

Traffic-Aware Selection Strategy for Application-Specific 3D NoC

Sanaz Azampanah¹, Azadeh Eskandari¹, Ahmad Khademzadeh² and Fathollah Karimi³

¹CE Department, Science and Research Branch, Islamic Azad University, Tehran, Iran s.azampanaah@gmail.com, a.eskandari@srbiau.ac.ir

²Telecommunication Research Center, Tehran, Iran zadeh@itrc.ac.ir

³Islamic Azad University, Arak, Iran karimi.fathollah@gmail.com

Abstract

Three-dimensional stacking of silicon layers is emerging as a promising solution to handle the design complexity and heterogeneity of Systems on Chips (SoCs). The use of Networks on Chips (NoCs) to connect components in a 3D chip is a necessity. NoC performance largely depends on the underlying deadlock-free and efficient routing algorithm. In this paper a novel selection strategy LATEX¹, is proposed that can be used with any adaptive routing algorithm for specified applications on 2D or 3D topologies. The selection function, which decides the final output channel when a set of admissible output channels exist, is essential for an adaptive routing algorithm. The objective of the proposed selection strategy is to efficiently balance traffic load and reach better performance. Experimental results show that the proposed selection strategy applied to routing algorithm improves average delay and throughput.

Keywords: 3D, Networks on Chip, Application-Specific, Selection Strategy, Contention

1. Introduction

With the increasing complexity of nowadays chips and in consequence their communication requests, networks-on-chip have become the most promising onchip communication architecture. While benefiting from the continuous technology down-scaling, however, the systems-on-chip keep on increasing their area, power consumption and communication latency. With larger chips, several additional problems such as clock distribution and signal integrity must be worked out.

3D integration paradigm is a promising solution to cope with very large chips problems [1] [2]. The 3D technology is based on the integration of multiple silicon layers vertically. The use of Through Silicon Via (TSV) to interconnect two neighboring 2D layers is commonly used in many 3D technologies [3]. This technology provides several major advantages: it addresses the critical interconnect bottleneck in today's chip designs. In 3D, the silicon footprint in each layer is much smaller than a 2D implementation of the system. This is intuitive, as a 3D design can be thought of as a folding of a 2D design into multiple layers. The other advantages of 3D integration include smaller form factor and ease of integration of diverse technologies in a single chip. For example, MEMS, sensors, RF and different memory technologies can be easily integrated by stacking one layer on top of another [4].

The use of NoCs for 3D integration presents several opportunities and challenges for the system designer. The overall performance of a network depends on several properties such as topology, routing algorithm, mapping, and switching techniques. An important concern in NoC implementation is selecting an efficient routing strategy while providing freedom from deadlock. The routing algorithm determines the path that each packet follows between a sourcedestination pair. Routing algorithms noticeably affect the cost and performance of NoC parameters i.e. area, power consumption and average message latency [5].

The effectiveness of any adaptive routing algorithm strongly depends on the underlying selection strategy. When the routing function returns a set of admissible output channels, a selection function is used to select the output channel to which the packet will be forwarded.

Application-specific NoC offers new opportunities to optimize the NoC for the target problem. These include improving performance, simplifying the estimation and control of congestion and design of a more effective selection policy [6].

This paper is an extension of [7] for 3D architecture. Here we propose a novel contention-aware applicationspecific selection strategy for 3D NoC, which

¹ Look-Ahead Traffic-aware EXecution



improves several metrics including average delay and throughput.

