status of a router is determined based on the Congestion Conditions of the immediate neighbors and the router itself. Congestion Condition of a router and its neighbors are obtained according to Table 1 where x indicates the number of congested input ports out of the total number of input ports at a router; while y shows the number of congested neighbors to the total number of neighbors. The 4-bit local congestion value of a router is obtained by concatenating the 2-bit congestion information of the local router and the received 2-bit congestion values from the neighbors. The computed congestion value of the router should be combined with those of carried by packets. For this purpose, 50-50 weight is assigned to local and non-local congestion values. Finally the newly computed value is stored in the header of the packet.

Table 1. Summarizing congestion values of local and neighboring nodes into two bits.

x	CC
$0<x\leq 1/4$	00
$1/4<x\leq 1/2$	01
$1/2<x\leq 3/4$	10
$3/4<x\leq 1$	11

y	CC
$0<y\leq 1/4$	00
$1/4<y\leq 1/2$	01
$1/2<y\leq 3/4$	10
$3/4<y\leq 1$	11

Using this mechanism, packets can carry the congestion information of the passing routers and their neighboring routers along the path. Since this value contains a global view of the quadrant from where the packet is routed, it can be used as an efficient metric for the arbitration process of intermediate routers to recognize the congested areas in the network. The input selection function examines the priority value of all input packets and gives a grant to a packet with the highest congestion level. In order to prevent starvation, each time after finding the highest value, the priorities of defeated packets are incremented. Fig. 2 shows the pseudo code of input selection function.

```
i : i(th) input channel
C : congestion value of packet
W : waiting periods of packet
---------------
process input-selection-function is
begin
 for 'i=0 to all input ports' loop
 if "packet in input(i) is newly arrived"
   then
        W(i) <= 0;
     else
        W(i) <= W(i) + 1;
     end if;
     if W(i)+ C(i) > MaxPriority
      then
       MaxPriority <= W(i)+ C(i);
       select   <= i;
     end if;
  end loop;
end process;
```

Fig. 2 the pseudo code of the input selection function.

IV. EXPERIMENTAL RESULTS

To evaluate the proposed GLB method a NoC simulator is implemented with VHDL. The simulator models all major components of the NoC such as network interfaces, routers, and wires [32]. Moreover, we two on-chip networks are implemented using RCA and CAIS as the baseline schemes.

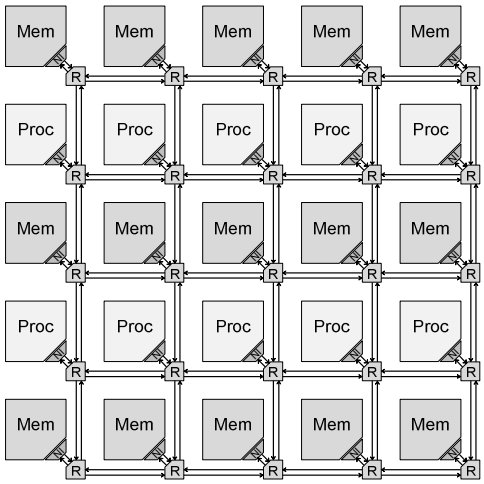

Fig. 3. 5x5 NoC layout.

A. System Configuration

We use a 25-node (5×5) 2D mesh on-chip network for the entire architecture. As illustrated in Fig. 3, out of 25 nodes, ten nodes are assumed to be processor (master cores, connected by master network interfaces) and other fifteen nodes are memories (slave cores, connected by slave network interfaces). The processors are 32b-AXI and the memories are DRAM (t_{RP}-t_{RCD}-t_{CL}=2-2-2, 32b). We adopt a commercial memory controller with memory interface, DDR2SPA module from Gaisler ip-cores [30] where it is placed between the memory and the slave network interface. In addition, each router structure including input buffers, a VC (Virtual Channel) allocator, a routing unit, a switch allocator and a crossbar. Each router has 5 input/output ports, and each input port of the router has 2 VCs. Packets of different message types (request and response) are assigned to corresponding VCs to avoid message dependency deadlock [31]. The arbitration policy of routers can be either round-robin or priority based scheme. The array size, routing algorithm, link width, number of VCs, buffer depth of each VC, and traffic type are the other parameters which are specified for the simulator. The routers adopt the Dynamic XY (DyXY) [9] routing and utilize wormhole switching. For all routers, the data width (flit size) was set to 32 bits, and the buffer depth of each VC to 5 flits. For the request, the command and all its control bits (flags) are included in the first flit of the packet, the memory address is set in the second flit, and the write data (in the case of a write command) are appended at the end. For the response message, the control bits are included in the first flit while the read data are appended at the end if the response relates to a read request.

