Journal of Network and Computer Applications 105 (2018) 79–87

Contents lists available at ScienceDirect

Journal of Network and Computer Applications

journal homepage: www.elsevier.com/locate/jnca

An energy-efficient fault-aware core mapping in mesh-based network on chip systems

Naresh Kumar Reddy Beechu [*], Vasantha Moodabettu Harishchandra, Nithin Kumar Yernad Balachandra

Department of Electronics and Communication Engineering, National Institute of Technology Goa, India

ARTICLE INFO

Keywords:
Network on chip (NoC)
Core
Node average distance
Placing unmapped vertices region
Failure rate

ABSTRACT

A fault aware core mapping on Network on Chip (NoC) requires an understanding the calculation of the functional metrics. The functional metrics are Node Average Distance (NAD), Placing unmapped Vertices Region (PVR) and Weighted Communication Energy(WCE). The traditional method of calculating functional metrics are using unmapped cores, distance and weight. However finding the exact values of the functional metrics is not always easy, particularly when busy cores and failed cores are mapped in the middle of available cores. This paper proposes an energy efficient fault aware core mapping algorithm that maps the cores onto the NoC under communication rate constraints to minimize the total communication energy using functional metrics. The simulation results show that proposed mapping algorithm has higher performance over BMAP, PMAP and NMAP algorithm present in literature by 25%, 42%, and 65% under single fault, 30%, 40%, and 60% under two faults and 40%, 44%, and 55% under four faults respectively.

1. Introduction

Numerous interconnection systems have been examined for node connections in parallel or circular. It is very hard to contrast, such systems with their suitability for a specific application domain, where one has to look into large number of static characteristics and dynamic characteristics. Therefore, the Network on Chip (NoC) was introduced to handle communication characteristics of a chip design, such as flexibility, performance and energy efficiency (Parhami, 2013). In an NoC, different vertices can be mapped to different cores, if more than one vertex is assigned to a single core, vertices are scheduled on the priority basis (Coskun et al., 2006). In an NoC, cores are connected with routers via a network interface, and data transferred in the NoC in the form of messages, packets, and phits/flits. Cores are three types: 1. Processing core 2. Manager core, and 3. Spare core. Processing cores process tasks in a given application and can either be free or busy. Manager cores manage the running core and update the status of the core. Spare cores are used to recover data from failed cores.

In literature, vertices were mapped on an NoC platform by considering preliminaries such as weighted manhattan distance (WMD), link contention count (LCC), system fragmentation algorithm (SFA) and available neighbouring tiles (ANT) (Chou and Marculescu, 2011), (Khalili and Zarandi, 2013), (Khalilia and Zarandi, 2012). Calculating these functional metrics are difficult because they are dependent on the unmapped vertices, communication rate and distance. The mapping process is based on unmapped vertices and free cores on the NoC platform. If any faults occur during core mapping, the reliability of entire chip is reduced. Faults can be classified into one of three categories: Transient faults - These occur once and then disappear. For example, a network message doesn't reach its destination but does when the message is retransmitted. Intermittent faults - Intermittent faults are characterized by a fault occurring, then vanishing again, then reoccurring, then vanishing. These can be the most annoying of component faults. A loose connection is an example of this kind of fault. Permanent faults - This type of failure is persistent: it continues to exist until the faulty component is repaired or replaced. Examples of this fault are core crashes. To overcome the errors caused by their faults, various fault tolerance methods are used and applied at the system level. A few strategies such as for error detecting/correcting codes, remapping strategy, and spare cores are used to improve reliability at the system level (Khalilia and. Zarandi, 2013).

In this paper, an energy efficient fault aware core mapping on NoC is proposed. Which encompass a mapping algorithm and faulty core mapping. In the mapping algorithm, initially identify the dynamic region

* Corresponding author.
E-mail address: naresh.nitg@gmail.com (N.K.R. Beechu).

https://doi.org/10.1016/j.jnca.2017.12.019
Received 20 September 2017; Received in revised form 6 November 2017; Accepted 23 December 2017
Available online 29 December 2017
1084-8045/© 2018 Elsevier Ltd. All rights reserved.

and vertices are mapped on dynamic region on the basis of preliminaries such as weighted communication energy (WCE), nodes average distance (NAD) and placing unmapped vertices region (PVR). When permanent faults occur in processing cores, it identifies the spare core efficiently in terms of the number and their positions for every incoming application using faulty core mapping. In this study, free cores were dedicated as spare cores, which serve as alternatives in case of core failure during operation. The proposed technique conserves communication energy and enhances the system performance. The proposed technique is based on many application core graphs (ACG) generated by the Task Graph For Free (TGFF). The proposed technique enhances system performance and conserves communication energy depending on various fault injection tests. In order to accurately estimate the power, area and performance of the proposed EMAP algorithm, simulation tool Vivado 2016.4 is used in this work.

The remaining part of this paper is as follows. Related work presented in literature is discussed in Section 2. Section 3 explains mathematical formulations and core mapping on NoC is proposed in Section 4. Mapping algorithm and faulty core mapping algorithm with examples are described in Section 5 and Section 6. Experimental results are discussed and presented in Section 7, followed by conclusion in Section 8.

2. Related work

Addressing different core mapping algorithms, numerous strategies have been proposed to improve the performance of NoC system. Nectarios Koziris and et al., presented physical mapping of clustered task graphs onto multiprocessor architectures. The task graph is mapped on nodes of processor network to reduce complexity (Koziris et al., 2000). In Ref. Murali and De Micheli (2004) Srinivasan Murali and et al., proposed NMAP algorithm, which maps the cores onto NoC architectures with respect to bandwidth and presented both single minimum-path routing and split traffic routing. It contains three phases, first one is mapping generation, based on module which has highest number of communication, next one is find shortest path using Dijkstra's algorithm and finally iterative improvement, by changing pairs of modules and recomputing shortest paths. A simple technique for low energy mapping and routing in NoC architectures by Krishnan Srinivasan and Karam S. Chatha (Srinivasan and Chatha, 2005), which can solve the mesh based NoC design problem with an objective of minimizing the communication energy. In Ref. Shen et al. (2007), described mapping algorithm which consequently maps a given set of property onto a NoC and builds a deadlock-free deterministic routing function for performance and minimizes the communication energy.

