An Efficient Real-Time Embedded Application Mapping For NoC Based Multiprocessor System on Chip

ARURU SAI KUMAR (asaikumar.nitw@gmail.com)
NIT Warangal: National Institute of Technology Warangal https://orcid.org/0000-0001-7924-8575
T.V.K Hanumantha Rao
NIT Warangal: National Institute of Technology Warangal
Naresh kumar reddy Beechu
IIITM-K: Indian Institute of Technology and Management-Kerala

Research Article

Keywords: System on Chip (SoC), Network on Chip (NoC), Core mapping, Real-Time embedded applications, Performance

Posted Date: June 2nd, 2022

DOI: https://doi.org/10.21203/rs.3.rs-880912/v1

Noname manuscript No.
(will be inserted by the editor)

An Efficient Real-Time Embedded Application Mapping For NoC Based Multiprocessor System on Chip

Aruru Sai Kumar[1],T.V.K. Hanumantha Rao[2] and B. Naresh Kumar Reddy[3]

Received: date / Accepted: date

Abstract The evolution of the components embedded on a single chip is growing faster day by day, resulting in a considerable impact on the performance metrics and communication between the cores in the NoC architecture. So, to overcome such issues, it is important to provide an efficient mapping between the cores, such that the communication between them increases. To improve the performance of a network, throughput and latency also play a significant role. In this research paper, an efficient mapping strategy implemented on the real-time embedded applications named ERTEAM. In this algorithm, based on the minimum Core Average Distance (CAD) the mapping region is finalized, ensuring the overall mapping area reduced. The PE's mapped according to the minimum communication energy in the selected mapping region. This research evaluated a set of embedded applications, which reveals a reduction in latency at 12.3% against BBPCR and 8.4% against SBMAP. The simulation time reduces at an average of 19% against BBPCR and 9.6% against SBMAP. The throughput increases at an average of 14.5% against BBPCR and 7.8% against SBMAP and reduces the communication energy by 15.6% against BBPCR and 5.2% against SBMAP.

Keywords System on Chip (SoC) · Network on Chip (NoC) · Core mapping, · Real-Time embedded applications, · Performance.

[1,2]Department of Electronics and Communication Engineering, National Institute of Technology Warangal, Telangana, India
Email: asaikumar.nitw@gmail.com[1],tvkhrao75@gmail.com[2]
[3] School of Electronic Systems and Automation, Kerala University of Digital Sciences, Innovation and Technology, Indian Institute of Information Technology and Management Kerala (IIITMK), Thiruvananthapuram 695317, India.
Email: naresh.nitg@gmail.com[3]

1 Introduction

A new ecosystem has originated for semiconductor devices, which allows complex tasks and features to be integrated into a single package, referred to as SoC. As per International Technology Roadmap for Semiconductors, named ITRS 2.0, an emerging ecosystem comprises heterogeneous implementation with electronic components linked to various application domains, including High-Performance Computing (HPC), IoT, Big Data, including Cloud Computing [1]. The architecture utilized in SoC design is bus-based structures that could not evolve well as an application's communication needs an expansion. The Network-on-Chip (NoC) connectivity approach is developed as a solution to resolve these limitations [2].

Network Interface (NI), Routers or Switches, including connectivity links, are the core elements of NoC represented in Fig. 1. The cores in NoC communicate with each other through interconnection links using a technique named packet-based switching. On the intended NoC platform, the detailed design of a NoC architecture comprises Task Partitioning in an application, Tasks Management and Scheduling, including Application Mapping mechanisms. Required tasks schedule and the processing time is strongly associated with task partitioning, along with task allocation. Later, the tasks aligned to the application cores to perform the execution. Therefore, the mapping of applications on NoC platform is the most critical and fundamental problems in NoC designs [3],[4].