2. Related works

2D or 3D NoCs and their associated routing scheme is a widely studied research field, as it offers the possibility to increase system performance and reliability. The majority of the works focus on mesh based topologies. Bartzas and al. directly address the issue of reducing the number of vertical links in 3Dmesh topologies [8]. This work focuses more on the 3D topology aspect than on the routing one. Their idea is to benefit from the advantages of 3D topologies while reducing the area dedicated to vertical links. Several models of vertically partially connected topology are proposed to adapt vertical link locations to the application flows. The authors also proposed a routing algorithm based on the utilization of temporary destinations in intermediate layers. Flexible Dimension Order Routing (FDOR) [9] is a distributed routing scheme in irregular mesh-based 2D or 3D topologies. It relies on the alternation of dimension-order routing algorithms (XYZ, ZYX...); this strategy requires very little additional hardware, but is not applicable to arbitrary topologies and the network is assumed to be vertically fully connected. An oblivious routing algorithm called Randomized Partially-Minimal (RPM) routing in the 3D case introduced in [10]. RPM works by first routing a packet in the minimal direction to a random intermediate "plane" in the vertical dimension. It then routes the packet on the XY plane using either minimal XY or YX routing. Finally, it routes the packet in the minimal direction in the Z vertical dimension to its final destination. In [11] a layer-multiplexed (LM) 3D network architecture have proposed. The LM architecture replaces the one-layerper-hop routing in a conventional 3D mesh with vertical demultiplexing and multiplexing structures and applies on RPM routing as RPM-LM.

In [12], authors presented a detailed simulation study of various selection functions for several fully adaptive wormhole routing algorithms for 2D mesh. The obtained results show that the choice of selection function has a significant effect on the average message latency and saturation behavior.

A selection function called Neighbors-on-Path (NoP) is proposed in [13] for NoC architectures. In NoP, each router grasps buffer availability information of every neighbor router and stores this congestion information locally. When a router is selecting output channel for a packet, it queries its adjacent routers' stored congestion information and make a best decision. In this approach, each router also needs the buffer availability information of its neighbor's neighbor to make the routing decision.

An application specific routing algorithm named Apsra has been proposed by Palesi et al [6]. Apsra exploits communication information to maximize the adaptivity while ensuring deadlock free routing for an application. In this paper, we applied the proposed selection strategy on Apsra.

3. Prerequisites of application-specific NoC design

Mapping an application to on-chip network is the first and the most important step in the design flow as it dominates the overall performance and cost.

Mapping algorithms are mostly focused on 3D mesh [14] topology which is the most popular topology in NoC design due to its suitability for on-chip implementation and low cost. In this paper we have used a random mapping which is presented in Fig. 1.



Fig. 1 Mapping of vopd core graph to 3D mesh.

Due to determined sources and destinations in application-specific NoC, minimum-distance Apsra is mostly considered in this area which the routing is computed offline and admissible paths stored into the routing tables.

In general, every routing algorithm should include deadlock freedom feature [15]. So channel dependency graph (CDG) is used to avoid any possible deadlocks. The CDG is a directed graph with the network channels as the vertices and the direct dependencies between the two channels as the edges. A dependency exists between the links $l_{i,j} \text{ and } l_{j,k}$ whenever there is a path to route packets from v_i to v_k through v_j which uses those links. An extension to CDG as a subgraph is the concept of application specific channel dependency graph (ASCDG) introduced in [6]. The ASCDG is a subgraph of the CDG and an edge in CDG between channels, $l_{i,j}$ and $l_{j,k}$ is removed if there was no application-specific dependency between li,j and lj,k. Deadlock is inevitable when there are any cycles through ASCDG graph. A cycle in the ASCDG is a



succession of application specific direct dependencies. Suppose that $D = \{d_1, d_2, \dots, d_n\}$, where $d \in D$ is a pair $(l_{i,j}, l_{j,k})$ with $l_{i,j}, l_{j,k} \in L$. If there is any cycle, we need to break it and by removing a dependency, all of the corresponding paths to that dependency will be removed. Using this method guarantees that routing algorithm is deadlock-free. So we make Apsra deadlock free by using ASCDG via breaking cycles [16].

4. Principles and Definitions

4.1 Offline procedure

Main part of our selection strategy computation, i.e. link contention and equivalent resistance are computed offline.

1) Link Contention (C): Link contentions defined as the aggregate amount of traffic which can pass through a specified link based on the communications given in the communication graph. After the IP core mapping phase of the NoC design flow, we have complete knowledge about the pairs of cores that communicate and other pairs that never communicate. Considering the communication graph, if we assume ρ_i to be a path for the desired communication, ρ_{comm} the set of all such possible paths, t_{comm} the traffic generated by that communications specified in traffic scenario extracted from communication graph, contention for each given link L can be formulated as follows:

$$C_{L} = \sum_{comm=1}^{ncomm} (\mu * t_{comm})$$

$$u = \begin{cases} 1 : \exists \rho_i \in \rho_{comm} : L \in \rho_i \\ 0 : else \end{cases}$$

We consider a realistic communication scenario, a generic Video Object Plane Decoder (*VOPD*), to elaborate the proposed strategy. As depicted in Fig. 1, under random mapping core 5 and 6 are mapped into tiles 8 and 10 respectively. The communication traffic volume between core 5 and core 6 is 357.