Wein-Tsung Shen et al., proposed a well-organized binomial core mapping and optimization algorithm (BMAP) to reduce the hardware cost of on-chip network (OCN) infrastructure. BMAP complexity is O(N^2log(N)) and provides more economical when compared to traditional mapping techniques and it is a fast and efficient technique when compared to NMAP (Rahmati et al., 2013). Wei Hu et al. (2009) discussed fast algorithm, which maps the cores onto a WK-Recursive NoC under Performance Constraints. A new energy efficient and bandwidth aware topological core mapping on NoC platform presented by Saurabh Agrawal et al. (2010), which has significant reduction in execution time. Nithin Michael et al., proposed mapping of the applications on NoC nodes (processors) with respect to communication cost. The basic idea presented here in that the minimizing communication cost translates to minimizing dynamic power and leads to better energy-efficiency (Michael et al., 2013).

While these papers examined the overall advantages and difficulties of the faulty aware core mapping on NoC. This paper proposes an energy efficient fault-aware core mapping in mesh based NoC. Previous studies focused mainly on the fault model notation for the core, detection of errors and determination of spare core position. Thus mapping algorithm and spare core placement not only reduces fault contamination area but also conserves energy and enhances performance. Moreover it is applicable to both random and distributed core graphs.

3. Mathematical formulations

We propose reliable core mapping onto the mesh-based NoC platform by using functional metrics and optimizing formulas.

3.1. Mathematical formulation of the mapping problem

By using the typical notation (Naresh Kumar Reddy et al, 2016), the communication rate between the vertices is denoted by a core graph (Beechu et al, 2017):

Definition 1. An Application Core Graph is a directed graph ACG(V, E). Where 'V' represents Vertex, each directed edge 'e_{ij}' in E characterizes the communication from vertex V_i to vertex V_j, while 'W(e_{ij})' characterizes communication rate from vertex V_i to vertex V_j.

According to Ref. Beechu et al (2017), the connectivity and communication of the NoC is represented by a topology graph:

Definition 2. NoC prefers the mesh topology, this topology graph can be uniquely described TG(N, D) is a directed graph. Where 'N' represents a node or tile in the topology $\forall t_{xy} \in$ N, 't_{xy}' represents the xth row and yth column of the tile. 'D' represents the communication distance $\forall N_{ij} \in$ D and 'N_{ij}' denotes the distance between node (N_i) and node (N_j).

3.2. Functional metrics

3.2.1. Nodes average distance (NAD)

NAD is the shortest path between any two nodes in the network (Yang, 2013). The shortest path in a K-dimensional mesh can determined by including the distances of the destination node from the source node along each K-dimension.

We consider a linear array of N-path, as shown in Fig. 1. Here, we incorporate the initial length of the path from each node to itself in calculating the average consequently, the normalizing term $\frac{1}{N^2}$ is substituted for $\frac{1}{N(N-1)}$ and the distances from node i to all the nodes to its left and right are individually calculated.

$$\begin{aligned}\Delta_{N-path} &= \frac{1}{N^2}\sum_{0\le i\le N-1}\left[\sum_{0\le j\le i}(i-j)+\sum_{i\le j\le N-1}(j-i)\right]\\ &= \frac{1}{N^2}\sum_{0\le i\le N-1}\left[i(i+1)-i(i+1)/2+(N-i)(N-1+i)/2-i(N-i)\right]\\ &= \frac{1}{N^2}\sum_{0\le i\le N-1}\left[i^2-(N-1)i+N(N-1)/2\right]\\ &= \frac{1}{N^2}\left[N(N-1)(2N-1)/6-N(N-1)^2/2+N^2(N-1)/2\right]\\ &= \frac{1}{3}[N-1/N]\end{aligned}$$

The average node distance in a q-dimensional mesh is as follows:

$$\Delta_{q-mesh} = \frac{1}{3}\left[\sum_{0\le i\le q-1}(n_i-1/n_i)\right]$$

The average node distance in a 2-dimensional mesh is:

$$\Delta_{2-mesh} = \frac{1}{3}[(n_1-1/n_1)+(n_2-1/n_2)]$$

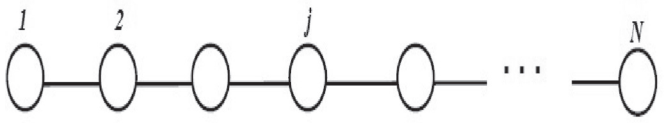

Fig. 1. Linear array with N nodes (N-path).

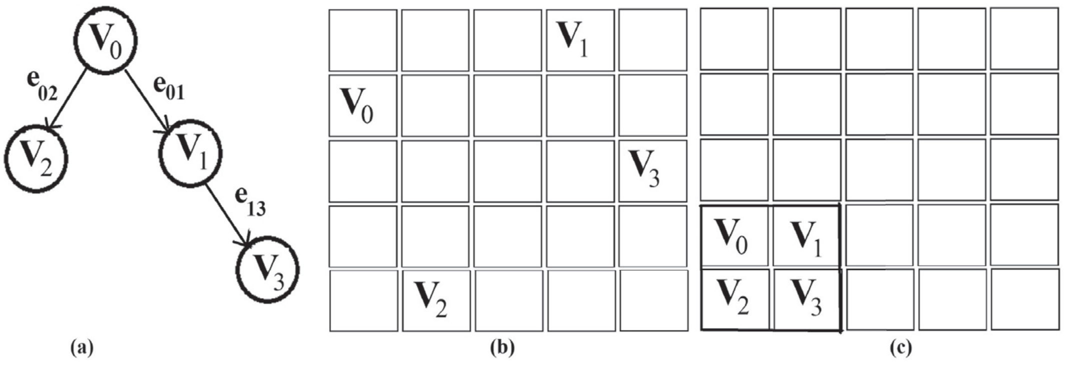

Fig. 2. Mapping region (a) ACG (b) Rough mapping (5 × 5) (c) effective mapping (2 × 2).

The average distance between the selected nodes on an NoC, with size X × Y.
The NAD is computed as given in (1).

$$\text{NAD} = \frac{1}{3}\left[(\text{X} - 1/\text{X}) + (\text{Y} - 1/\text{Y})\right]$$

$$NAD = \frac{X+Y}{3} \times \left(1 - \frac{1}{XY}\right) \tag{1}$$

For example, Fig. 2, shows an NAD mapping region. The vertices are mapped roughly in a region with size 5 × 5 and efficiently in a region size 2 × 2. The NAD of the rough mapping is 3.2 compared with 1 for the most effective mapping using (1). Hence, to achieve minimum communication energy, the region with minimal NAD should be selected for mapping an application.