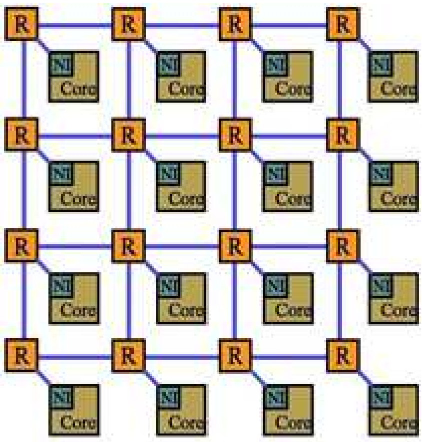

Fig. 1 3 X 3 Mesh based NoC.

The primary responsibility of any mapping technique is to map the tasks to the cores available in the chosen platform. Then, the mapping of an application allows to perform the tasks as mapped accordingly and provide the suitable output. As the number of cores is increasing drastically, many mapping techniques came into existence to provide a reliable result. So, it is essential to follow certain rules by considering the critical shortcomings in the present NoC methodologies. Therefore, an efficient mapping technique is implemented by following the above rules, entitled ERTEAM (Efficient Real-Time Embedded Application Mapping). This mapping strategy mainly implemented for the real-time embedded applications and deals with providing a mapping region through minimum Core Average Distance (CAD) and mapping the PE's in the reduced mapping area.

The organisation of this research is determined as follows; Section II provides the related work and, Section III provides the Model Analysis of NoC architecture. Section IV explains the details of the proposed mapping strategy and the experimented outcomes represented in Section V. The research paper concludes with Section VI.

2 Related Work

As Core mapping plays a crucial role in NoC architectures, many research works were proposed to provide efficient mapping strategies to improve performance metrics. Below listed were some of the recent mapping methodologies and their outcomes.

Bing Li et al.,[5] implemented a runtime mapping that is a thermal-aware algorithm that optimizes the overall performance for 3D NoC. The available core regions restored through the defragmentation algorithm introduced in this mapping. LI Guangshun et al.,[6] implemented a mechanism for mapping the irregular IPs embedded on a regular 2D mesh topology for NoC architectures. The core principle is to break down each big IP into several smaller dummy IPs, each of which can move into a single tile, reducing energy consumption and avoiding congestion.

Weichen Liu et al.,[7] proposed a TopoMap algorithm for the SMART NoC architectures to improve performance. The topology of the architecture is selected dynamically based on the configuration by the thermal aware task mapping algorithm. Guoyue Jiang et al.,[8] developed a mapping strategy based on the BB algorithm to provide both the core and the communication mapping. This scheme reduces the overall latency and the energy of the hybrid NoC and optimizes the overall mapping. P.V Bhanu et al.,[9] proposed a technique to provide a fault-tolerant system and verified it through both simulations and FPGA validation. Firstly, the mapping of an application by considering the fault-tolerant mechanism for a Torus topology performed through ILP and PSO. Therefore, it provides a complete mathematical approach for replacing a faulty core with a spare core.

Sarzamin Khan et al.,[10] implemented a BEMAP algorithm where the real-time applications mapped, considering bandwidth constraints. This mapping mechanism used the modular systematic searching technique, where the system is divided into small possible modules and performed the mapping on both Torus and Mesh topologies. Therefore, it reduces the overall latency and energy consumption for 2D NoC architectures. Leibo Liu et al.,[11] proposed a BBPCR algorithm to find the optimal mapping for an application. Firstly, a PCM model that is highly accurate and flexible developed, containing both the energy and reliability parameters. Later, using this model, BBPCR is implemented for figuring out the best mapping solution for an application. Therefore, it significantly impacts the improvement of reliability, low energy consumption, and low latency. The SBMAP [12] mechanism implemented, considering bandwidth constraints to minimize energy consumption and computational complexity. This mapping mechanism used the modular systematic searching technique. The system is divided into small possible modules and performed the mapping on it, resulting in high performance and less simulation time.

