Torus Topology based Fault-Tolerant Network-on-Chip Design with Flexible Spare Core Placement

P. Veda Bhanu*, Pranav Kulkarni*, Soumya J*, Linga Reddy Cenkarmaddi†, and Henning Idsøe†
*Department of EEE, Birla Institue of Technology and Science-Pilani, Hyderabad, Telangana, India - 500078
*{vedabhanuiit2010, kulkarni.pranav2, soumyatkgp}@gmail.com
†Department of Information and Communication Technology, University of Agder, Norway
†{linga.cenkeramaddi, henning.idsoe}@uia.no

***Abstract*—The increase in the density of the IP cores being fabricated on a chip poses on-chip communication challenges and heat dissipation. To overcome these issues, Network-on-Chip (NoC) based communication architecture is introduced. In the nanoscale era NoCs are prone to faults which results in performance degradation and un-reliability. Hence efficient fault-tolerant methods are required to make the system reliable in contrast to diverse component failures. This paper presents a flexible spare core placement in torus topology based fault-tolerant NoC design. The communications related to the failed core is taken care by selecting the best position for a spare core in the torus network. By considering this we propose a meta-heuristic based Particle Swarm Optimization (PSO) technique to find suitable position for the spare core that minimizes the communication cost. We have experimented with several application benchmarks reported in the literature by varying the network size and by varying the fault-percentage in the network. The results show significant reduction in terms of communication cost compared to other approaches reported in literature.**

***Keywords*—Fault-tolerance, Network-on-Chip, Particle Swarm Optimization, Torus topology, Communication cost, Spare core.**

I. Introduction

The advancements in the VLSI technology has led to an increase in the density of the IP cores being fabricated onto a chip. This has led to an increase in the communication complexity on a chip. To achieve high communication efficiency and to overcome communication complexity, a Network-on-Chip (NoC) has been proposed in [1]. NoCs consist of various network components such as switches and network interfaces which facilitate communication. The cores are connected to the routers and the information is exchanged using packet switching technique. Designing a NoC system requires taking into consideration many aspects such as mapping of the cores, routing algorithms, power consumption, reliability and so on. Many performance parameters like latency and throughput in turn depend on these various factors. The NoCs are scalable unlike the traditional buses and are designed to handle the high inter-core bandwidth requirements. The high bandwidth requirements translates to high switching activity leading to higher heat dissipation. The non-uniform heat dissipation might lead to failure of the components. It is therefore essential to introduce fault tolerant techniques to build reliable and robust NoC systems subject to minimizing the communication costs. The design of an NoC system for satisfying incoming applications with high inter-core bandwidths requirements necessitates proper application mapping strategies to develop reliable systems. To cater the core failures, this paper mainly focuses on flexible placement of a spare core in the given network. The remainder of the paper is organized as follows. Section II outlines the literature survey. Section III is devoted to introduction of PSO and explains about the evolution of generations. Section IV details the spare core placement using DPSO. Experimental results are presented in Section V. Section VI concludes the paper.

II. Related Work

Quite recently, considerable attention has been paid to fault-tolerant methodologies in NoC design. Much research on fault-tolerant NoC architectures in the literature which include redundancy in router, link has been done. To make the system reliable, most of the works have been concentrated on different routing techniques proposed for fault-tolerant NoC. However, to the best of our knowledge, very few publications can be found in the literature that address the issue of fault-tolerant application mapping in NoC. As reported by [2], energy and reliable aware application mapping approach proposes a cost model by combining energy and reliability. In this work it was observed that branch-and bound-based mapping technique for regular topologies has been used. However, they have assumed fault-free cores and link faults in this work. FoToNoC, folded torus like NoC based many core SoC in the dark silicon era has been presented in [3], they have considered hierarchical mapping including cluster management strategy and task mapping strategy. A methodological study of optimizing application performance on a five-dimensional torus network via the technique of topology-aware task mapping has been presented in [4]. Their work focuses on placement of processes on computing nodes while carefully considering the network topology between the nodes and the communication behavior of the application. However approaches in [3] and [4] does not provide a solution in the event of core faults. In [5], an Integer Linear Programming (ILP) based technique has been presented for task remapping onto fault-tolerant NoC. The authors in [5] have also developed heuristics for the online task remapping problem. In [6], a fault-tolerant multi-application mapping has been presented. It works in two stages, initially an application is mapped onto non-faulty processing cores and in the next stage the spare core is placed onto the fault free cores. The authors in [6] has developed heuristic for mapping phase and design space exploration algorithm has been proposed for spare core placement. In [7] it was shown that spare core has been placed at the center of the mesh network using a mapping algorithm. Authors in [8] have considered the faults in links and developed a design using spare link. The faults in routers have been considered and developed a design using spare routers [9]. To tolerate router faults the same research group in [9] has proposed a fault-tolerant NoC design in [10] using spare links and double network interface routers. The reliability aware application mapping method has been presented in [11] by proposing an integrated cost function to generate multiple solutions according to the requirements of the user. Although several studies have indicated that faults in cores, links and routers can be addressed using different techniques proposed in the literature, little attention has been paid to faults in cores of an application that has to be mapped onto NoC.