3.2.2. Placing unmapped vertices region (PVR)

Placing unmapped Vertices Region depends upon the unmapped core and is computed as given in (2).

Unmapped core is denoted as 'C_{um}'and unmapped Vertices denoted as 'V_{um}'

$$C_{um} > V_{um} \tag{2}$$

$$C_{um} = V_{um} + S \quad (S \in [1, n])$$

The number of free cores should be greater than the number of vertices in ACG for successfully mapping of the ACG for each NoC size.

3.2.3. Weighted communication energy (WCE)

WCE between two vertices is calculated as given in (3) (Naresh Kumar Reddy et al., 2015a).

$$WCE(V_i, V_j) = W(e_{ij}) \times |I_{V(i)V(j)}| \tag{3}$$

W(e_{ij}) is communication rate between two vertices (V_i, V_j).
$I_{V(i)V(j)}$ is link or communication distance between two vertices (V_i, V_j). V_i parameters (a_1, b_1) and V_j parameters (a_2, b_2).

$$|I_{V(i)V(j)}| = |(a_1 - a_2)| + |(b_1 - b_2)| \tag{4}$$

Problem 1. Given an Application Core Graph (V, E) and NoC Topology Graph (N, D); Find a mapping function Ω: V → N with $\forall V_i \in$ V and calculate the total communication energy.

such that:

$$\forall V_i \in V, \forall N_i \in N.$$

$$\Omega(V_i) \in N,$$

$$V_i \neq V_j \implies \Omega(V_i) \neq \Omega(V_j)$$

in Definition 2

$\forall N_{ij} \in D$

Let $\forall V_i \in$ V be mapped to some $t_{xy} \in$ N.

then $\quad t_{xy} = \Omega(V_i) \in$ N.

Communication energy is calculated using Eq. (3).

$$\text{WCE}(V_i, V_j) = \text{W}(e_{ij}) \times |I_{V(i)V(j)}|$$

in terms of nodes

$$\text{WCE}(V_i, V_j) = \text{W}(e_{ij}) \times N_{ij}$$

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). $W(e_{ij})$ denotes the communication rate from V_i to V_j. Total communication energy

$$T_{WCE} = \Sigma_{\forall(i,j)}(W(e_{ij}) * N_{ij}) \tag{5}$$

Problem 2. Given an Application Core Graph (V, E) and NoC Topology Graph (N, D); Find a mapping function Ω: V → N with $\forall V_i \in V$ and calculate the minimum communication energy.

such that:

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

size (ACG) ≤ size (NoC topology)

map(V_i) ∈ N,

Problem 1 clearly explained mapping and communication energy. The minimum communication energy is obtained by f(T_{WCE})

$$= \left\{ \begin{array}{c} \min T_{WCE} \ : \text{initial mapping} \\ \min \{t_{wce} : t_{wce} \in T_{WCE}\} \ : \text{changing mapping} \end{array} \right\} \tag{6}$$

4. Core mapping on NoC

In this study, we focused on the assignment application problem with different communication rates and mapping on an NoC platform, which provides an optimal solution to an n × n assignment problem. This is an efficient problem that includes special case nodes (busy core and failed cores mapped on the NoC). We propose the use of a dynamic optimal region and mapping of given application vertices on a dynamic region in an NoC Platform (Naresh Kumar Reddy et al., 2015b).

In the mapping algorithm, initially identify the dynamic region, which is clearly explained in Algorithm 1. Dynamic region based on minimum NAD and PVR, select region based on communication energy when two regions having similar NAD and PVR.

Core mapping on dynamic region clearly explained in Algorithm 2. Initialize the mapped and unmapped vertices, dynamic region is selected according Algorithm 1 whose value corresponds to the smallest value when NAD and PVR values are considered (whichever is minimum). It reduces the communication energy and failure probability. A vertex with the highest communication rate is selected and the tile "t_{xy}" is first selected which should be a highest neighbouring free cores. The vertex to the first selected tile is mapped, if the tile is already mapped, then it is updated as "busy". Simultaneously the mapped and unmapped vertex sets are also updated. Rest of the vertices are mapped with the available cores based on the priority, which is determine by the communication rate i.e. high communication rate vertices are selected first, correspondingly neighbouring cores are mapped. Iterations are performed till all the vertices are mapped with corresponding cores. All vertices can be mapped in the dynamic region with minimum T_{WCE}. Then, we can obtain energy-efficient mapping.

Algorithm 1 IdentifyingDynamicRegion

```
Input: ACG, NoC Platform;
Output: Efficient Dynamic Region i.e., BestDR;
        Let DR be the Dynamic Region;

BestDR= {};
foreach  $dr \in \{ DR \}$ do
    Calculate NAD [use (1)]
    Calculate PVR [use (2)]
    Satisfying PVR in NAD region
end
Select dynamic region corresponds to minimum NAD & PVR ;
```

Algorithm 2 CoreMapping

```
Input:  Let V be the order set (descending order of communication
         rate) of Vertices;
        Let C be the set of cores;
        Let GDR be the Given Dynamic Region;
Output: Mapped Dynamic Region ;

foreach  $v \in \{ V \}$ do
    Find the core $c \in C$ such that $c$ is having maximum neighboring free
     cores in the GDR;
    if multiple cores are available then
        Find the maximum neighboring free core $c \in C$ by considering all
         neighboring cores (free, busy and failed cores) of the GDR;
    else
        Map the vertex $v$ onto $c$ in the GDR;
    end
    Update $V$ by eliminating $v$ ;
    TotalCommunicationEnergy ← Compute total communication energy
     for CoreMapping using (5);
    min ← TotalCommunicationEnergy ;
    if $min > TotalCommunicationEnergy$ then
        min←TotalCommunicationEnergy ;
        BestCoreMapping ← CoreMapping;
    end
end
Return BestCoreMapping;
```

5. Mapping algorithm: an example

We consider an example where five vertices (V_0, V_1, V_2, V_3, and V_4) should be mapped on an NoC Platform as shown in Fig. 3.

5.1. Finding the dynamic region

1. Find the minimum unmapped cores by using a PVR. In our example, five unmapped vertices exist on the basis of (2). The minimum number of free cores should be six.
2. Determine the effective mapping by using NAD. In our example, according to (1), a size of 2×3 is suitable.
3. Satisfy the minimum unmapped cores in the NAD. Here, if the unmapped cores are six, then efficient size of the NAD is 3×3, because it satisfies both minimum NAD and minimum PVR.
4. If different 3×3 regions are satisfying both minimum NAD and minimum PVR, above dynamic region algorithm clearly explained choose which dynamic region has minimum communication energy based on (6), output the resulting region as shown in Fig. 4.