3 Model Analysis

3.1 Background

A Network Core Graph (NCG), G = G(P, A) is a directed graph in which vertices of the graph represent the available processing elements PE's (P = P_1, P_2, P_3, . . . , P_n) for the task execution. The directed arc ($a_{ij} \in$ A) shows characteristic parameters and required bandwidth between the IP cores (P_i to P_j).

NoC Architecture Graph (NAG), A = A(C, D) is a topology graph in which the node of the graph, (C = C_1, C_2, C_3, . . . , C_n) shows a network cores and the directed arc, 'D' represents the communication distance $\forall$ $C_{ij} \in$ D, and C_{ij} denotes the distance between core (C_i) and core (C_j).

NoC Architecture Mapping Graph (NMG), M = M(C, D) is a topology graph in which node of the mapping graph, (C = C_1, C_2, C_3, . . . , C_n) shows a network mapping cores and the directed arc, 'D' represents the communication distance $\forall$ $C_{ij} \in$ D and C_{ij} denotes the distance between core (C_i) and core (C_j).

3.2 Core Average Distance (CAD)

CAD is the shortest average path length between any two cores in the network. The average distance between any two selected vertices of a network, which is of X*Y size in NoC is evaluated as shown in below Eq.(1), such that the evaluation of CAD provides the mapping region for a NoC network [13],[14].

$$CAD = \frac{X+Y}{3} \cdot (1 - \frac{1}{XY}) \tag{1}$$

3.3 Measurement of Communication Energy

Communication energy is considered the same as the distance between two tiles or nodes [15-17]. It is calculated as the sum of differences between their corresponding modules determines the distance among two vertices, i.e. V_i and V_j, where V_i having parameters as (a_0, b_0) and V_j having parameters as (a_1,b_1).

Therefore, the Total Communication Energy (TCE) calculated as mentioned in Eq.(2).

$$TCE = \sum_{\forall t_i \in \{T\}} W(E_{ij}) \times \{|(a_1 - a_0)| + |(b_1 - b_0)|\} \tag{2}$$

where $W(E_{ij})$ is illustrated as the weighted communication energy between any two nodes in a network.

3.4 Measurement of Performance

The throughput and latency considered important metrics for performance improvement [18], [19]. Since network congestion significantly impacts latency, avoiding congestion for each node is an efficient way to minimize latency. Simultaneously, less congestion will result in increased throughput. As a result, the bandwidth limitation, which is interrelated to congestion, is considered the performance limitation. Therefore, the communication volumes for each node managed through bandwidth restrictions, so congestion is diminished, and performance, including latency and throughput, is guaranteed [20].

4 Proposed ERTEAM Technique

4.1 Problem Definition

An efficient real time embedded application mapping problem is defined as:
Given a set of Network Core Graph (NCG), G = G(P, A) and NoC Architecture Graph (NAG), A = A(C, D), finding a mapping function M(C, D) that maps an IP core $c_i \in$ C in the NCG to a PE in the NoC.
$\forall P_i \in P$,
$\forall C_i \in C$,
$\Omega(P_i) \in C$,
$P_i \neq P_j \Rightarrow \Omega(P_i) \neq \Omega(P_j)$
$\forall \mathrm{C}_{ij} \in D$
Let $a_{ij} \in$ A be mapped to some $P_{xy} \in$ C then $P_{xy} = \Omega(\mathrm{C}_i) \in$ D.
CE(P_i, P_j) = W(e_{ij}) $\times$ | P(i)P(j) | in terms of nodes.
CE(P_i, P_j) = W(e_{ij}) $\times$ C_{ij}

C_{ij} denotes the distance between core (C_i) and core (C_j). C_i parameters (a_1, b_1), C_j parameters (a_2, b_2). W(e_{ij}) denotes the communication rate from P_i to P_j.

