ENERGY- AND RELIABILITY-AWARE MAPPING FOR NOC-BASED ARCHITECTURES

B. NARESH KUMAR REDDY
Department of Electronics and Communication Engineering
National Institute of Technology, Goa. Email : nareshkumarreddy123456@gmail.com

Abstract: *There has been broad research on Task scheduling and mapping on a Multi-Processor System on Chip (MPSoC). In any case, none considers the optimization of communication types, which can fundamentally in uence communication energy and performance. Proposed work targets ongoing ap-plications which are mapped dynamically on NoC based architectures. We address precisely the energy and reliability aware mapping issues for NoC based architectures and propose an e cient technique to solve it. Moreover, the proposed technique takes into consideration new applications to be added to the system with insigni cant inter-processor communication overhead. Ex-perimental results shows that proposed technique gains not only a signi cant reduction in communication energy but also improvement in reliability.*

I. INTRODUCTION

As technology advances, we have been encountering increasingly integration of chips, and therefore System on Chip (SoC) has turned into a major pattern in popularized items since 1980s. It is probable that more than several pro-cessors will be integrated on a single chip [1]. Indeed, many researchers are as of now building many core architectures. Moreover, the complexity of running application on a chip will be extremely high, then it supports high bandwidth communication and high resolution. Hence communication between proces-sors/cores gets more critical, from this point of view Networks-on-Chip (NoC) may be a promising interconnect solution for complicated chips consisting of the many heterogeneous intellectual property (IP) cores [2]. The NoC solution conveys a networking approach to on-chip communication and gives outstand-ing enhancement as far as reliability and performance over the conventional
bus based structures (e.g., AMBA bus). While it is as of now di cult to ful ll each energy and reliability necessities for a faults at mapping cores on NoC.

In this paper, we target real time applications represented as task graphs. These applications are mapped onto NoC based processors. It proposes ERAM - Energy and Reliability Aware Mapping algorithm, in the mapping algorithm cores are mapped on NoC platform with respect to communication energy. During mapping cores any faults occur e ciently migrated tasks from failed core to nearest free core with respect to communication energy. Simulation results not only show the communication energy reduction of the ERAM, but also show the advantages of ERAM in terms computation time, reliability and throughput [3].

The remainder of this paper is planned as follow. First we review the related work and novel contribution in Section II and noti cation of application char-acteristics, NoC platform and a motivational example are explained in Section III, in Section IV, we formulate the problem of run-time mapping. Then we propose an ERAM algorithm. Experiment results are shown in Section V, and Section VI concludes the paper.

II RELATED WORK AND NOVEL CONTRIBUTION

We refer the reader to a few recent surveys on core mapping NoCs for pointers to modern research and improvement. Srinivasan Murali and et al., proposed NMAP algorithm, which maps the cores onto NoC architectures with respect to bandwidth and it is presented both single minimum-path routing and split tra c routing. It contains three phases, initial one is mapping generation, based on a which module has highest number of communication, next one is nd shortest path using Dijkstrs algorithm and nally iterative improve-ment, by changing pairs of modules and recomputing shortest paths [4]. In [5] Chen-Ling Chou and Radu Marculescu presented an integer linear program-ming (ILP) formulation of the contention-aware application mapping, which objects minimizing the inter-tile network contention. To solve the scalability problem caused by ILP formulation. The energy and performance aware incre-mental mapping problem for NoCs with multiple voltage levels proposed by Chen-Ling Cho and Umit Y. Ogras [6], explained a run-time approach to incre-mentally map a real application onto the NoC with multiple voltage levels. In [7] Jinho Lee and et al., described mapping and scheduling of tasks on a Multi Processor SoC, and decide the sort of every communication between message passing and shared memory when we do the mapping and scheduling. Leibo Liu and et al., [8] proposed a exible energy and reliability aware application mapping approach for network-on chip (NoC)-based recon gurable architec-ture, applications are mapped on various NoC platforms and running

with numerous routing algorithms when considering both energy and reliability.