The routing function provides all possible paths between two nodes for a given communication. Minimum-distance Apsra which is proposed in [6] is used as the routing function in our explanation. According to Fig. 1, core 5 and 6 is mapped to tile 8 and 10 respectively. Considering Fig. 2(a), presume node 8 is the source and node 10 the destination, Apsra gives West (W), North (N) and Down (D) as the possible output channels toward destination so we place 357, the traffic volume for $Comm_{8,10}$ on West, North and Down output links of node 8. From West

direction moving on to node 7, Apsra specifies directions north and Down can be followed to reach destination, so as before we assign 357 to the output link along the North and Down directions of node 7. Going along North from 7, we have again North and Down at node 4 to reach 10 so 357 is placed on the corresponding links. Back to node 7 and going along Down, we have only North at node 16 and 13 to get to 10. At node 4, we may follow either Down, which is traversed already, and North which brings us to node 1 from there we reach destination (node 10) by following Down. Following the North direction of node 8 we get to 5, from there Apsra states we can take West, North or Down to reach node 10 so the output links along these directions also take the value of 357 as the contention value so far. The same is followed to reach the destination node and contention values are updated for each link alongside the complete path. We also compute contention for each link alongside Down direction of node 8, exactly the same as previous. Following the same procedure described above for each of the 21 communications specified in Vopd application, the contention for each link is calculated as the sum of all values assigned to the link during applying all communications' possible paths. Now we move on to the next step in offline phase which involves computing equivalent resistance for all communications.

2) Equivalent Resistance: We define equivalent resistance (ER) for a given communication using electrical concepts in Kirchoff's laws such that each node in our topology is considered as a circuit node and each link as a resistor with a magnitude equal to the link's contention. ER calculation is best explained by means of an example.



Fig. 2 Communication traffic volume between node 8 and 10.

There are eleven possible paths between node 8 and node 10 according to Apsra and the ER for this communication is shown in Fig. 2(b). Each resistor has a value equal to the computed contention in previous section. For each possible direction at each node and for each possible path of a given





Fig. 3 Equivalent circuit for (a) west (b) north (c) down directions.

communication, ER should be computed separately. The procedure for calculation of ER in the example is as follows:

Starting from source node, there are three possible outputs at node 8 towards node 10, namely directions West, North and Down, so first we compute ER for West direction and then proceed similarly with North and South directions.

There are three paths at node 8 alongside West direction so we have to simplify the circuit by employing the series and parallel resistors' simplification rules and Y-Delta Transform. Fig. 3(a) shows the entire circuit considering only West direction from node 8 to node 10. By simplifying the circuit, we come up with one equal resistor between node 8 and 10, so:

 $ER_{8.10}[West] = C(L_{8.10})$

Back to node 8 and proceeding along the North direction (Fig. 3(b)), there are five possible path from source to destination. Same as previous by simplifying the resistors we have:

 $ER_{8,10}[North] = C(L_{8,10})$

Notice that these two $C(L_{8,10})$ are not equal. One of them is for West direction and another for North.

Again back to node 8 and this time proceeding along the Down direction as shown in Fig. 3(c), after simplification we come up with:

$$ER_{8,10}[Down] = C(L_{8,10})$$

Finally the traffic load percentage alongside each direction is calculated based on these ER values such that the higher calculated ER for a direction indicates higher traffic load along that direction, therefore the respective link gets a lower selection probability in our algorithm in order to transmit the packet on the link at each runtime. For calculating the selection probability of the direction, first we sort the *ER*s of output links ascending. So in this example the order of ERs is: $ER_{8,10}$ [North], $ER_{8,10}$ [Down], $ER_{8,10}$ [West]. As it is

clear, *ER* of West direction has the maximum value hence its selection probability will be minimum.