III. Discrete Particle Swarm Optimization

Particle Swarm Optimization (PSO) [12] is a population based stochastic technique designed and developed by Eberhart and Kennedy in 1995. The following optimization problem requires a discrete variant of PSO - DPSO [13]. In a DPSO system, multiple candidate solutions coexist and collaborate simultaneously. Instead of using genetic operators, each individual particle (solution) adjusts its flying in problem space according to its own flying experience as well as its companions flying experience. In the search space of DPSO, each individual particle has a fitness value, which determines its quality. Each particle is treated as a point in n-dimensional space. In every run, the fitness function is evaluated by taking the position of the particle in solution space. The position of i^{th} particle at k^{th} iteration is denoted as $p_k{}^{\text{i}}$. Each particle keeps track of its best value obtained so far. This is called as personal best or local best of i^{th} particle (*pbest*$^{\text{i}}$). Similarly, the best fitness value across the whole swarm is called global best ($gbest_k$) for the generation *k*. The new position of the i^{th} particle can be calculated by

$$p^i_{k+1} = (c_1 * I \oplus c_2 * (p_k \rightarrow pbest^i) \oplus c_3 * (p_k \rightarrow gbest_k))p^i_k \tag{1}$$

In the preceding expressions, $a \rightarrow b$ represents the minimum length sequence of swappings to be applied on components of *a* to transform it to *b*. For example, if a = <6, 7, 8, 9> and b = <9, 6, 7, 8>, $a \rightarrow b$ = < swap (1, 4), swap (2, 4), swap (3, 4)>. The operator $\oplus$ is the fusion operator, applied on two swap sequences. The sequence of swaps in *a* is followed by the sequence of swaps in *b* is given by $a \oplus b$. The constants c_1, c_2, c_3 are the inertia, self-confidence and swarm confidence values, respectively. The quantity means that the swaps in the sequence $a \rightarrow b$ will be applied with a probability of c_i . Swap sequences such as <swap (1, 1), swap (2, 2), swap (3, 3) ... swap (n, n)> are known as identity swaps which is denoted by *I*. It corresponds to the inertia of the particle to maintain its current configuration. To generate a new particle p^i_{k+1}, the final swap sequence is applied on particle p^i_k mentioned in above equation (1). From [14], it can be found that the convergence condition for this PSO is given by,

$$(1-\sqrt{c_1})^2 \leq c_2 + c_3 \leq (1+\sqrt{c_1})^2 \tag{2}$$

Accordingly, we have worked with various values of c_1, c_2 and c_3. The results reported in this paper are based upon the values of c_1= 1,c_2=0.5 and c_3= 0.5. This completes an overview of PSO.

IV. Spare Core Placement via DPSO

We present our formulation for mapping the incoming core graph onto the torus architecture along with the spare core. The input torus floorplan gives the number of available position onto which the cores can be mapped to. The input torus floorplan should have enough number of available positions to accommodate all the cores in the core graph along with the spare core. In our formulation, the spare core has been numbered the last. However, any entry in the particle can be taken.

A. Particle formulation and Fitness Function

Each particle is a specific arrangement of cores which represents their association with routers. The quality or fitness of a particle depends on the association of the cores to the routers. The fitness of a particle is the communication cost calculated by the following formula:

$$Communication\ cost = \sum_{\forall Edges} (Number\ of\ Hops * Bandwidth) \tag{3}$$

For the problem of spare core placement, each particle is modeled as single dimensional array wherein the index of the array determines the router number and the value at the index determines the core number. This scheme conveys the association of the core with the router. The formulation takes the failed core as user input and in the event of a core failure, the communications associated with the failed core are taken care by the spare core.