Chen-Ling Chou et al., illuminated the spare core placement [9], the place-ment of the spare core was randomly assigned in the system. Lastly, spare core placement depends not only on the minimum distance between the faulty core and spare core but also on the manner of failure propagation avoidance over the rest of the system. Fatemeh Khalili and Hamid R. Zarandi [10], discussed spare core placement based on resource management, which enhanced failure containment inside the system. Finally, FASA reduces the fault contamination area and is applicable to distributed core graphs.

Compared to related work, in this paper focus on the scheduling and map-ping on a NoC based multi-processor system on chip. We show that on-going applications are dynamically mapped and e cient algorithm used when faults occur during faults on mapped cores. Furthermore, our mapping solution can be used to improve the reliability. Last but not least, this algorithms can be applied to any reliable mapping applications.

III PRELIMINARIES

3.1 Application Characteristics

The approaching applications are depicted by the application core graph (ACG), which is generated by [11]. Each ACG = (C, E) (see Fig. 1) is a coordinated diagram and contains the following [12].

1. Cores. Each Core C_i 2 C contains an arrangement of tasks from an o ine task partition process. The tasks having a same core are assigned to the same PE.

2. Edges. Each edge e_{ij} 2 E corresponds communication between the two cores C_i and C_j. Where Core C_i is any neighbor of core C_j. Weights W (e_{ij}) characterize the communication volume (bits) between core C_i and C_j.

3.2 NoC Platform

The platform is a two-dimensional (2D) mesh (mxn) Network on Chip, which contains cores, routers, core interface and link between cores. There are three kinds of cores namely Processing Core, Spare Core and Manager Core [13]. Processing cores deals with task associated with application. These processing cores can be failed or non-failed and every non-failed core can be either free or busy. Spare core is an extra core that can be used when any processing cores or manager cores fails [14]. Manager core manages and tracks all processing cores in the NoC, and do the task migration if a processing core fails. These cores are similar in nature but has di erent functionality in a given time.

NoC prefers the mesh topology, this topology graph can be uniquely de-scribed TG(N, D) is a directed graph. Where `N' represents a node or tile in the topology $8t_{xy}$ 2 N, `t_{xy}' represents the x^{th} row and y^{th} column of the tile. `D' represents the communication distance $8N_{ij}$ 2 D and `N_{ij}' denotes the distance between node (N_i) and node (N_j).

Communication between the routers denoted as inter-tile communication and communication between the core and the router is denoted as intra-tile

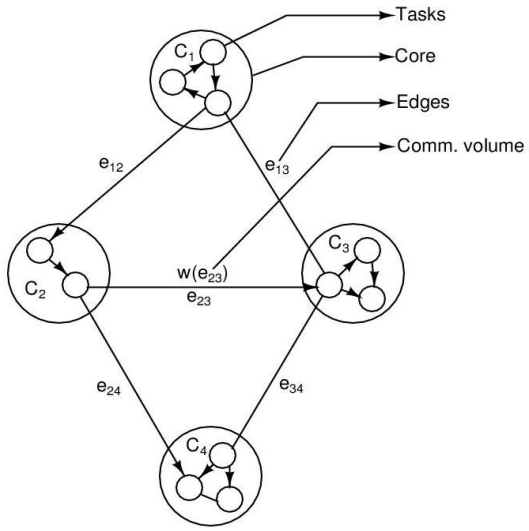

Fig. 1 Example of Application Core Graph

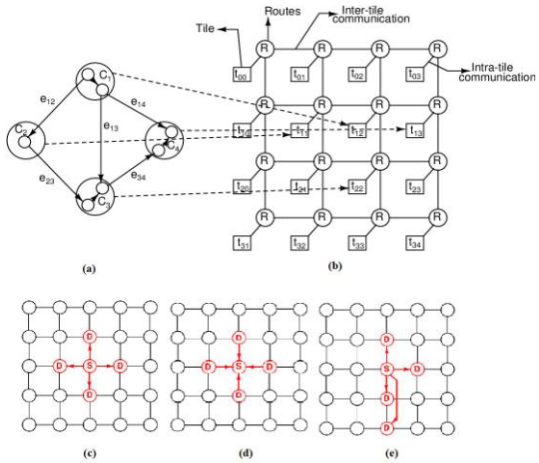

Fig. 2 (a) Application Core Graph, (b) 4x4 NoC Platform, (c) source based communication, (d) destination based communication and (e) path based communication.