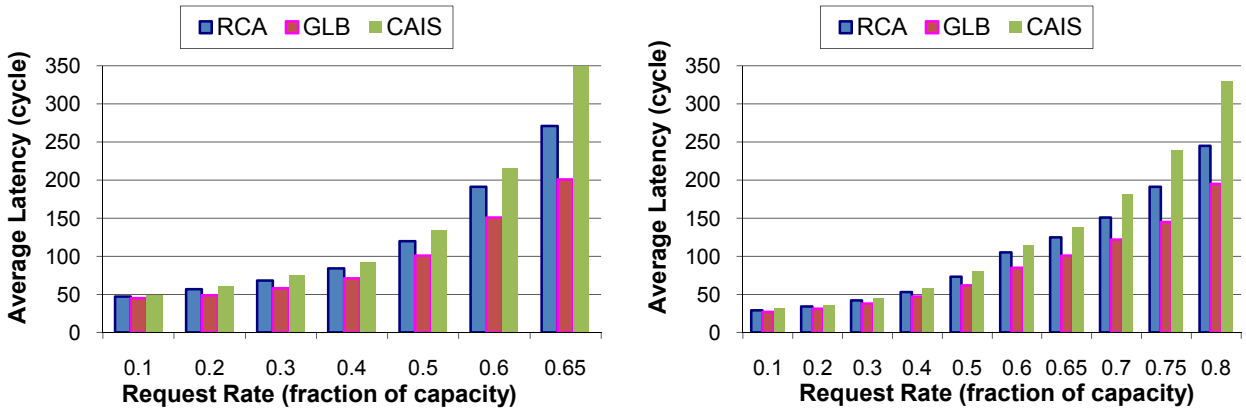

Fig. 4. Performance evaluation under (a) the uniform and (b) non-uniform traffic models.

Hence, the packet length for write responses and read requests is 1 flit and 2 flits, respectively, while the packet length for data messages, representative of read responses and write requests, is variable and depends on the write request/read response length (burst size) produced by a master/slave core. As a performance metric, we use latency defined as the number of cycles between the initiation of a request operation issued by a master (processor) and the time when the response is completely delivered to the master from the slave (memory). The request rate is defined as the ratio of the successful read/write request injections into the network interface over the total number of injection attempts. All the cores and routers are assumed to operate at 1GHz; and for fair comparison, we keep the bisection bandwidth constant in all configurations. All memories (slave cores) can be accessed simultaneously by each master core continuously generating memory requests. Furthermore, the size of the reorder buffer in each master core is set to 48 words. If the maximum burst size is set to 8, the master core can support at most 6 outstanding read requests in a 48-word reorder buffer (regardless of the exact size of the requests).

B. Performance evaluation

To evaluate the performance of the proposed schemes, uniform and non-uniform/localized synthetic traffic patterns are considered. These workloads provide insight into the strengths and weaknesses of the different buffer management mechanisms in the congestion-aware on-chip networks, and we expect applications stand between these two synthetic traffic patterns [34][36]. The random traffic represents the most generic case, where each processor sends in-order read/write requests to memories with a uniform probability. Hence, the target memory and request type (read or write) are selected randomly. Eight burst sizes, from 1 to 8, are stochastically chosen according to the data length of the request. In the non-uniform mode, 70% of the traffic is local requests, where the destination memory is one hop away from the master core, and the rest 30% of the traffic is uniformly distributed to the non-local memory modules.

Fig. 4 (a) and (b) show the simulation results under the uniform and non-uniform traffic models, respectively. In the presented configuration, the on-chip network utilizing the proposed GLB method is compared with the networks equipped with CAIS and RCA methods. As demonstrated in both figures, compared with CAIS and RCA architectures, the NoC using GLB reduces the average latency when the request rate increases under the uniform and non-uniform traffic models. The performance gain near the saturation point (0.6) under the uniform and non-uniform traffic models is about 21%~30% and 19%~26%, respectively. The reason for such an improvement is due to using the presented GLB method which can diminish the congested areas so that the average network latency decreases.

Table 2. Hardware costs.