B. Local and Global bests

During the course of evolution, the particles undergo modification which changes the association of the cores with the routers. This leads to a change in the quality/fitness of the particles. Each particle keeps track of the best set of core positions that it has encountered resulting in minimum fitness known as local best (*lbest*). Across the generations, the swarm keeps track of the set of core positions which it has encountered resulting in minimum fitness known global best (*gbest*). Both these values help in guiding the particles towards minimum fitness during the process of evolution. These are updated if newer values are lesser than the previously stored values during evolution.

C. *Evolution of generation*

The particles evolve over the generations and transform into new particles having fitness closer to optimum. Initially, the particles are randomly generated and fitness of each particle is evaluated. In the first generation, local best *(lbest)* of each particle is initialized as the particle itself and the global best is evaluated. The remaining generations are evaluated according to the eqn. 1. The evolution centers around the particles being swapped with a probability towards *lbest* and *gbest* set of core positions. The termination condition is reached when the number of generations reach a preset number of generations or if there is no improvement in the solution for a certain number of generations.

V. Experimental Results

We present the results obtained by simulating our technique on a PC embedded with Intel Xeon processor operating at 3.5 GHz with 32 GB internal memory. The DPSO was coded in a high level language *C++* and simulated for several benchmark applications reported in the literature. Communication cost is calculated by equation (3). For fair comparison we have extended the work followed in approach [7] to torus topology. We compare the results generated by DPSO with the approach followed by [7]. A sample application mapping with spare core placement using approach [7] and our approach has been shown in Fig. 1.

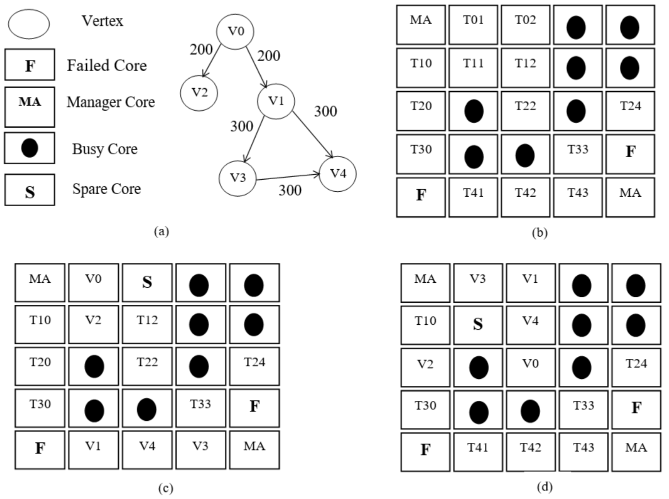

Fig. 1: Spare core placement. (a) An example core graph, (b) 5x5 Torus based NoC, (c) Spare core placement using our approach, (d) Spare core placement using approach [7].

TABLE I: Calculation of the communication cost for Fig. 1

Source Core	Destination Core	Communication Bandwidth	Approach [7]		Our Approach	
			Number of hops (N)	Communication cost (N x BW)	Number of hops (N)	Communication cost (N x BW)
V0	V2	200	2	400	1	200
V0	V1	200	2	400	1	200
V1	V3	300	1	300	2	600
V1	V4	300	1	300	1	300
V3	V4	300	2	600	1	300
Communication Cost				2000		1600

Fig. 1(a) shows an example core graph, let us assume core V1 as failed core input given by user, hence the spare core has to be placed taking V1 into account. It can be observed from Fig. 1(a), that the core V1 communicates with cores V0, V3 and V4 with a bandwidth of 200, 300 and 300 Mbps respectively. The spare core should be placed such that in the event of core failure (V1), the spare core (represented by S in Fig. 1(c) and 1(d)) should take over the communication associated to V1 with minimum communication cost. In the Table I, our approach could give a better communication cost of 1600 against 2000 given by the approach [7]. To test for the scalability and efficiency of our approach compared to approach [7], the size of the torus network was increased (5x5, 10x10 and 15x15). The simulation results showing the percentage improvement in communication cost in our approach compared to approach [7] are tabulated in Table II.