Psel is the selection probability of the direction, having reverse relation to its contention.

$$\begin{split} Psel_{8,10}[West] &= ER_{8,10}[North] / (ER_{8,10}[West] \\ &+ ER_{8,10}[North] + ER_{8,10}[Down]) \\ Psel_{8,10}[North] &= ER_{8,10}[West] / (ER_{8,10}[West] \\ &+ ER_{8,10}[North] + ER_{8,10}[Down]) \\ Psel_{8,10}[Down] &= ER_{8,10}[Down] / (ER_{8,10}[West] \\ &+ ER_{8,10}[North] + ER_{8,10}[Down]) \end{split}$$

The final values for *Psel* calculated above are stored at router 8 for destination 10. It should be noted that a router may have different selection probability values for its output links based on destination, so the selection probability for any link is determined by the destination of the packet.

The same procedure is applied at intermediate routers 5, 7, 2, 4, 17 and 14 which also have indecision points for packets destined to node 10. By indecision we mean there are two or three possible output links returned by minimal routing function.

To get the offline phase completely done, steps explained above have to be repeated for every router with indecision in the possible paths, considering all communications.

Notice that if there is only one possible link returned by routing function to follow, we do not need store a value for *Psel* towards that destination in the router. But if there are two or three possible output links for a given routing decision, then a *Psel* value is set in the router for each destination node and for each possible direction.

Much of the selection process is handled offline without imposing any additional processing or traffic load on the router, providing an important advantage in favor of our technique over many other selection



strategies. Also we fully anticipate the traffic caused by all of the communications in the network.

4.2 Online operation requirements

At runtime, we make use of some additional information available to router which is described below:

- Free Buffer Slot[d](*B*): number of free buffer slots available at the input buffer of the adjacent neighbor along the direction d where d is one of (North, South, East, West, Up, Down);
- Instant Power (Δp) : we define instant power as the difference between power consumed by the router at instant t and t - 1 where t is the moment we want to select final output channel for a packet;

 $\Delta p = power(t) - power(t-1)$

According to [17], we assumed power consumption and delay of vertical links equal to horizontal links in each layer.

These state information help us to better estimate the traffic load of the network and creates a more efficient decision at the time of next hop selection.

5. Proposed selection function algorithm

When the routing function provides several output channels, for each candidate channel the selection algorithm first examines whether the channel is available to send the packet (header flit) along or it is reserved by some previous header flit, through consulting the reservation table. Should the channel be available selection score is calculated for the channel and finally the channel with the highest score is chosen. If more than one channel has the same maximum score, the first one will be selected. Score computation is defined as the formula below:

 $Score[d] = \alpha * Psel[d] + \beta (B[d] / max_buffer_size)$ $+ \gamma (\Delta P / max_instant_power)$

 α , β , and γ are weight factors for link selection probability, free buffer slots, and instant power consumption, respectively. These coefficients allow for full dynamic adaptivity of selection algorithm and setting their value judiciously results in best optimization as we will see in section 6.

Since free buffer slots (*B*) and instant power consumption (Δp) have different units, we normalize them by making use of factors *max_buffer_size* and *max_instant_power*. Obviously *Psel* is already in [0,1] span thus no need for normalization.

As an example, suppose that node 0 decides to send a packet toward the destination node 5. Employing Apsra, there are two admissible output channels toward node 1 and node 3. As shown in Fig. 4, router 0 is aware of free buffer slots and power consumption state of its neighbors. So using this information and those saved in the router's own memory, a score is computed and the best channel is selected.



Fig. 4 Structure of the router.

```
bestDirectionMark = -1, bestDirection = 0;
foreach direction in possibleDirections {
  if isReserved(direction)
     continue;
  selectionProbability =
     probabilityMatrix[currentId]
       [destinationId][direction];
  nextHopInstantPower =
     nocNodes[nextHopId].power -
        lastPowers[nextHopId];
  nextHopBuffer = freeSlotsNeighbor[direction];
  thisDirectionMark = alpha *
        selectionProbability + beta * (nextHopBuffer
        / BUFEER_DEPTH) - gamma * (instantPower
        / MAX_POWER);
  if thisDirectionMark > bestDirectionMark {
     bestDirection = direction;
     bestDirectionMark = thisDirectionMark:
 }
return bestDirection;
```

.