5.2. Mapping of vertices in a dynamic region

1. Input vertices and dynamic regional nodes or cores in the matrix are vertically and horizontally as shown in Fig. 5.
2. Map the vertex in order beginning with the vertex with the maximum communication rate. In our example (Fig. 3(a)), the total vertices are five, V_1 has a communication rate of 900 (250 + 300 + 350) and V_4, V_3, V_0 and V_2 have communication rates 550, 500, 450 and 200 respectively. These values are clearly mentioned in the matrix as shown in Fig. 6.
3. Place the core in order with the core that first communicates with the maximum number of neighbouring free cores on a dynamic region. In our example, t_{11} and t_{12} have three neighbouring free cores, t_{01}, t_{02} and t_{10} have two neighbouring free cores, t_{20} and t_{22} have one neighbouring free core as shown in Fig. 7. When 5 unmapped vertices exist, select the first five unmapped cores, that communicate with the maximum number of neighbouring free cores on a dynamic region.
4. Map the vertices on a dynamic region.

   $V_1 \in \{t_{11} \,\&\, t_{12}\}$

   $V_4 \in \{t_{11} \,\&\, t_{12}\} - V_1$

   $V_3 \in \{t_{01},\, t_{10} \,\&\, t_{02}\}$

   $V_0 \in \{t_{01},\, t_{10} \,\&\, t_{02}\} - V_3$

   $V_2 \in \{t_{01},\, t_{10} \,\&\, t_{02}\} - \{V_3 \cup V_0\}.$

   Here, if the cores have the same number of neighbouring free cores, proceed to Step 5.

5. Place the core that has the maximum number of neighbouring cores (free, busy and failed cores) in a dynamic region. In our example, t_{11} has four neighbouring cores, t_{12} has three neighbouring cores, t_{01} and t_{10} have three neighbouring cores and t_{02} has two neighbouring cores as shown in Fig. 8.

   $V_1 \in t_{11}$

   $V_4 \in t_{12}$

   $V_3 \in \{t_{01} \,\&\, t_{10}\}$

   $V_0 \in \{t_{01} \,\&\, t_{10}\} - V_3$

   $V_2 \in t_{02}.$

   Here, if the cores have the same number of neighbouring cores, proceed to Step 6.

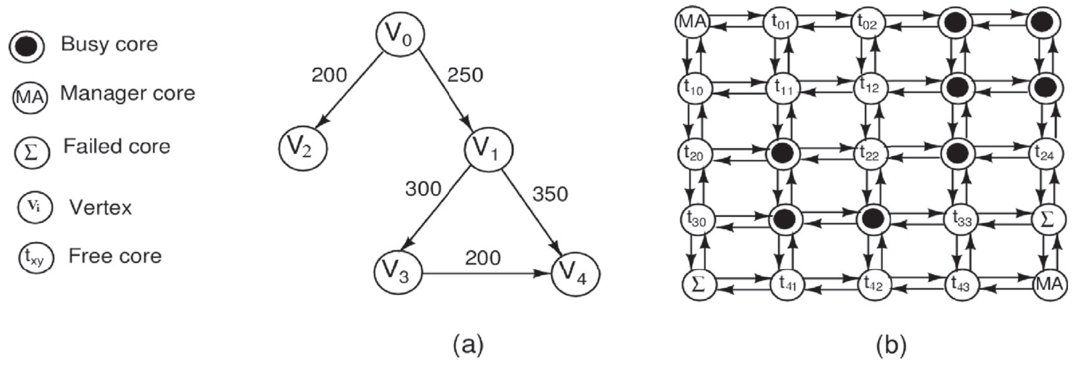

Fig. 3. Mapping an Application: (a) an example core graph, (b) 5 × 5 mesh NoC.

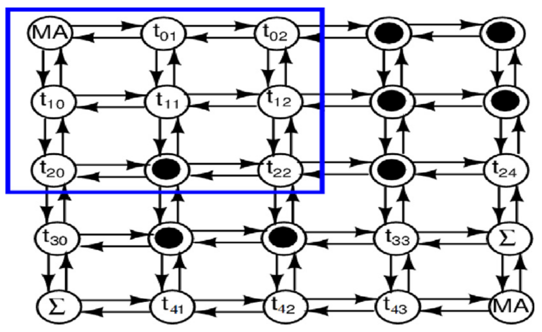

Fig. 4. NoC dynamic region.

Vertices	Cores						
	t_{01}	t_{02}	t_{10}	t_{11}	t_{12}	t_{20}	t_{22}
V_0							
V_1							
V_2							
V_3							
V_4							

Fig. 5. Initial mapping process.

Vertices	Cores							Communication Rate
	t_{01}	t_{02}	t_{10}	t_{11}	t_{12}	t_{20}	t_{22}	
V_1								900
V_4								550
V_3								500
V_0								450
V_2								200

Fig. 6. Arrangement of vertices in mapping process.

6. Even after performing step 5, t_{01} and t_{10} have the same number of communicating cores. In this condition in one case t_{01} has the highest priority and in another case t_{10} has the highest priority and communication energy is calculated.
   Case I:
   $V_1 \in t_{11}$
   $V_4 \in t_{12}$
   $V_3 \in t_{01}$
   $V_0 \in t_{10}$
   $V_2 \in t_{02}$ is as shown in Fig. 9.

Vertices	Cores							Communication Rate
	t_{11}	t_{12}	t_{01}	t_{02}	t_{10}	t_{20}	t_{22}	
V_1								900
V_4								550
V_3								500
V_0								450
V_2								200
Neighbouring Free cores	3	3	2	2	2	1	1	

Fig. 7. Arrangement of cores in mapping process.

Vertices	Cores							Communication Rate
	t_{11}	t_{12}	t_{01}	t_{10}	t_{02}	t_{20}	t_{22}	
V_1	✓							900
V_4		✓						550
V_3								500
V_0								450
V_2					✓			200
Neighbouring Free cores	4	3	3	3	2	1	1	

Fig. 8. Arrangement of vertices and cores in mapping process.

Vertices	Cores							Communication Rate
	t_{11}	t_{12}	t_{01}	t_{10}	t_{02}	t_{20}	t_{22}	
V_1	✓							900
V_4		✓						550
V_3			✓					500
V_0				✓				450
V_2					✓			200
Neighbouring Free cores	4	3	3	3	2	1	1	

Fig. 9. Mapping process case I.