Components	Area (mm^2)	Power (mW)
GLB Router	0.1873	64
CAIS Router	0.1913	66
RCA Router	0.2104	71

C. Hardware Cost

In this section, the hardware cost and the power consumption of the GLB scheme are computed. The on-chip router of each scheme (GLB, CAIS, and RCA) is synthesized with Synopsys Design Compiler using the 65nm standard CMOS technology with a timing constraint of 1 GHz for the system clock and supply voltage 1 V. The synthesized net-list is then again verified through post synthesis simulations. Finally, we perform place-and-route, using Cadence SoC Encounter, to have precise power and area estimation in wire-dominated structures. The power dissipation of each scheme is calculated under the uniform traffic model near the saturation point (in the aforementioned 5×5 2D mesh) using Synopsys PrimePower. The layout areas and power consumptions of presented schemes are listed in Table 2. As can be seen from the table, the router employing the GLB scheme is compared with other routers.

As the results, reported after place-and-route, the GLB router has lower area and power consumption in comparison with the CAIS and RCA routers.

V. SUMMARY AND CONCLUSION

A reordering mechanism is necessitated to handle deadlocks caused by memory parallelism in network-based multiprocessor architectures. In addition, network congestion is a critical issue in such architectures where processors communicate with memory modules through the on-chip network. In this work, we present a high performance dynamic reordering mechanism integrated in the network interface to improve the resource utilization. To manage the network congestion, an efficient method based on the global congestion information is presented. The micro-architectures of the proposed network interface architectures which are compatible with the AMBA AXI protocol are presented. A cycle-accurate simulator is used to evaluate the efficiency of the proposed congestion management method along with employing the presented network interfaces.

REFERENCES

[1] Y. Hoskote, et al., "A 5-GHz mesh interconnect for a teraflops processor," IEEE Micro, 27:51–61, September-October 2007.

[2] B.Towles and W.Dally, "Route packets, not wires: on-chip interconnection networks", Proc. DAC 2001.

[3] L.Benini and G.De Micheli, "Networks on chips: a new SoC paradigm", IEEE Computer, January 2002.

[4] T. C. Xu et al., "Optimal Memory Controller Placement for Chip Multiprocessor," In Proceedings of the 9th IEEE/ACM/IFIP International Conference on Hardware/Software Codesign and System Synthesis (CODES/ISSS), pp. 217-226, 2011, Taiwan.

[5] D. Bertozzi and L. Benini, "Xpipes: A Network-on-Chip Architecture for Gigascale Systems-on-Chip," IEEE Circuits and systems magazine, 2, p. 18-31,2004.

[6] T. C. Xu et al., "A Study of 3D Network-on-Chip Design for Data Parallel H.264 Coding," Journal of Microprocessors and Microsystems, Vol. 35, No. 7, pp. 603-612, October 2011.

[7] J. Duato, S. Yalamanchili, and L. Ni, "Interconnection Networks: An Engineering Approach." Morgan Kaufmann, 2002.

[8] P. Gratz, B. Grot and S.W. Keckler, "Regional Congestion Awareness for Load Balance in Networks-on-Chip", Proc. HPCA, pp. 203–214, 2008.

[9] A. Singh, W.J. Dally, et al.,"GOAL: A Load-Balanced Adaptive Routing Algorithm for Torus Networks", In International Symposium on Computer Architecture, pages 194–205, 2003.

[10] A. Singh, W.J. Dally, B. Towles, and A. K. Gupta, "Globally Adaptive Load-Balanced Routing on Tori" IEEE Computer Architecture Letters, v.3, I.1,pp.2-6, 2004.

[11] M. Li, Q. Zeng, W. Jone, "DyXY - a proximity congestion-aware deadlock-free dynamic routing method for network on chip", Proc. DAC, pp. 849-852, 2006.

[12] G. Chiu, "The Odd-Even Turn Model for Adaptive Routing," IEEE Tran. On Parallel and Distributed System, pp 729-738, July 2000.

[13] P. Lotfi-Kamran et al., "BARP-A Dynamic Routing Protocol for Balanced Distribution of Traffic in NoCs", In DATE conference, pp. 1408-1413, 2008.

[14] M. Ebrahimi et al., "CATRA ─ Congestion Aware Trapezoid-based Routing Algorithm for On-Chip Networks," in Proceedings of 15th ACM/IEEE Design, Automation, and Test in Europe (DATE), pp. 320-325, Mar. 2012, Germany.

[15] J. Hu and R. Marculescu, "DyAD-Smart Routing for Networks-on-Chip," DAC 2004, pp. 260–263, San Diego, California, USA, 2004.

[16] M. Daneshtalab et al., "A Generic Adaptive path-based routing method for MPSoCs," Journal of Systems Architecture (JSA-elsevier), Vol. 57, No. 1, pp. 109-120, 2011.