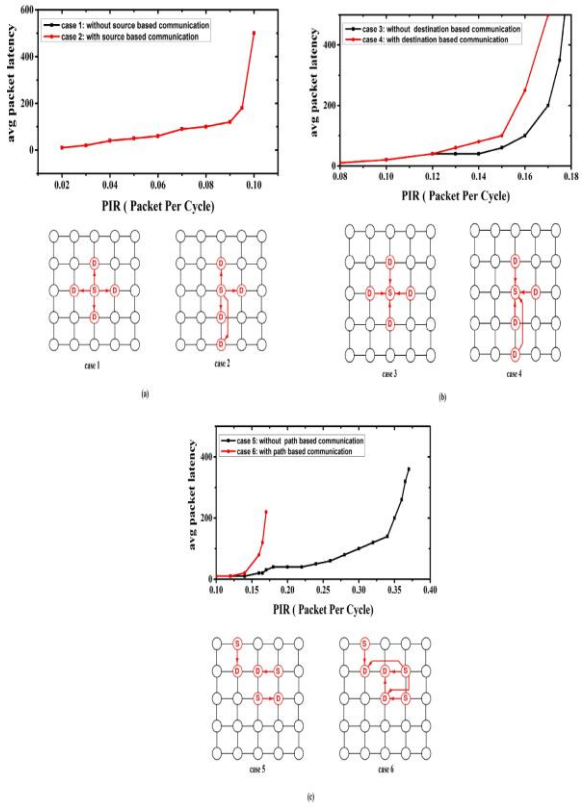

Fig. 3 The (a) source based, (b) destination based and (c) path based communication impact on average packet latency.

communication. Rough mapping of core graph to the NoC platform, source-based, destination-based and path-based communication clearly shown in Fig. 2.

3.3 Motivational Example

To demonstrate the impact of the source-based, destination-based, and path-based network communication on the packet latency, we consider an example mapping con guration (see Fig. 3.) in a 5X5 mesh NoC: with-out/with only source-based communication (case 1 vs. case 2), with-out/with only destination-based communication (case 3 vs. case 4), with-out/with only path- based communication (case 5 vs. case 6). Data transmission with 8 bits per packet and routing algorithm chosen XY. Fixed injection rate and data transmission rate is same to all con gurations. We did nearly 100 various experiments and cal-culated average packet latency: latency is calculated the time when packet is transferred from source to destination. The total experiment results shown in Fig. 3, X- axis is packet injection rate and Y-axis is average packet latency. As seen in Fig. 3(a), Fig. 3(b) and Fig. 3(c) represents source based, destination based and path based communication. Moreover we observe that the frequency of occurrence of the path-based communication much higher contrasted with the source-based and destination-based communication as the size increases. By doing a few experiments, we observed that the ratio of path-based to source-based communication and the ratio of path-based to destination-based communication increase linearly with the network size. Therefore, we concen-trate on minimizing the path based communication since this has the most signi cant impact on the packet latency and can be alleviated over the map-ping process.

IV. ENERGY AND RELIABILITY ORIENTED MAPPING

4.1 Problem Formulation

Given an application core graph and the NoC design, our goal is to map the cores onto the NoC platform such that the total weighted communication energy and path-based communication network minimized under a mapping mechanism. More formally:

4.1.1 Weighted Communication Energy (WCE)

Energy is directly proportional to the distance.

$$E / D$$

The distance function calculates the distance that would be traveled to get from one position to the other position if a square grid-path is followed. The distance between two positions is the sum of the di erences of their equivalent modules [19]. Distance between two cores (C_i, C_j). C_i parameters (a_1, b_1), C_j parameters (a_2, b_2).

$$\text{Distance} = j(a_1 \quad a_2)j + j(b_1 \quad b_2)j$$

Communication Energy is directly proportional to the distance between nodes, and is denoted by `E_{ij}'

$$E_{ij} / j(a_1 \quad a_2)j + j(b_1 \quad b_2)j$$

$$E_{ij} = e_{ij} \quad fj(a_1 \quad a_2)j + j(b_1 \quad b_2)jg$$

Where e_{ij} is a constant, which denotes the communication rate from C_i to C_j. E_{ij} is also called weighted communication energy [15], [16].