Vertices	Cores							Communication Rate
	t_{11}	t_{12}	t_{01}	t_{10}	t_{02}	t_{20}	t_{22}	
V_1	✓							900
V_4		✓						550
V_3				✓				500
V_0			✓					450
V_2					✓			200
Neighbouring Free cores	4	3	3	3	2	1	1	

Fig. 10. Mapping process case II.

   Case II:
   $V_1 \in t_{11}$
   $V_4 \in t_{12}$
   $V_3 \in t_{10}$
   $V_0 \in t_{01}$
   $V_2 \in t_{02}$ is as shown in Fig. 10.
7. Map the vertices on a dynamic region on the basis of minimum communication energy as shown in Fig. 11. Finally map the vertices at

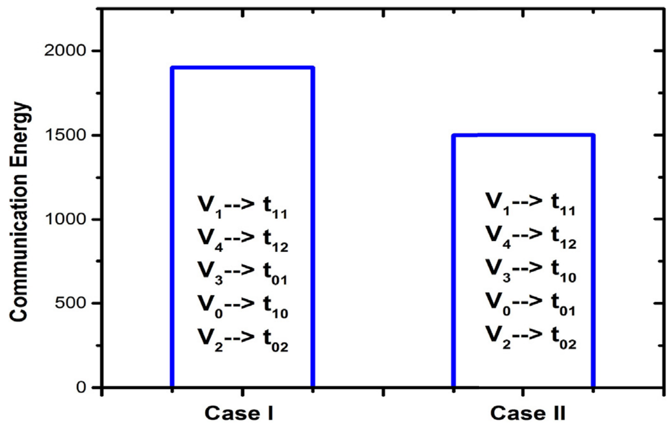

Fig. 11. Communication Energy of the two cases.

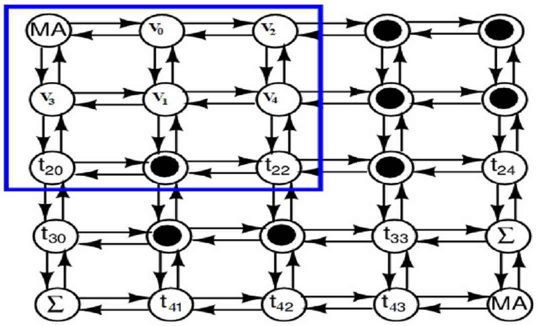

Fig. 12. Mapping Vertices in a dynamic region.

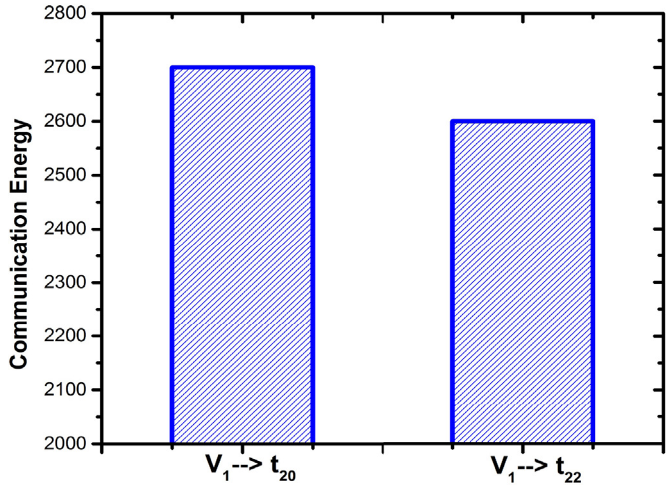

Fig. 13. Selecting the free core according to the communication energy.

the following locations on the NoC.
$V_1 \in t_{11}$
$V_4 \in t_{12}$
$V_3 \in t_{10}$
$V_0 \in t_{01}$
$V_2 \in t_{02}$ is as shown in Fig. 12.

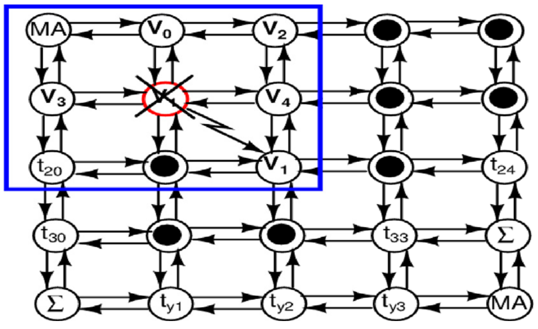

Fig. 14. Faulty Core Mapping in a dynamic region.

6. Faulty core mapping

After completion of the mapping, if faults occur at the processing core, faulty core tasks are migrated to the free core in the dynamic region on the NoC platform, which should be near the faulty core in the dynamic region. In our example t_{20} and t_{22} are the free cores in a dynamic region.

Free core $\in \{t_{20}$ & $t_{22}\}$ $\Longrightarrow$ Faulty core $\in \{t_{20}$ & $t_{22}\}$

Faults occur based on distribution model. Proposed mapping is used to model the failure rate of processing core given in (7) (Ellerman, 2012).

$$F_{xy}(t) = 1 - e^{-(\lambda_{xy} t)}. \tag{7}$$

where F_{xy}(t) is the failure probability of processing core, which is located in the *xth* row and *yth* column, where 't' is the life time of NoC.

λ_{xy}(t) is failure rate. It is a measure of failure per unit of time. The failure rate characteristically falls with life time and become constant until wear-out which shows an increase in failure rate (Ellerman, 2012).

Failure rate is calculated by dividing the total number of failures by the cumulative time of operation. A simple failure rate is indirectly proportional to total core hours and acceleration factor.

$$\lambda_{xy} \propto \left(\frac{1}{TCH \times AF}\right) \tag{8}$$

Here λ is failure rate. TCH is total core hours, it is multiplying the number of units and total time. AF is the acceleration factor, which is the test time multiplier derived from the Arrhenius equation. This equation calculates the time acceleration value obtained when a device is operated at an elevated temperature.

$$AF = e^{(E/KT_{xy})}. \tag{9}$$

E = Activation Energy (eV) of the failure mode.
K(Boltzmann Constant) = 8.617×10^{-5} eV/K.
T_{xy} = Temperature of the core, which is located at the xth row and yth column.

$$\lambda_{xy} = \frac{1}{TCH} \times e^{\left(-\frac{E}{KT_{xy}}\right)} \tag{10}$$

where TCH is a constant, then 1/TCH is also a constant and is denoted as '*A*'

$$\lambda_{xy} = A \times e^{\left(-\frac{E}{KT_{xy}}\right)} \tag{11}$$

A is a constant, which indicates the failure rate per cycle for each processing core operating at useful life of 10^{-9} for a typical core temperature.