[17] M. Dehyadegari et al., "An Adaptive Fuzzy Logic-based Routing Algorithm for Networks-on-Chip," in Proceedings of 13th IEEE/NASA-ESA International Conference on Adaptive Hardware and Systems (AHS), pp. 208-214, June 2011, USA.

[18] C. A. Zeferino, M. E. Kreutz, and A. A. Susin, "RASoC: A Router Soft-Core for Networks-on-Chip", Proceedings of DATE'04, pp. 1530-1591, 2004.

[19] M. Daneshtalab et al., "Input-Output Selection Based Router for Networks-on-Chip" in Proceedings of 9th International Symposium on VLSI (ISVLSI), IEEE Press, pp. 92-97, July 2010.

[20] D. Wu, B. M. Al-Hashimi, and M. T. Schmitz, "Improving Routing Efficiency for Network-on-Chip through Contention-Aware Input Selection," In Proc. of 11th ASP-DAC, pp. 36 – 41, 2006.

[21] M. Daneshtalab et al., "Distributing Congestions in NoCs through a Dynamic Routing Algorithm based on Input and Output Selections," in Proceedings of 20th IEEE Conference on VLSI Design (VLSID), pp. 546-550, Jan 2007, India.

[22] M. Ebrahimi et al., "An Efficent Dynamic Multicast Routing Protocol for Distributing Traffic in NOCs," in Proc. of 12th IEEE/ACM Design, Automation, and Test in Europe (DATE), pp. 1064-1069, April 2009, France.

[23] J. Liang, S. Swaminathan, and R. Tessier, "aSOC: a scalable, single-chip communication architectures," in IEEE Int. Conf. on PACT, pp. 37–46, Oct. 2000.

[24] H.G. Badr, S. Podar, "An optimal shortest-path routing policy for network computers with regular mesh-connected topologies", v.38, I.10, pp.1362-1371, 1989.

[25] A. Sobhani et al., "Dynamic Routing Algorithm for Avoiding HotSpots in On-chip Networks," in Proceedings of 2th IEEE International Conference on Design & Technology of Integrated Systems in Nanoscale Era (DTIS), pp. 179-183, Sep 2006, Tunis.

[26] W. Feng and K. G. Shin, "Impact of Selection Functions on Routing Algorithm Performance in Multicomputer Networks", In International Conference on Supercomputing, pages 132–139, 1997.

[27] L. T. Thiago, R. d. Rosa, F. Clermidy, et al., "Implementation and evaluation of a congestion aware routing algorithm for networks-on-chip" pp. 91-96, 2010.

[28] P. Lotfi-Kamran et al., "EDXY - A low cost congestion-aware routing algorithm for network-on-chips", Journal of Systems Architecture, v.56, I.7, 2010.

[29] G. Ascia, V. Catania, M. Palesi, "Implementation and Analysis of a New Selection Strategy for Adaptive Routing in Networks-on-Chip", IEEE Transaction on Computers, v.57, I.6, pp. 809 – 820, 2008.

[30] Gaisler IP Cores, http://www.gaisler.com/products/grlib/, 2009.

[31] S. Murali et al., "Designing message-dependent deadlock free networks on chips for application-specific systems on chips," In Proc. VLSI-SoC, pages 158-163, 2006.

[32] M. Daneshtalab et al., "Memory-Efficient On-Chip Network with Adaptive Interfaces," IEEE Transaction on Computer-Aided Design of Integrated Circuits and Systems (IEEE-TCAD), Vol. 31, No. 1, pp. 146-159, Jan 2012.

[33] T. C. Xu et al., "Optimal Number and Placement of Through Silicon Vias in 3D Network-on-Chip," In Proceedings of the 14th IEEE International Symposium on Design and Diagnostics of Electronic Circuits & Systems (DDECS), pp.105-110, 13-15 April 2011, Germany.

[34] R. Das et al., "Design and Evaluation of a Hierarchical On-Chip Interconnect for Next-Generation CMPs," in proc. of 15th International Symposium on High-Performance Computer Architecture (HPCA), pp. 175-186, 2009.

[35] M. Ebrahimi et al., "Exploring Partitioning Methods for 3D Networks-on-Chip Utilizing Adaptive Routing Model," in Proceedings of 5th ACM/IEEE International Symposium on Networks-on-Chip (NOCS), pp. 73-80, May 2011, USA.

[36] P. P. Pande, C. Grecu, M. Jones, A. Ivanov, R. Saleh, "Performance Evaluation and Design Trade-offs for Network on Chip Interconnect Architectures," IEEE Transactions on Computers, vol. 54, no. 8, pp. 1025-1040, August 2005.