Multi-Application based Network-on-Chip Design for Mesh-of-Tree topology using Global Mapping and Reconfigurable Architecture

Mohit Upadhyay*, Monil Shah*, P. Veda Bhanu*, Soumya J*, and Linga Reddy Cenkarmaddi†
*Department of EEE, BITS-Pilani, Hyderabad, Telangana, India - 500078
*{mupadhyay09, shahmonil1996, vedabhanuiit2010, soumyatkgp}@gmail.com
†Department of Information and Communication Technology, University of Agder, Norway
†linga.cenkeramaddi@uia.no

***Abstract*—This paper outlines a multi-application mapping for Mesh-of-Tree (MoT) topology based Network-on-Chip (NoC) design using reconfigurable architecture. A two phase Particle Swarm Optimization (PSO) has been proposed for reconfigurable architecture to minimize the communication cost. In first phase global mapping is done by combining multiple applications and in second phase, reconfiguration is achieved by switching the cores to near by routers using multiplexers. Experimentations have been carried out for several application benchmarks and synthetic applications generated using TGFF tool. The results show significant improvement in terms of communication cost after reconfiguration.**

***Keywords*—Network-on-Chip, Particle Swarm Optimization, Mesh-of-Tree topology, Reconfiguration, Communication cost.**

I. Introduction

In nano scale era, bus based architectures namely System-on-Chips (SoCs) do not scale well with increased application requirements. To mitigate the architectural limitations in SoCs, an efficient and scalable on-chip interconnection paradigm known as Network-on-Chip (NoC) has been proposed. The communication among several IP cores in NoC has been achieved using packet switching techniques via underlying router fabric. Recent study in the literature has revealed that most of the NoCs are designed to run single application [1]. Since the application requirement is increasing, NoCs designed for single application may not suffice for high performance, throughput and network latency. Hence, there is a need for reconfigurable architecture that can run multiple applications on a NoC. Reconfiguration can be achieved by deploying few hardware components like switches, and multiplexers over the existing NoC architectures.

A very few works have been proposed in the literature which address multiple application mapping using reconfiguration in NoC. A reconfigurable NoC that enables the network topology to be modified using switches based on the application currently being run has been proposed in [2]. To meet the application requirements, the works proposed in [3, 4] perform reconfiguration using programmable switches that allows NoC platform to dynamically change. Based on the communication requirements of an application the work proposed in [5] dynamically configures the switching, routing, packet size during run-time. However, these approaches either modify the network topology or change the switching logic to achieve reconfiguration in NoC. In contrast, approach followed in [6] have fixed the network topology i.e. mesh and reconfiguration is achieved using multiplexers. The approach followed in [6] is promising compared to other approaches reported in the literature, but it is limited to mesh topology only. Mesh-of-Tree (MoT) topology has more advantages compared to mesh topology and its detailed architecture, addressing scheme has been outlined in [7]. With this background, to leverage the advantage of reconfiguration and Mesh-of-Tree topology, this paper proposes a two phase Particle Swarm Optimization (PSO) based multi-application mapping for Mesh-of-Tree topology based NoC design. The rest of the paper is organized as follows. Section II details our proposed reconfigurable NoC architecture and methodology followed. Section III enumerates the experimental results. Section IV concludes the paper.

II. Proposed Architecture & Methodology

Proposed architecture: We propose a reconfigurable architecture for MoT topology. Fig. 1 shows the proposed architecture with multiplexers (2:1, 4:1) for 4x4 MoT topology. Communication cost is figure of merit for our approach, it is computed as the multiplication of number of hops and bandwidth requirements between the cores present in an application. As can be seen from Fig. 1, 'L', 'S', 'R' represents leaf, stem, and root level routers respectively. Each leaf router can accommodate two cores of an application. Multiplexers are added in between the clusters (C1, C2, C3, and C4) of leaf level routers to bring the flexibility of reconfiguration of cores to near by routers using local links. The reconfiguration is achieved to minimize the communication cost by decreasing the number of hops. The cores connected to L6, L7, L10, and L11, have 4:1 multiplexers connected to it with fourth input as high-impedance. The remaining cores other than extreme corners (L1, L4, L13, and L16) are connected to neighboring leaf routers via 2:1 multiplexers, as each core can connect to only two routers. Based on application requirements, the cores can be reconfigured to neighboring leaf routers via multiplexers.