6.1. Faulty core mapping in a dynamic region

Algorithm 3 FaultyCoreMapping

```
Input: Mapped Dynamic Region;
Output: Best Faulty Core Mapped Dynamic Region i.e.,
        BestFaultyMapping;

Let FC be the set of free cores available in a mapped dynamic region;
BestFaultyMapping= {};
Identify the FaultyCore;
foreach $fc \in \{ FC \}$ do
    fc ← FaultyCore ;
    FaultyMapping ← Dynamic region where $FaultyCore$ mapped to fc ;
    TotalCommunicationEnergy ← Compute total communication energy
     for FaultyMapping using (5);
    min ← TotalCommunicationEnergy ;
    if $min > TotalCommunicationEnergy$ then
        min←TotalCommunicationEnergy ;
        BestFaultyMapping ← FaultyMapping;
    end
end
Return BestFaultyMapping;
```

Faulty core mapping algorithm clearly explained in Algorithm 3. Once mapping of vertices is done in dynamic region and if there is any faulty core present, then the task of faulty core is migrated to spare core. After mapping is completed every free core act as a spare core, but it is important to highlight that the assigning of the spare core can be done only in case of faulty core situation. If N faulty cores are present in an NoC, we need N spare cores. Finally, spare core position is updated with the spare core that has the minimum energy among all the available cores in dynamic region. The iteration is repeated for all the available faulty cores. In this example (Fig. 3.) according to Eq. (7), the processing core at the t_{11} position has a high probability of failure. Therefore the t_{11} tasks are migrated to the free cores. Here, two free cores are at the same distance from the failed (t_{11}) position. Finally, a core is selected according to the minimum communication energy, as shown in Fig. 13, and the tasks are migrated from the failed core position (t_{11}) to the free core (t_{22}) shown in Fig. 14.

7. Evaluation

In this section we evaluate the communication energy, performance and area of our fault aware core mapping algorithm (EMAP) and compare the existing mapping algorithms. We applied different core mapping algorithms on above example Fig. 3. Section 5 clearly explained EMAP algorithm with example shown in Fig. 12. In order to compare algorithm domain and optimization is shown in Table 1.

To calculate the effectiveness of the EMAP algorithm, various ACGs with the number of vertices ranging from 4 to 20, which were generated using TGFF were examined (TGFF,). The EMAP algorithm was also evaluated by simulating ACGs on a 5 × 5 and 10 × 10 mesh NoC platforms.

The comparison of communication energy conservation and performance improvement of EMAP against of NMAP (Murali and De Micheli, 2004), BMAP (Shen et al., 2007) and PMAP (Koziris et al., 2000) is presented in Table 2. The average communication energy conservation and performance for simulated applications are presented in the last row of Table 2. When the core fails in any application, the performance and energy conservation automatically decreases, and the spare core replaces the failed core. When we compare the performance metrics of EMAP with those of the previous spare core placements, we observe the EMAP improved 17.78%, 11.9% and 14.25% on the communication energy conservation with 12.2%, 7.9% and 11.6% performance respectively on 5 × 5 NoC.

7.1. Test method

The simulation results are assessed by Noxim simulator (Catania et al., 2016), it is a cyclical accurate SystemC simulator for NoC sys-

Table 1
Characteristics of different algorithms and optimization algorithm.

	NMAP (Murali and De Micheli, 2004)	BMAP (Shen et al., 2007)	PMAP (Koziris et al., 2000)	EMAP
Algorithm	Intialization & Iteration	Greedy Binomial merge	NN- Embedded Algorithm	Efficient Mapping & Swapping
Optimization	Shortest Path	Low cost	Low complexity	Low Area & Communication Energy

Table 2
Evaluation of Communication Energy conservation and Performance Improvement of EMAP against of NMAP (Murali and De Micheli, 2004), BMAP (Shen et al., 2007) and PMAP (Koziris et al., 2000).

			NoC Size											
			EMAP against of NMAP				EMAP against of BMAP				EMAP against of PMAP			
			Communication energy conservation (%)		Performance improvement (%)		Communication energy conservation (%)		Performance improvement (%)		Communication energy conservation (%)		Performance improvement (%)	
Graph	Vertices	Edges	5 × 5	10 × 10	5 × 5	10 × 10	5 × 5	10 × 10	5 × 5	10 × 10	5 × 5	10 × 10	5 × 5	10 × 10
MPEG4 Decoder	12	13	17.82	19.8	12.8	15.4	12.1	15.4	8.4	9.2	14.67	17.8	12.2	12.8
MWD	12	13	16.4	16.2	10.2	12.82	10.4	11	6.6	8	12.3	13.2	9.8	11.4
VOPD	16	20	19.13	21.62	13.6	16.1	13.4	14.2	8.8	10.4	15.8	19.4	12.8	13.6
Average			17.78	19.2	12.2	14.77	11.9	13.5	7.9	9.2	14.25	16.8	11.6	12.6

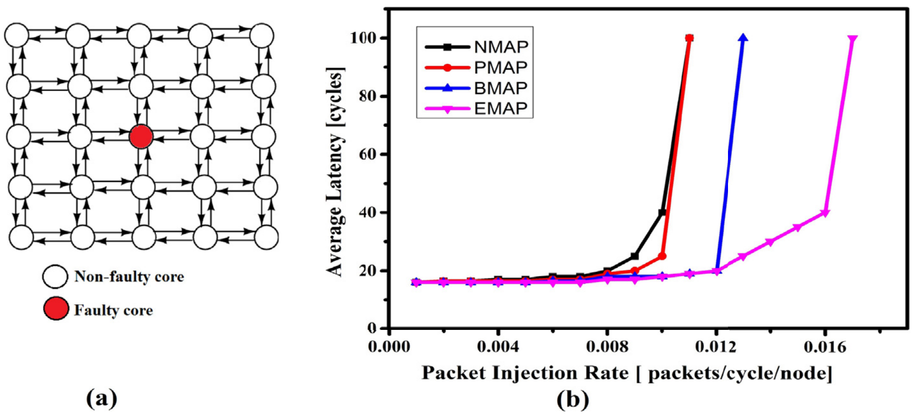

Fig. 15. (a) NoC platform with single faulty core. (b) Performance of different mapping algorithms under single faulty core.