Problem 1: Given an Application Core Graph (C, E) and NoC Topology Graph (N, D); Find a mapping function : C ! N with $8C_i$ 2 C and the total communication energy.

such that :

$8C_i$ 2 C, $8N_i$ 2 N:
(C_i) 2 N,
C_i 6= C_j =) (C_i) 6= (C_j)
$8e_{ij}$ 2 e
$8N_{ij}$ 2 D
Let $8C_i$ 2 C be mapped to
some t_{xy} 2 N. then t_{xy} = (C_i)
2 N.
Communication energy is calculated using Eq. (4).
E_{ij} = e_{ij} fj (a_1 - a_2) j + j (b_1 - b_2) jg
^{E}ij = ^{e}ij ^{N}ij

N_{ij} denotes the distance between node (N_i) and node (N_j). N_i parameters (a_1, b_1), N_j parameters (a_2, b_2). e_{ij} denotes the communication rate from C_i to C_j.

Total communication energy (E)= $\sum_{C_i 2 C}^{8^X}$ ^{e}ij ^{N}ij: (5)

Problem 2: Given an Application Core Graph (C, E) and NoC Topology Graph (N, D); Find a mapping function : C ! N with $8C_i$ 2 C and the minimum communication energy. Which

$$E = \sum_{C_i 2 C}^{8X} {}^{e}ij \; {}^{N}ij$$

such that :

By using this de nition, the problem of mapping can be formulated as follows. Given an ACG and an NoC topology that satisfy

size (ACG) size (NoC topology)

map(C_i) 2 N,

Problem 1 clearly explained mapping and communication energy. The min-imum communication energy is obtained by

min E : initial mapping
f(E) = (6) min fe: e 2 Eg : changing mapping

4.2 Proposed Mapping Heuristic

We propose the mapping of given application cores on a NoC Platform. Our goal to map the cores onto the NoC platform such that the total weighted communication energy and path based communication are minimized.

Algorithm 1 CoreMapping

Input: Let C be the set of cores;

Let NoCP be the Given NoC Platform;

Output: Mapped NOC Platform ;

for Iter MaxIter do
  generate positions of empty cores randomly;
  map each c 2 C in orderly;
  calculate total communication energy;
  min TotalCommunicationEnergy ;
  if min > TotalCommunicationEnergy then
    m
    in TotalCommunicationEnergy ;
    BestNoCMapping NoCMapping;
  end
end
Return BestNocMapping;

Using Algorithm 1, cores are e ciently mapped on NoC platform and min-imize the communication energy. If faults occur at the processing core, faulty core tasks are migrated to the free core in the NoC platform, which should be near the faulty core in the NoC platform. Faulty core mapping explained below.

Algorithm 2 FaultyCoreMapping

Input: ***Mapped NoC Platform;***

Output: ***Best Faulty Core Mapped on NoC Platform i.e., BestFaultyMapping;***

Let FC be the set of free cores available in a mapped NoC Platform; BestFaultyMapping= fg***;***
Identify the FaultyCore;

foreac
h fc 2 f FC g do
  fc FaultyCore ;
  FaultyMapping NoC Platform where F aultyCore ***mapped to fc ;***
  TotalCommunicationEnergy Compute total communication energy for FaultyMapping using Eq. (5);
  min TotalCommunicationEnergy ;
  if min > TotalCommunicationEnergy then
    min TotalCommunicationEnergy ;
    BestFaultyMapping FaultyMapping;
  end

end

Return BestFaultyMapping;