Total communication energy (TCE) is illustrated in Eq. (3).

$$TCE = \sum_{\forall(i,j)} W(e_{ij}) \times C_{ij} \tag{3}$$

Algorithm 1 ERTEAM Algorithm

```
Input: Network Core Graph (NCG) G = (P,A) ;
       NoC Architecture Graph (NAG) A = (C,D) ;
Output: NoC Architecture Mapping Graph (NMG) = M(C,D) ;
       M: Mapping Region ;

foreach mapping region do
    Calculate Effective Region ;
    {
    Select Effective Region corresponding to the min Core Average Distance
     (CAD) ;
    }
end
Initialize Mapping ;
min cost = ∞ ;
do
    Calculate Core Bandwidth (BW) ;
    Calculate Communication Distance (CD) ;
    Calculate Communication Energy (CE) ;
    {
    CE = BW ×CD
    }
    if min CE = Total CE ;
     then Total Communication Energy < min Communication Energy
    end
    Core Mapping = min CE mapping ;
    Total CE (TCE) = ∑ BW_(Pi,Pj) × CD_(Ci,Cj)
while Next Mapping;
return Best Communicative Mapping with lowest Communication Energy ;
Calculate Core Mapping Execution Time ;
Calculate Latency and Throughput ;
```

4.2 Proposed Mapping

Proposed core mapping algorithm explained in Algorithm 1. Network core graph (NCG) and NoC architecture graphs (NAG) are acts as input, NoC Architecture Mapping Graph (NMG) as output. Initially, select the efficient mapping region using minimum core average distance (CAD), reducing the mapping area. Then, Processing Element (PE's) in NCG mapped on efficient

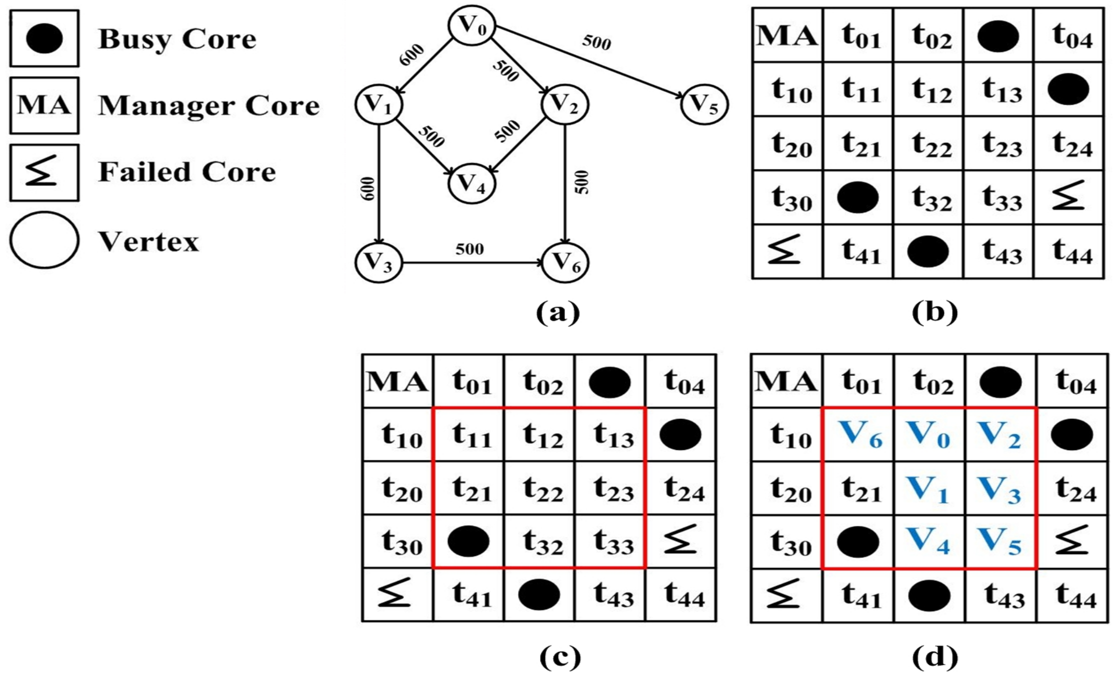

Fig. 2 ERTEAM Algorithm: (a) An example of Application Task Graph, (b) 5 x 5 mesh NoC, (c) Mapping region obtained through minimum CAD, and (g) Efficient ERTEAM Core Mapping.