TABLE II: Comparison of communication cost by varying the torus network size to 5x5, 10x10 and 15x15 with 30% faults in the network.

Application	5x5			10x10			15x15		
	A	B	C	A	B	C	A	B	C
MPEG	5751	3850	27.47%	6431	3946	27.47%	7880	5213	33.84%
MWD	1568	1184	12.24%	1824	1664	22.80%	2464	1792	27.27%
263Encoder	30.85	23.04	17.34%	36.48	29.28	28.00%	43.58	28.01	35.72%
MP3Encoder	24.21	17.05	26.76%	39.58	17.97	44.11%	24.24	19.99	17.52%
263Decoder	29.29	19.94	30.62%	30.88	24.53	32.40%	65.49	24.01	63.33%
VOPD	6374	4105	29.55%	5819	4708	7.92%	11790	7962	32.47%

A = Approach in [7]; B = Our approach; C = % Improvement by our approach;

A major drawback of approach [7] is that it tries to place the spare core at the center of the network while ignoring the communication cost of the failed core. On the other hand, in our approach, flexibility is given for placement of the spare core among the available positions in the input torus floorplan. The flexibility helps in reducing the communication cost. From Table II, we can observe that the average percentage of improvement (23.99%, 26.32% and 35.02%) increases as the size of the torus network (5x5, 10x10 and 15x15) increases since the search space for the spare core placement increases and in turn finds the suitable position. This shows the scalability of the proposed approach. Fault percentage is termed as number of non-available positions to total number of available positions in the torus network. Table III shows the comparison of communication cost between our approach and the approach followed in [7] for different application benchmarks by varying the fault percentage (0%, 15%, 30% and 50%) in the torus network of size 10x10. We could get a maximum of 45.14% improvement compared to the approach [7] in terms of communication cost. From the Table III, it can be observed that the average percentage improvement (20.72, 21.15, 24.24 and 30.77) increases as the fault percentage (0%, 15%, 30% and 50%) increases, which shows the applicability of our approach in serious fault environments. Fig. 2 represents the communication cost calculated for MPEG application by taking failed core as an input from the user in our approach

TABLE III: Comparison of communication cost by varying the faults from 0% to 15%, 30%, 50% in 10x10 torus network.

Application	0%			15%			30%			50%		
	A	B	C	A	B	C	A	B	C	A	B	C
MPEG	4314	3570	17.24%	5823	3780	35.08%	6431	3946	41.38%	6773	4487	33.75%
MWD	1440	1408	2.22%	1856	1536	17.24%	1824	1664	8.77%	2464	1664	32.46%
263Encoder	29.28	25.52	12.84%	33.17	25.52	23.06%	36.48	29.28	19.73%	43.17	25.11	41.89%
MP3Encoder	22.73	17.07	24.90%	21.58	17.74	17.79%	39.58	17.97	31.98%	31.23	17.13	45.14%
263Decoder	35.88	24.39	32.03%	31.47	25.43	19.19%	30.88	24.53	20.56%	45.15	38.10	15.69%
VOPD	7160	4647	35.09%	5508	4708	35.09%	5819	4708	23.05%	7724	6514	15.66%

A = Approach in [7] B = Our Approach C = % Improvement by our approach

and approach followed in [7]. On X-axis we have taken failed core and on Y-axis the communication cost is considered. We can observe significant improvement in communication cost for each core failure in the MPEG application by our approach. However, we have also experimented with all other benchmarks reported in Table I, but we have reported only MPEG application with 30% faults in 5x5 torus network. This shows that our approach can be efficiently applicable to any core failures given as an input by the user in different benchmark applications.

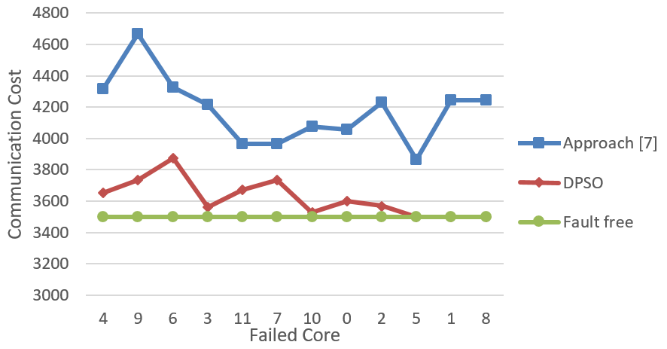

Fig. 2: Communication cost with varying core faults

VI. Conclusion

In this paper, a DPSO based meta-heuristic technique has been proposed to select the best position for spare core in the torus based floorplan which reduces the communication cost. Based on the results, it can be concluded that our approach has achieved minimal communication cost by flexible spare core placement in the torus topology. Future work involves flexible placement of multiple spare cores from the available positions in the torus based floorplan and proposing exact methods like Integer Linear Programming.