Bench mark	Number of IP's	Application
DVOPD	***32***	***Dual video object plane decoder***
VOPD	***16***	***Video object plane decoder***
MPEG4	***9***	***MPEG4 decoder***
PIP	***8***	***Picture in picture***
MWD	***12***	***Multi window display***
mp3encmp3dec	***13***	***Mp3 encoder and Mp3 decoder***
263encmp3dec	***12***	***H.263 encoder and Mp3 decoder***
263decmp3dec	***14***	***H.263 decoder and Mp3 decoder***
H.264	***14***	***H.264decoder***
HEVC	***16***	***High e ciency video coding decoder***
Freqmine	***12***	***Data mining application***
Swaption	***15***	***compute portfolio prices using Monte-Carlo simulation***
random1	***16***	***Generated by TGFF***
random2	***16***	***Generated by TGFF***

Table 1 Benchmarks Used in the Simulation

V. EXPERIMENT RESULTS

In this section we evaluate the performance and communication energy of our mapping algorithm (ERAM- Energy and Reliability Aware Mapping) and compare the existing mapping algorithms. The mapping pattern is found using a C++ program and computation time can be obtained as well. Simulations are performed on a Noxim simulator [17], [18].

Table 2 Communication Energy and Throughput comparison ERAM with FARM and ***FASA***

W	ERAM is compared with FARM					ERAM is compared with FASA				
number of edges										
communication energy conservation										
throughput improvement										

evaluate the contention impact on core graph on 5x5 NoC platform, several set of synthetic applications are generated using TGFF [11]. In this experiment number of cores are used 12-16 and edges are 10-50. Fourteen commonly used application benchmarks, twelve are real applications and two are random benchmarks generated by TGFF shown in Table I. The compu-tation time consumed by ERAM and existing algorithms (FARM and FASA) to nd the best mapping is measured as an indicator for the computational complexity. Reliability and energy consumption comparison results of ERAM and existing algorithms are shown in Fig. 4. Reliability and energy consump-tion comparison results of ERAM and existing algorithms under a faulty core environment as shown in Fig. 5, and Fig. 6.

As it can be found in Table 2, the communication energy conservation is up to 14% ERAM is compared with FARM and 10% ERAM is compared with FASA. System throughput can be improved by 22.36% on average ERAM is compared with FARM and 15.34% ERAM is compared with FASA. It clearly shows ERAM e ectively reduces communication energy and great sys-tem throughput improvements.

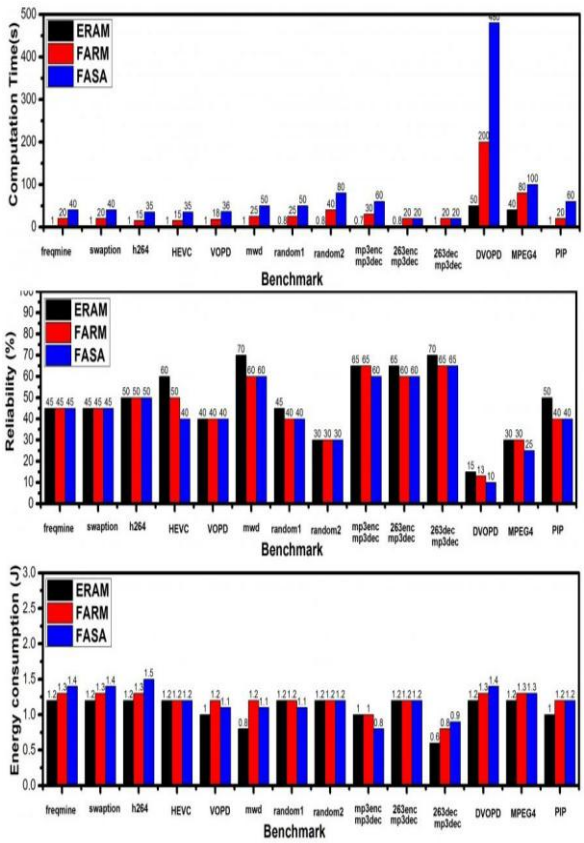

Fig. 4 Comparison of Computation Time, Reliability and Energy consumption of ERAM with FARM & FASA