mapping region in NoC according to the minimum communication energy. A simple example clearly explained in Fig. 2. A simple network core graph has shown in Fig. 2(a) and 5x5 NoC Architecture Graph shown in Fig. 2(b). As the number of vertices is 7 in the NCG, the efficient mapping region is selected based on CAD, preferably a size 3x3 region shown in Fig. 2(c). Finally, NCG vertices mapped on 3x3 region according to the minimum communication energy of the network shown in Fig. 2(d).

5 Experimental Results

In this section, we conduct sets of comprehensive experiments to evaluate the effectiveness of the ERTEAM algorithm, mapping performance and communication energy. The mentioned metrics are compared with state-of-the-art approaches on embedded applications. A set of embedded applications exploited for evaluation. Application names and their numbers of cores are shown in Table 1 [21]. The best mapping pattern found using a C++ program, the simulations carried out on Noxim simulator [22], and the time consumed can be obtained. .

Table 1 Specifications of Embedded Applications.

Application	No. Cores	Network Size
H264 encoder (H264_enc)	36	6 x 6
MP3 decoder (MP3_dec)	16	4 x 4
Network processing (NP)	16	4 x 4
MPEG2 encoder (MPEG2)	16	4 x 4
Multimedia Systems (MMS)	16	4 x 4
Video object plan decoder (VOPD)	16	4 x 4

For all of the following simulations, Network Core Graph and NoC Architecture Graph are identical. This research methodology evaluates performance metrics such as Latency, Simulation Time, Throughput, and Communication Energy.

5.1 Latency

The time taken by the packet's header flit to migrate between any source to destination in the network referred to as latency. According to network congestion, latency frequently involves a packet's waiting time between any source to the destination node, illustrated in Eq.(4).

$$Latency = \frac{1}{K}\sum_{n=1}^{K}(L_n) \quad (4)$$

K = Total number of packets reaching their destination cores. L_n = The clock cycle latency for the n^{th} node.

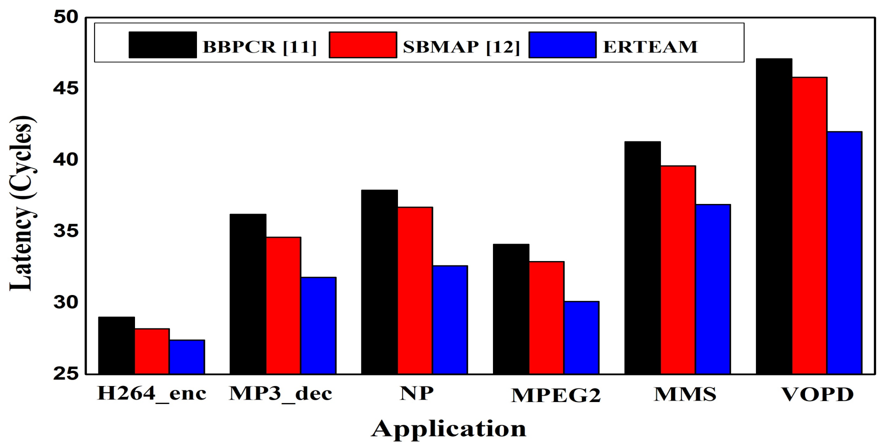

Fig. 3 Latency of the proposed algorithm ERTEAM (in terms of cycles) compared to the BBPCR[11] and SBMAP[12].