Acknowledgements

This work is partially supported by the research project No. ECR/2016/001389 Dt. 06/03/2017, sponsored by the SERB, Govt. of India

References

[1] W. J. Dally and B. Towles, "Route packets, not wires: on-chip interconnection networks," in *Proceedings of the 38th Design Automation Conference (IEEE Cat. No.01CH37232)*, 2001, pp. 684–689.

[2] L. Liu, C. Wu, C. Deng, S. Yin, Q. Wu, J. Han, and S. Wei, "A flexible energy- and reliability-aware application mapping for noc-based reconfigurable architectures," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 23, no. 11, pp. 2566–2580, Nov 2015.

[3] L. Yang, W. Liu, W. Jiang, M. Li, P. Chen, and E. H. M. Sha, "Fotonoc: A folded torus-like network-on-chip based many-core systems-on-chip in the dark silicon era," *IEEE Transactions on Parallel and Distributed Systems*, vol. 28, no. 7, pp. 1905–1918, July 2017.

[4] A. Bhatele, N. Jain, K. E. Isaacs, R. Buch, T. Gamblin, S. H. Langer, and L. V. Kale, "Optimizing the performance of parallel applications on a 5d torus via task mapping," in *2014 21st International Conference on High Performance Computing (HiPC)*, Dec 2014, pp. 1–10.

[5] O. Derin, D. Kabakci, and L. Fiorin, "Online task remapping strategies for fault-tolerant network-on-chip multiprocessors," in *Proceedings of the Fifth ACM/IEEE International Symposium*, May 2011, pp. 129–136.

[6] F. Khalili and H. R. Zarandi, "A fault-tolerant low-energy multi-application mapping onto noc-based multiprocessors," in *2012 IEEE 15th International Conference on Computational Science and Engineering*, Dec 2012, pp. 421–428.

[7] B. N. K. Reddy, M. H. Vasantha, and Y. B. N. Kumar, "A gracefully degrading and energy-efficient fault tolerant noc using spare core," in *2016 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*, July 2016, pp. 146–151.

[8] N. Chatterjee, N. Prasad, and S. Chattapadhya, "A spare link based reliable network-on-chip design," in *18th International Symposium on VLSI Design and Test*, July 2014, pp. 1–6.

[9] N. Chatterjee, S. Chattopadhyay, and K. Manna, "A spare router based reliable network-on-chip design," in *2014 IEEE International Symposium on Circuits and Systems (ISCAS)*, June 2014, pp. 1957–1960.

[10] N. Chatterjee and S. Chattopadhyay, "Fault tolerant mesh based network-on-chip architecture," in *2015 IEEE International Symposium on Circuits and Systems (ISCAS)*, May 2015, pp. 417–420.

[11] N. Chatterjee, S. Reddy, S. Reddy, and S. Chattopadhyay, "A reliability aware application mapping onto mesh based network-on-chip," in *2016 3rd International Conference on Recent Advances in Information Technology (RAIT)*, March 2016, pp. 537–542.

[12] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Neural Networks, 1995. Proceedings., IEEE International Conference on*, vol. 4, Nov 1995, pp. 1942–1948 vol.4.

[13] K.-P. Wang, L. Huang, C.-G. Zhou, and W. Pang, "Particle swarm optimization for traveling salesman problem," vol. 3, 2003, pp. 1583–1585, cited By 218. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-1542316003partnerID=40md5=f0d49a515af704754d12beb9bbba63ca

[14] G. Luo, H. Zhao, and C. Song, "Convergence analysis of a dynamic discrete pso algorithm," in *2008 First International Conference on Intelligent Networks and Intelligent Systems*, Nov 2008, pp. 89–92.