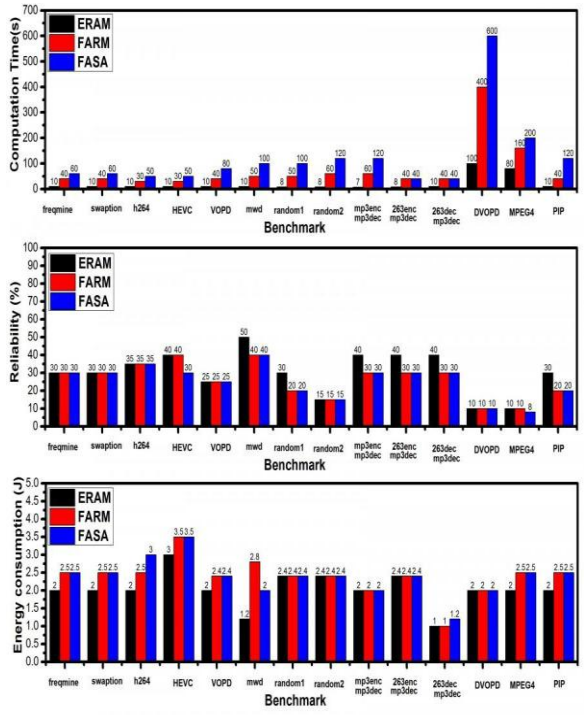

Fig.5 Comparison of Computation Time, Reliability and Energy consumption of ERAM with FARM & FASA under single fault

VI CONCLUSION

In this paper, we have addressed a runtime approach to map a number of applications onto a NoC based multiprocessor system on chip. We have re-ported our results achieved from many tests including both synthetic and real benchmarks. The outcomes indicate signi cant reduction on communication energy and improvement on system throughput. Finally mapping results of our algorithm can be obtained very ecient.

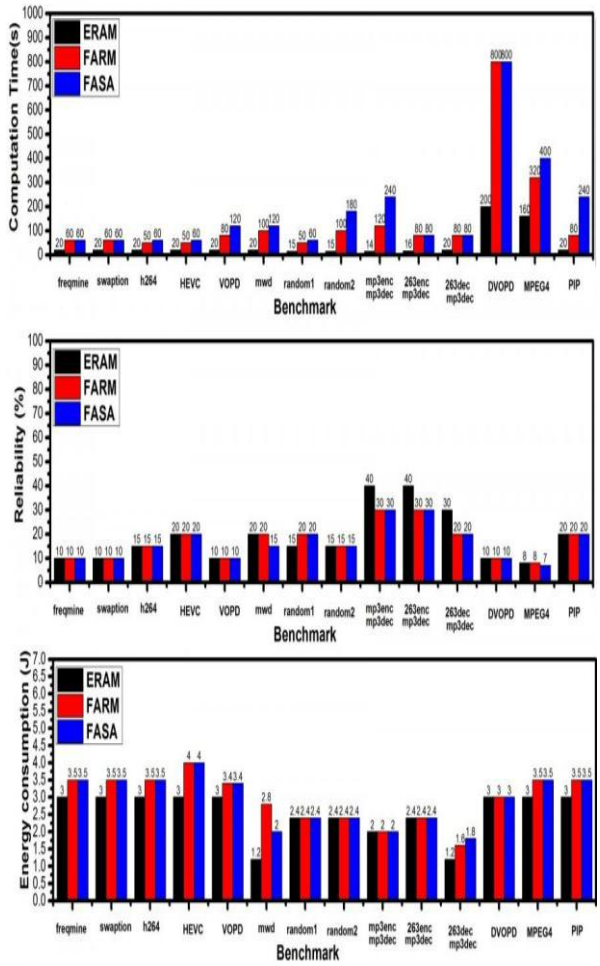

Fig. 6 Comparison of Computation Time, Reliability and Energy consumption of ERAM with FARM & FASA under two faults

ACKNOWLEDGEMENT

The authors thanks the Department of Electronics and Information Technol-ogy, Ministry of Communication & IT, Government of India and Media Lab Asia, to fund this work under Visvesvaraya PhD scheme.