Table 2 explains the obtained latency of the proposed algorithm ERTEAM (in terms of cycles) compared to the BBPCR[11] and SBMAP[12]. Therefore, the graphical representation of the latency depicted in Fig. (3).

Table 2 Latency of the proposed algorithm for various embedded applications.

Latency (Cycles)			
Application	BBPCR[11]	SBMAP[12]	ERTEAM
H264 encoder (H264_enc)	29	28.2	27.4
MP3 decoder (MP3_dec)	36.2	34.6	31.8
Network processing (NP)	37.9	36.7	32.6
MPEG2 encoder (MPEG2)	34.1	32.9	30.1
Multimedia Systems (MMS)	41.3	39.6	36.9
Video object plan decoder (VOPD)	47.1	45.8	42

5.2 Simulation Time

The term simulation time is defined as the overall time required by the system to execute the tasks during the mapping of cores, known as the simulation time or the execution time. Thus, lesser simulation time provides an increase in the performance of the system. Table 3 illustrates the obtained simulation time of the proposed algorithm ERTEAM (in terms of seconds) compared to the BBPCR[11] and SBMAP[12]. Therefore, the graphical representation of the simulation time depicted in Fig. (4).

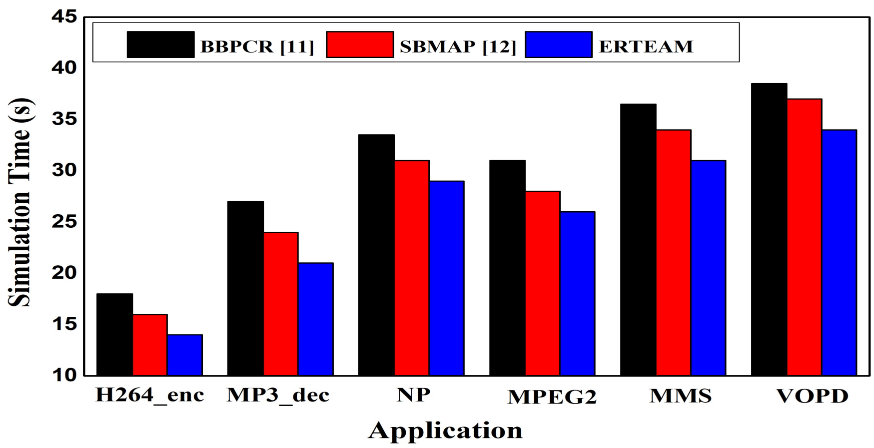

Fig. 4 Simulation Time of the proposed algorithm ERTEAM (in terms of seconds) compared to the BBPCR[11] and SBMAP[12].

Table 3 Simulation time of the proposed algorithm for various embedded applications.

Simulation Time (s)			
Application	BBPCR[11]	SBMAP[12]	ERTEAM
H264 encoder (H264_enc)	18	16	14
MP3 decoder (MP3_dec)	27	24	21
Network processing (NP)	33.5	31	29
MPEG2 encoder (MPEG2)	31	28	26
Multimedia Systems (MMS)	36.5	34	31
Video object plan decoder (VOPD)	38.5	37	34

5.3 Throughput

Throughput considered as one of the important parameters regarding the performance of the system. It represents the maximum amount of information that transferred in a given amount of time. Therefore, the mathematical formulation for throughput illustrated in Eq. (5).

$$Throughput = \frac{R_p}{N \times N_p} \tag{5}$$

Where, R_p = total number of received packets, N = the total number of cores, N_p = number of clocks cycles lapsed from the first generated packet to the last received packet.