Fig. 5 Pseudo-code for Latex algorithm.

We evaluated score formula for Apsra, for all possible values of α , β and γ satisfying formula below and earned the best coefficients.

 $\begin{aligned} \alpha + \beta + \gamma &= 1; \alpha = 0, 0.1, \dots, 1; \beta = 0, 0.1, \dots, (1 - \alpha) \\ \gamma &= 1 - (\alpha + \beta). \end{aligned}$



The best α , β , and γ is 0.3, 0.4, and 0.3 respectively. Another important feature of our selection strategy is that it can be coupled with any type of network topology and adaptive routing function. Fig. 5 shows the pseudo-code for the Latex algorithm.

6. Experimental results

For evaluation of the performance and simulating the proposed selection strategy we used Active HDL simulation platform. For performance metrics, we chose average delay and throughput.

Delay is defined as the time that header flit injects into the network till the tail flit receives at destination node. We formulate average delay (D) as follow:

$$D = \frac{1}{n} \sum_{i=1}^{n} D_i$$

Where *n* is the total number of packets reaching their destination nodes and D_i is the delay of packet *i*. Throughput defines as follows similarly to [18]:

$$Throughput = \frac{Total received flits}{Number of nodes \times Total cycles}$$

Where Total received flits refers to the number of whole flits that arrive at their destination, Number of nodes is the number of nodes in our topology and Total cycles is the number of clock cycles between the time which first message generation and the last message reception.





Fig. 7 Comparison results under MMS traffic scenario.

We evaluate the Latex selection strategy using two real traffic scenarios generated by a Video Object Plane Decoder (*VOPD*) and MultiMedia System (*MMS*), on $3 \times 3 \times 2$ and $4 \times 4 \times 2$ mesh sizes respectively. For our experiments, packets are generated



Table	1: Improven	nents in Ave	erage Dela	y of Lat	ex Compared to	XYZ Ro	uting Algori	thms under	VOPD and	MMS traff	fic scenario
MMS Avg. delay improvement				Average	VOPD Avg. delay improvement					Average	
Traffic load			improvement	Traffic load					improvement		
1x	2x	3x	4x	5x		1x	2x	3x	4x	5x	
14.12%	13.90%	11.74%	9.54%	6.93%	11.24%	17.8%	17.08%	16.41%	12.16%	9.73%	14.63%

according to a uniform distribution with the rates extracting from communication volumes in the *VOPD* and *MMS* core graph. It means the number of packets generated in the source cores are derived from the core graph edges and in order to increase the traffic load, communication volumes are multiplied by a traffic factor *i*. The FIFO buffers have a capacity of four flits.

For each traffic scenario, average delay and throughput with various traffic loads has been evaluated.

To evaluate Latex strategy, we run Latex in conjunction with Apsra and compared the results with XYZ routing algorithm. Fig. 6 and 7 show the results obtained under application of *VOPD and MMS* traffic scenarios. As depicted, Latex algorithm has

outperformed XYZ in terms of both average delay and throughput. Table 1, shows the percent improvement of Latex over XYZ routing algorithm below saturation point. Table 2, shows received flit rate for different simulation times. Expectedly, as the traffic increases with the time, more flits receive at destination compared to XYZ. We believe Latex outperforms others because of two important design characteristics, first taking into account the load condition of the entire communication path, and second delegating some processing load from runtime to offline phase.

	VO	PD	MMS			
Simulation	Received	l flits rate	Received flits rate			
cycles	XYZ	Apsra-	XYZ	Apsra-		
		Latex		Latex		
10,000	12.1%	11.6%	14.0%	12.8%		
20,000	38.9%	43.8%	29.0%	31.2%		
30,000	73.6%	89.1%	51.7%	58.6%		
40,000	84.0%	100.0%	78.6%	85.4%		
50,000	100.0%		91.2%	100.0%		
60,000			100.0%			

Table 2: Received flit rate for Latex and XYZ

7. Conclusion

In this paper we proposed a selection strategy for adaptive routing aiming at network contention minimization and improved performance for 3D NoC. We have reported our results obtained from experiments involving two real benchmarks. The selection strategy has been applied to minimal-distance Apsra and compared to XYZ routing. The results show reduction in packet latency with acceptable improvement in throughput. A further step will be the power analysis of Latex algorithm on 3D architecture to assess the validity of the proposed selection strategy.