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

2380-6923/19/$31.00 ©2019 IEEE
DOI 10.1109/VLSID.2019.00119

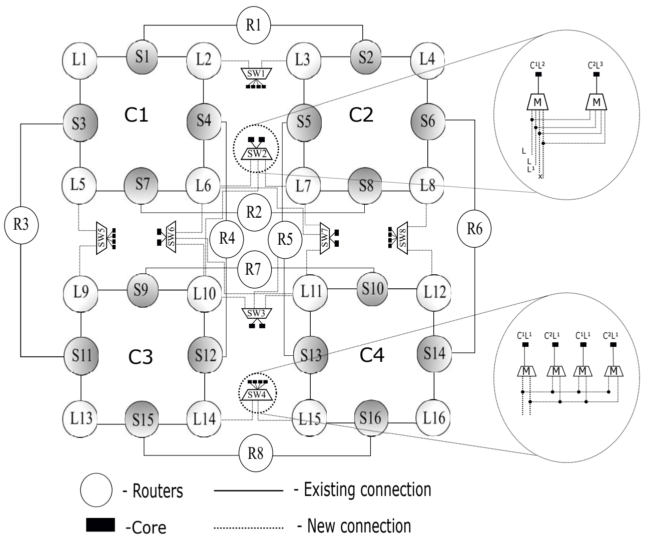

Fig. 1: Proposed reconfigurable NoC architecture

Methodology: A two phase Particle Swarm Optimization (PSO) [8] based technique has been proposed for multi-application mapping and reconfiguration. Detailed description of PSO can be found in [6]. Fig. 2 shows the particle structure used in our approach. It is an array, the contents of an array are core numbers while index of an array is router number in 4x4 MoT topology. The quality of a particle is defined by fitness function, i.e., communication cost in our approach. First, we perform mapping of the Combined Core Graph (CCG) generated by combining multiple applications, out of which few cores may be in common to multiple applications. Second, reconfiguration is performed by swapping the cores between the neighboring routers in 4x4 MoT topology. Particle structure remains the same for two phases, with difference being in position of cores before and after reconfiguration. For mapping, cores can connect to any router position, whereas in reconfiguration cores can choose from set of positions available in the reconfigurable NoC architecture governed by multiplexers.

C1	C3	C10	C7	C4	C2	C6	C9	C8	C5
1	1	2	2	3	3	4	4	5	5

Fig. 2: Particle structure

III. Experimental Results

In this section, we present the results of our experimentation performed on several application benchmarks and synthetic applications generated using TGFF tool [9]. Table I shows the communication cost comparison results obtained by performing reconfiguration for different set of applications. First column in Table I shows the application currently running on the network and the braces show complete application set of which it is a part of. Second column shows number of cores in that particular application and column 3, 4 show communication cost before and after reconfiguration. As can be seen in first set of applications MPEG and MWD, there is an improvement in communication cost of 35.82% and 19.19% after reconfiguration. This is due to the flexibility provided by multiplexers for mapping multiple applications onto 4x4 MoT network. This trend can also be seen in other set of applications. On an average, there is an improvement of 11.31% in communication cost after reconfiguration for multiple applications.

TABLE I: Comparison of communication cost for different set of applications before and after reconfiguration in 4x4 MoT network

Application	Cores	BR^a	AR^b	%Improvement
MPEG(MPEG+WMD)	12	14196	9110	35.82
MWD(MPEG+WMD)	22	6336	5120	19.19
263Enc(263Enc+263Dec)	12	871.36	727.87	16.46
263Dec(263Enc+263Dec)	14	118.11	81.13	31.30
MP3(MP3+263Enc+263Dec)	13	102.96	92.19	9.80
263Enc(MP3+263Enc+263Dec)	12	1003.12	850.29	15.25
263Dec(MP3+263Enc+263Dec)	14	96.81	90.51	5.73
G_1(G_1+G_2)	19	13298	10836	18.51
G_2(G_1+G_2)	11	12562	11741	6.53
G_1(G_1+G_3)	19	14434	12164	15.72
G_3(G_1+G_3)	10	3200	2688	16
G_2(G_2+G_3)	11	12234	7874	35.64
G_3(G_2+G_3)	10	2432	2048	15.79
			Average Improvement	11.31%

^{a}Before Reconfiguration, ^{b}After Reconfiguration

IV. Conclusion

In this paper, we have proposed reconfigurable architecture for MoT topology using two phase PSO based multi-application mapping and reconfiguration. Results show significant reduction in communication cost after reconfiguration for different applications. Future work includes proposing exact methods for the reconfiguration methodology.

References

[1] P. K. Sahu and S. Chattopadhyay, “A survey on application mapping strategies for network-on-chip design,” *Journal of Systems Architecture*, vol. 59, no. 1, pp. 60 – 76, 2013.

[2] M. B. Stensgaard and J. Spars, “Renoc: A network-on-chip architecture with reconfigurable topology,” in *Second ACM/IEEE International Symposium on Networks-on-Chip (nocs 2008)*, April 2008, pp. 55–64.

[3] M. Modarressi, H. Sarbazi-Azad, and A. Tavakkol, “An efficient dynamically reconfigurable on-chip network architecture,” in *Design Automation Conference*, June 2010, pp. 166–169.

[4] M. Modarressi, A. Tavakkol, and H. Sarbazi-Azad, “Application-aware topology reconfiguration for on-chip networks,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 19, no. 11, pp. 2010–2022, Nov 2011.

[5] B. Ahmad, A. T. Erdogan, and S. Khawam, “Architecture of a dynamically reconfigurable noc for adaptive reconfigurable mpsoc,” in *First NASA/ESA Conference on Adaptive Hardware and Systems (AHS’06)*, June 2006, pp. 405–411.

[6] S. J., A. Sharma, and S. Chattopadhyay, “Multi-application network-on-chip design using global mapping and local reconfiguration,” *ACM Trans. Reconfigurable Technol. Syst.*, vol. 7, no. 2, pp. 7:1–7:24, Jul. 2014. [Online]. Available: http://doi.acm.org/10.1145/2556944

[7] P. K. Sahu, A. Sharma, and S. Chattopadhyay, “Application mapping onto mesh-of-tree based network-on-chip using discrete particle swarm optimization,” in *2012 International Symposium on Electronic System Design (ISED)*, Dec 2012, pp. 172–176.

[8] K.-P. Wang, L. Huang, C.-G. Zhou, and W. Pang, “Particle swarm optimization for traveling salesman problem,” in *Proceedings of the 2003 International Conference on Machine Learning and Cybernetics (IEEE Cat. No.03EX693)*, vol. 3, Nov 2003, pp. 1583–1585 Vol.3.

[9] R. P. Dick, D. L. Rhodes, and W. Wolf, “Tgff: task graphs for free,” in *Hardware/Software Codesign, 1998. (CODES/CASHE ’98) Proceedings of the Sixth International Workshop on*, Mar 1998, pp. 97–101.