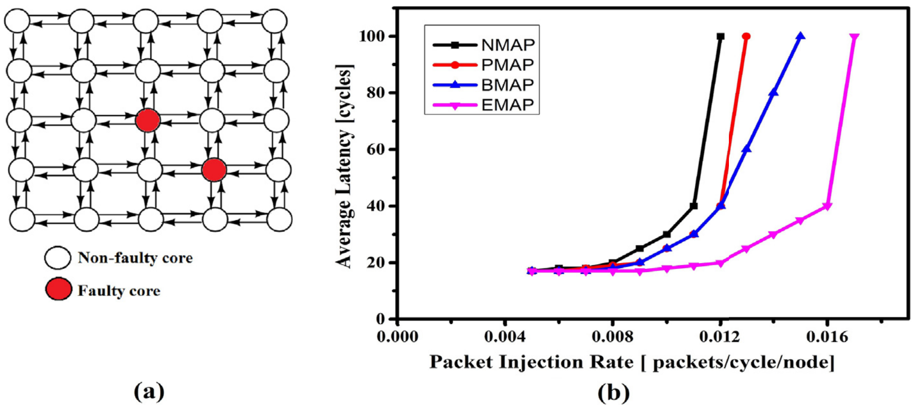

Fig. 16. (a) NoC platform with two faulty cores. (b) Performance of different mapping algorithms under two faulty cores.

tems. We consider the packet length is set to 8 flits, the simulation time is 20,000 cycles and the first 10,000 cycles is warm-up time. The average latency under various packet injection rate is utilized as the performance index of the simulations. Proposed algorithm and comparison algorithm are used XY routing, which gives the lower path diversity due to its deterministic properties. For the traffic pattern, we assess the network performance with transpose distribution in 5 by 5 mesh. The average latency under various packet injection rate is utilized as the performance index of the simulations.

7.2. System performance at single & multiple-faulty cores on NoC

In a single fault simulation setting, the faulty core is middle of NoC platform is as shown in Fig. 15 (a), and performance of different mapping algorithms under a single faulty core environment as shown in Fig. 15 (b), which can map the application on NoC platform. As it shows the performance of the NMAP algorithm is the worst among all, PMAP and BMAP better performance over NMAP algorithm. The average performance of EMAP algorithm is 25%, 42% and 65% compared with BMAP, PMAP and NMAP correspondingly.

Furthermore, we additionally asses the system performance with different number of faulty cores. In Fig. 16 (a) shows two faulty cores in NoC platform, the results shows that EMAP algorithm is outperformed BMAP, PMAP and NMAP with average performance of 30%, 40% and 60% compared with BMAP, PMAP and NMAP correspondingly, under environment of 2 cores shown in Fig. 16 (b). EMAP algorithm won't effect on performance under single fault and two faults, because faulty cores are outside of core mapping region. The circumstances is similar when 4 faulty cores located around the center area of mesh as shown in Fig. 17 (a). So the congestion is more severe than previous simulations.

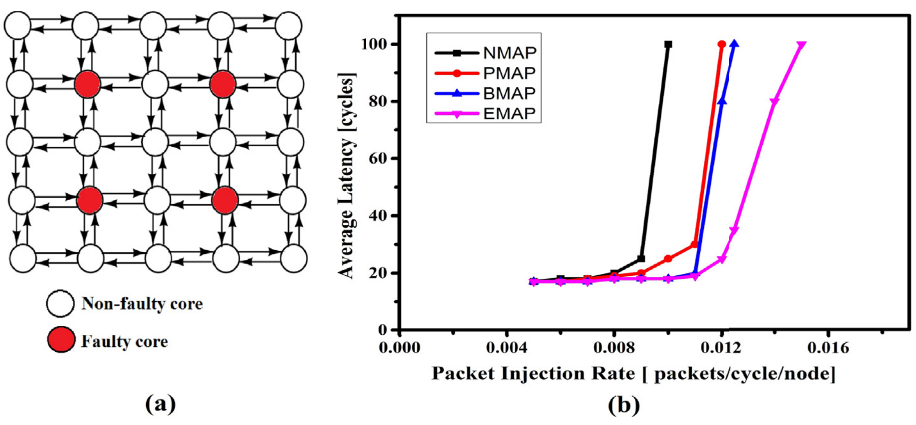

Fig. 17. (a) NoC platform with four faulty cores. (b) Performance of different mapping algorithms under four faulty cores.

Table 3
Evaluation of power, area and throughput of EMAP against NMAP, PMAP & BMAP.

	EMAP against NMAP	EMAP against PMAP	EMAP against BMAP
Area (Equivalent Logic Blocks)	14%	8%	6%
Power Consumption (W)	19.2%	14%	10%
Throughput (Gbps)	26%	19.6%	14%

The outcome demonstrates that EMAP algorithm has preferred execution over BMAP, PMAP and NMAP algorithm by 40%, 44% and 55% as shown in Fig. 17 (b). Finally, the performance of EMAP algorithm during single and multiple faults is great tolerance ability in serious faulty environment compared with related work.

7.3. Synthesis results

To accurately estimate the power, area and performance of the proposed EMAP, NMAP, PMAP & BMAP, Vivado 2016.4 is used (Vivado Design Suite,). The evaluated power, area, and performance of the EMAP compared with those of NMAP, PMAP & BMAP are tabulated in Table 3. Area is one of the most significant design metrics of an NoC realized in an FPGA. One of the most important advantages of the EMAP over NMAP, PMAP & BMAP is its area efficiency, we compute the equivalent number of logic blocks that represent its area. For power analysis, this EMAP uses the XY routing algorithm, and packets are distributed along minimum hop paths. The performance is evaluated as the throughput. The area and power consumption of EMAP algorithm is decreased by an average of 14% and 19.2%, 8% and 14%, and 6% and 10% when compared to NMAP, PMAP, and BMAP. The performance improved by an average of 26%, 19.6%, and 14% when EMAP algorithm is compared to NMAP, PMAP, and BMAP. It can be seen that EMAP algorithm is efficient core mapping compared with related algorithms.

8. Conclusion

In this paper presented an energy efficient fault aware core mapping algorithm on NoC platform. To reduce the mapping area using NAD & PVR, the communication energy was efficiently calculated by using WCE. If any faults occur at any core during the mapping process, the faulty core is effectively identified using failure probability. Finally, proposed mapping algorithm has better performance substantially under faulty conditions, when compared to related mapping algorithms. Further, to assess the efficiency of the proposed EMAP algorithm, the results showed lower area, reduced power consumption, and better performance compared with related algorithms.