References

- C. Rusu, L. Anghel and D. Avresky, "Adaptive interlayer message routing in 3D networks-on-chip," *Microprocessors and Microsystems* 35, 2011.
- [2] A. Zia, S. Kannan, H. Jonathan Chao and Garrett S. Rose, "3D NOC for many-core processors," *Microelectronics Journal* 42, pp. 1380–1390, 2011.
- [3] Paramasivam and S. Viswanathan, "Exploring Optimal Topology and Routing Algorithm for 3D Network on Chip," *American Journal of Applied Sciences 9*, 2012.
- [4] S. Murali, L. Benini and G. De Micheli, "Design of Networks on Chips for 3D ICs," in *Design Automation Conference (ASP-DAC)*, 2010, pp. 167 - 168.
- [5] M. Janidarmian, A. Khademzadeh, A. Roshanfekr, V. Samadi, "A Methodology for Application-Specific Network-on-Chips Design," in *International Journal of Computer Science*, 2011.
- [6] M. Palesi, R. D. Holsmark, S. Kumar, and V. Catania, "Application Specific Routing Algorithms for Networks on Chip," in *IEEE Transactions on Parallel and Distributed Systems, vol. 20, no. 3*, March 2009.
- [7] S. Azampanah ,A. Khademzadeh, N. Bagherzadeh, M. Janidarmian and R. Shojaee, "LATEX: New Selection Policy for Adaptive Routing in Application-Specific NoC," in 20th Euromicro International Conference on Parallel, Distributed and Network-based Processing (PDP), 2012.
- [8] A., Skalis, N., Siozios, K., Soudris, D. Bartzas, "Exploration of Alternative Topologies for Application-Specific 3D Networks-on-Chip," in WASP, 2007.
- [9] T. Skeie et al., "Flexible DOR Routing for Virtualization of Multicore Chips," in *System-on-Chip*, SOC'09. *International Symposium*, 2009, pp. 073-076.
- [10] R. Sunkam Ramanujam and Bill Lin, "Randomized Partially-Minimal Routing on Three-Dimensional Mesh Networks," in *IEEE Computer Architecture Letters*, 2008.
- [11] R. Sunkam Ramanujam and Bill Lin, "A Layer-Multiplexed 3D On-Chip Network Architecture," in *IEEE Embedded Systems Letters*, 2009.
- [12] L. Schwiebert and R. Bell, "Performance Tuning of Adaptive Wormhole Routing through Selection Function Choice," *Parallel and Distributed Computing*, July 2002.
- [13] V.Catania, M.Palesi,D.Patti G. Acsia, "Implementation and Analysis of a New Selection Strategy for Adaptive Routing in Networks-on-Chip," in *IEEE Transactions on Computers, vol. 57, no. 6*, June 2008, pp. 809-820.



- [14] B. Stanley Feero, P. P. Pande, "Networks-on-Chip in a Three-Dimensional Environment: A Performance Evaluation," *IEEE Transaction on Computers*, 2009.
- [15] F. Dubois, A. Sheibanyrad, F. Pétrot and M. Bahmani, "Elevator-First: a Deadlock-Free Distributed Routing Algorithm for Vertically Partially Connected 3D-NoCs," *IEEE Transaction on Computers*, 2011.
- [16] F. Karimi, A. Khademzadeh and M. Janidarmian, "Fault-Tolerant Application-Specific Network-on-Chip," in Proceedings of the World Congress on Engineering and Computer Science (WCECS), 2011.
- [17] Luca P. Carloni, P. Pande and Y. Xie, "Networks-on-Chip in Emerging Interconnect Paradigms: Advantages and Challenges," in *Third International Symposium on Networks-on-Chip (NOCS)*, 2009.
- [18] P.P. Pande, C. Grecu, M. Jones, A. Ivanov, and R. Saleh, "Performance Evaluation and Design Trade-Offs for Network-on-Chip Interconnect Architectures," *IEEE Trans. Computers*, pp. 1025-1040, 2005.