REFERENCES

1. Luca Benini and Micheli G.D., *Networks on Chips: A New SoC Paradigm, IEEE Comput., Vol. 35, No. 1, pp. 70-78, Jan. 2002.*

2. D. Bertozzi and L. Benini., *Xpipes: A Network-on-Chip Architecture for Gigascale Sys-tem on Chip, In IEEE Circuits and Systems, Vol. 4, No. 2, pp. 18-31, 2004.*

3. Teijo Lehtonen, P. Liljeberg and J Plosila, *Fault Tolerance Analysis of NoC Architectures, IEEE International Symposium on Circuits and Systems, pp. 361-364, 2007.*

4. Srinivasan Murali, Giovanni De Micheli, *Bandwidth-constrained mapping of cores onto NoC architectures, Proceedings Design Automation and Test in Europe Conference and Exhibition, 2004.*

5. Chen-Ling Chou and Radu Marculescu, *Contention-aware Application Mapping for Network-on-Chip Communication Architectures, IEEE International Conference on Com-puter Design, pp. 164-169, 2008.*

6. Chen-Ling Chou, UmitY.Ogras and Radu Marculescu, *Energy- and Performance-Aware Incremental Mapping for Networks on Chip With Multiple Voltage Levels, IEEE Trans-action on Computer-Aided Design of Integrated Circuits and Systems, Vol. 27, No. 10, October 2008.*

7. Jinho Lee, Moo-Kyoung Chung, Yeon-Gon Cho, Soojung Ryu,Jung Ho Ahn and Kiyoung Choi, *Mapping and Scheduling of Tasks and Communications on Many-Core SoC Under Local Memory Constraint, IEEE Transaction on Computer-Aided Design of Integrated Circuits and Systems, Vol. 32, No. 11, November, 2013.*

8. Leibo Liu, Chen Wu, Chenchen Deng, Shouyi Yin, Qinghua Wu, Jie Han, and Shao-jun Wei, *A Flexible Energy- and Reliability-Aware Application Mapping for NoC-Based Recongurable Architectures, IEEE Transaction on Very Large Scale Integration (VLSI) Systems, 2014.*

9. Chen-Ling Chou and Radu Marculescu, *FARM: Fault-Aware Resource Management in NoC based Multiprocessor Platforms, Design Automation& Test in Europe Conference & Exhibition (DATE), 2011.*

10. Fatemeh Khalili and Hamid R. Zarandi, *A fault-tolerant core mapping technique in networks-on-chip, IET Comput. Digit. Tech, Vol. 7, Iss.6, pp. 238-245, 2013.*

11. *Task graphs for free (TGFF) Available: http://ziyang.eecs.umich.edu/~dickrp/tg /*

12. B. Naresh Kumar Reddy, M.H.Vasantha, Y.B.Nithin Kumar and Dheeraj Sharma, *A Fine Grained Position for Modular Core on NoC, IEEE International Conference on Com-puter, Communication and Control (IC4), pp. 1-4, 2015.*

13. V. Nollet, T. Marescaux and D. Verkest, *Operating System Controlled Network on Chip, Design Automation Conference, pp. 256-259, 2004.*

14. Fatemeh Khalilia and Hamid R. Zarandi, *A Fault-Tolerant Low-Energy Multi-Application Mapping onto NoC-based Multiprocessors, IEEE 15th International Confer-ence on Computational Science and Engineering, pp. 421- 428, 2012.*

15. *http://grokbase.com/t/lucene/mahout-dev/082festv1e/weighted-manhattan-distance-metric.*

16. B. Naresh Kumar Reddy, M.H.Vasantha, Y.B.Nithin Kumar and Dheeraj Sharma, *Com-munication Energy Constrained Spare Core on NoC, 6th International Conference on Computing, Communication and Networking Technologies (ICCCNT), pp. 1-4, 2015.*

17. *Noxim the NoC simulator Available: https://github.com/davidepatti/noxim*

18. B. Naresh Kumar Reddy, Vasantha.M.H. and Nithin Kumar Y.B., *A Gracefully De-grading and Energy-E cient Fault Tolerant NoC Using Spare core, 2016 IEEE Computer Society Annual Symposium on VLSI (ISVLSI 2016), Pennsylvania, U.S.A., pp. 146-151, 2016.*