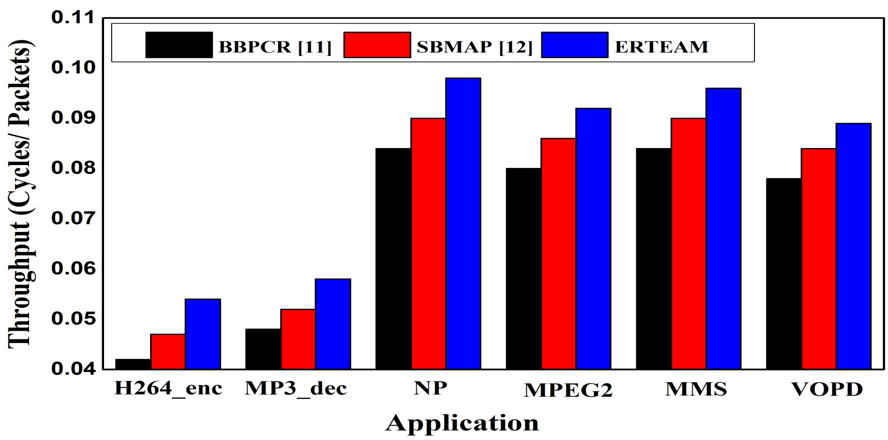

Fig. 5 Throughput of the proposed algorithm ERTEAM (in terms of cycles/packets) compared to the BBPCR[11] and SBMAP[12].

Table 4 describes the resultant throughput of the proposed algorithm ERTEAM (in terms of cycles/packets) compared to the BBPCR[11] and SBMAP[12], whereas the graphical representation of throughput depicted in Fig. (5).

Table 4 Throughput of the proposed algorithm for various embedded applications.

Throughput (Cycles/ Packets)			
Application	BBPCR[11]	SBMAP[12]	ERTEAM
H264 encoder (H264_enc)	0.042	0.047	0.054
MP3 decoder (MP3_dec)	0.048	0.052	0.058
Network processing (NP)	0.084	0.09	0.098
MPEG2 encoder (MPEG2)	0.08	0.086	0.092
Multimedia Systems (MMS)	0.084	0.09	0.096
Video object plan decoder (VOPD)	0.078	0.084	0.089

5.4 Communication Energy

The term Communication Energy defined as the sum of differences between their respective modules establishes the distance between any two nodes in a chosen topology of a network. Table 5 illustrates the communication energy of the proposed algorithm ERTEAM (in terms of μJ) compared to the BBPCR[11] and SBMAP[12]. Therefore, the graphical representation of the communication energy depicted in Fig. (6).

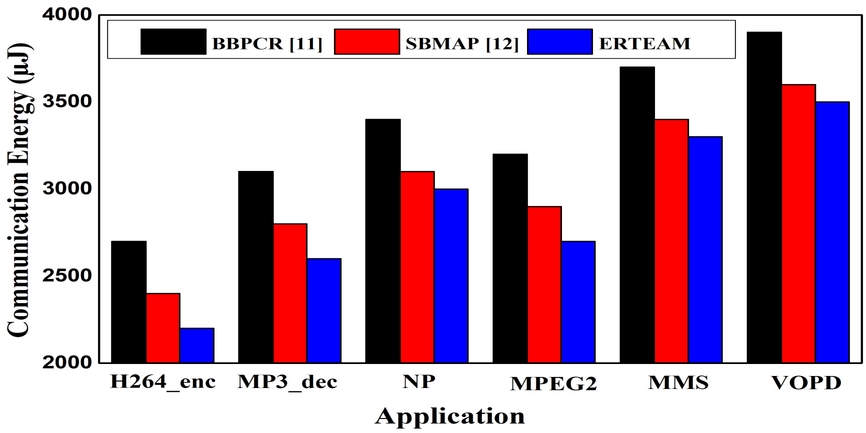

Fig. 6 Communication Energy of the proposed algorithm ERTEAM (in terms of μJ) compared to the BBPCR[11] and SBMAP[12].