Acknowledgement

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

References

Agrawal, Saurabh, Sant, Dhawal, Sharma, G.K., 2010. An efficient energy- and bandwidth- aware mapping algorithm for regular NoC architectures. In: International Symposium on Networks-on-chip.

Beechu, N.K.R., et al., 2017. System level fault-tolerance core mapping and FPGA-based verification of NoC. Microelectron. J. 70, 16–26.

Beechu, N.K.R., et al., 2017. High-performance and energy-efficient fault-tolerance core mapping in NoC. Sustain. Comput. Inform. Syst. 16, 1–10.

Catania, Vincenzo, Mineo, Andrea, Monteleone, Salvatore, 2016. Cycle-accurate network on chip simulation with Noxim. ACM Trans. Model Comput. Simulat. 27 (1).

Chou, Chen-Ling, Marculescu, Radu, 2011. FARM: fault-Aware resource management in NoC based multiprocessor platforms. In: Design Automation& Test in Europe Conference & Exhibition (DATE).

Coskun, Ayse Kivilcim, Rosing, Tajana Simunic, Mihic, Kresimir, De Micheli, Giovanni, Leblebici, Yusuf, 2006. Analysis and optimization of MPSoC reliability. J. Low Power Electron. 2, 56–69.

Ellerman, Paul, 2012. Calculating reliability using FIT and MTTF: Arrhenius HTOL model. MicroNote.

Hu, Wei, Du, Chen, Yan, Like, Tianzhou, Chen, 2009. A fast algorithm for energy-aware mapping of cores onto WK-recursive NoC under performance constraints. In: International Conference on High Performance Computing (HiPC).

Khalili, Fatemeh, Zarandi, Hamid R., 2013. A fault-tolerant core mapping technique in networks-on-chip. IET Comput. Digital Tech. 7 (6), 238–245.

Khalilia, Fatemeh, Zarandi, Hamid R., 2012. A Fault-tolerant low-energy multi-application mapping onto NoC-based multiprocessors. In: IEEE 15th International Conference on Computational Science and Engineering, pp. 421–428.

Khalilia, Fatemeh, Zarandi, Hamid R., 2013. A reliability-aware multi-application mapping technique in networks-on-chip. In: 21st Euromicro International Conference on Parallel, Distributed, and Network-based Processing, pp. 478–485.

Koziris, Nectarios, Romesis, Michael, Tsanakas, Panayiotis, Papakonstantinou, George, 2000. An efficient algorithm for the physical mapping of clustered task graphs onto multiprocessor architectures. In: Proceedings. 8th Euromicro Workshop on Parallel and Distributed Processing.

Michael, Nithin, Wang, Yao, Edward Suh, G., Tang, Ao, 2013. Quadrisection-based task mapping on many-core processors for energy-efficient on-chip communication. In: International Symposium on Networks-on-chip.

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

Naresh Kumar Reddy, B., Vasantha, M.H., Nithin Kumar, Y.B., Sharma, Dheeraj, 2015. A fine grained position for modular core on NoC. In: IEEE International Conference on computer, Communication and Control (IC4), pp. 1–4.

Naresh Kumar Reddy, B., Vasantha, M.H., Nithin Kumar, Y.B., Sharma, Dheeraj, 2015. Communication energy constrained spare core on NoC. In: 6th International Conference on Computing, Communication and Networking Technologies (ICCCNT), pp. 1–4.

Naresh Kumar Reddy, B., Vasantha, M.H., Nithin Kumar, Y.B., 2016. A gracefully degrading and Energy-efficient fault tolerant NoC using spare core. In: IEEE Computer Society Annual Symposium on VLSI (ISVLSI 2016), Pennsylvania, U.S.A, pp. 146–151.

Parhami, Behrooz, 2013. Exact formulas for the average internode distance in mesh and binary tree networks. Computer Science and Information Technology 165–168.

Rahmati, Dara, Murali, Srinivasan, Benini, Luca, De Micheli, Giovanni, Sarbazi-Azad, Hamid, March 2013. Computing accurate performance bounds for best effort networks-on-chip. IEEE Trans. Comput. 62 (3).

Shen, Wein-Tsung, Chao, Chih-Hao, Lien, Yu-Kuang, Wu, An-Yeu, 2007. A new binomial mapping and optimization algorithm for reduce complexity mesh based on chip-networks. In: Proceedings of the First International Symposium on Networks-on-chip.

Srinivasan, Krishnan, Chatha, Karam S., 2005. A technique for low energy mapping and routing in network-on-chip architectures. In: International Symposium on Low Power Electronics and Design, pp. 387–392.

Task graphs for free (TGFF) Available: http://ziyang.eecs.umich.edu/&percnt;CB&percnt;9Cdickrp/tgff/.

Vivado Design Suite - HLx Editions: https://www.xilinx.com/products/design-tools/vivado.html.

Yang, Bo, December 2013. Towards Optimal Application Mapping for Energy-efficient Many-Core Platforms, vol. 167. Turku Centre for Computer Science.

B. Naresh Kumar Reddy received the bachelor of engineering degree in electronics and communication in 2010 from the S.V.University, India, and master's of engineering degree in embedded system in 2012 from the K.L.University, India. He is currently pursuing Ph.D. in NIT Goa, and working in Intel, India. His research interests focus on fault-tolerant and networks-on-chip and embedded systems, on-chip multiprocessors, reconfigurable systems. He is coauthor of several scientific papers on networks-on-chip, design methodologies for systems-on-chip, embedded system.

Vasantha M.H received the masters of engineering degree in electronics and communication in 2003 from the IITM, India, and Ph.D in 2014 from the NIT Surathkal, India. He is currently Assistant professor and HOD of Electronics department in NIT Goa, India. His research interests focus on Low voltage, Low power analog mixed signal circuits, Continuous-time filter Circuits, System on Chip, fault-tolerant and networks-on-chip and embedded systems.

Y.B.Nithin Kumar received the masters of engineering degree in electronics and communication in 2003 from the NIT Surathkal, India, and Ph.D in 2014 from the IIT Kharagpur, India. He is currently Assistant professor in Electronics department in NIT Goa, India. His research interests focus on Analog and Mixed Signal Design, System on Chip, fault-tolerant and networks-on-chip and embedded systems.