Table 6 demonstrates the evaluation of the metrics for the proposed ERTEAM algorithm against the BBPCR[11] and SBMAP[12]. The reduction of latency improved by an average of 12.3% and 8.4% against BBPCR[11] and SBMAP[12], the overall simulation time reduced to 19%, 9.6% compared to BBPCR[11] and SBMAP[12]. Furthermore, the throughput of ERTEAM improved by an average of 14.5%, 7.8% compared to BBPCR[11] and SBMAP[12] and the communication energy reduced to 15.6%, 5.2% against BBPCR[11] and SBMAP[12].

Table 5 Communication Energy of the proposed algorithm for various embedded applications.

Communication Energy (μJ)			
Application	BBPCR[11]	SBMAP[12]	ERTEAM
H264 encoder (H264_enc)	2700	2400	2200
MP3 decoder (MP3_dec)	3100	2800	2600
Network processing (NP)	3400	3100	3000
MPEG2 encoder (MPEG2)	3200	2900	2700
Multimedia Systems (MMS)	3700	3400	3300
Video object plan decoder (VOPD)	3900	3600	3500

Table 6 Evaluation of latency, simulation time, throughput and communication energy of ERTEAM against BBPCR[11] and SBMAP[12].

	ERTEAM against BBPCR[11]	ERTEAM against SBMAP[12]
Latency (Cycles)	12.3%	8.4%
Simulation Time (s)	19%	9.6%
Throughput (Cycles/ Packets)	14.5%	7.8%
Communication Energy (μJ)	15.6%	5.2%

6 Conclusion

The proposed mapping strategy entitled ERTEAM is applied to real-time embedded applications to improve the network's performance. This implementation chooses the mapping region based on the minimum Core Average Distance. After providing the mapping area, the PEs embedded in the arrangement of minimum communication energy between the cores. The resultant outcome of the proposed mapping technique provides low latency at an average of 12.3%, 8.4% against BBPCR and SBMAP, less simulation time of 19% against BBPCR and 9.6% against SBMAP. In addition, the overall throughput increased at an average of 14.5%, 7.8% compared to BBPCR and SBMAP. The communication energy of ERTEAM reduced by 15.6% and 5.2% against BBPCR and SBMAP respectively.

Acknowledgment: The authors would like thank the anonymous reviewers.

Data Availability Statement: The referred papers will be available on request.

Funding: Not Applicable

Conflict of Interest: Authors declare no conflict of Interest.

Ethics Approval: The authors declare that they have no known competing financial interest or personal relationships that could have appeared to influence the work reported in this paper.

Consent to participate: All authors voluntarily agree to participate in this review paper.

Consent for Publication: All authors give the permission to the Journal to publish this review paper

References

1. ITRS, International technology roadmap for semiconductor 2.0, Executive report (2015).
2. Luca Benini and Micheli G.D.,(2002) "Networks on Chips: A New SoC Paradigm," IEEE computer, vol. 35, no. 1, pp. 70-78. DOI: 10.1109/2.976921
3. A. Sai Kumar and T. V. K. Hanumantha Rao, (2019)"Efficient Core Mapping on Customization of NoC Platforms", 2019 IEEE International Symposium on Smart Electronic Systems (iSES) (Formerly iNiS), Rourkela, India, pp. 57-62.
4. Aruru Sai Kumar, T.V.K. Hanumantha Rao,(2020) "Scalable benchmark synthesis for performance evaluation of NoC core mapping," Microprocessors and Microsystems, Volume 79, 103272.
5. B. Li, X. Wang, A. K. Singh and T. Mak,(2019) "On Runtime Communication and Thermal-Aware Application Mapping and Defragmentation in 3D NoC Systems," in IEEE Transactions on Parallel and Distributed Systems, vol. 30, no. 12, pp. 2775-2789.
6. G. Li, J. Wu and G. Ma,(2007) "Mapping of irregular IP onto NoC architecture with optimal energy consumption," in Tsinghua Science and Technology, vol. 12, no. S1, pp. 146-149.
7. W. Liu et al., (2018)"Thermal-Aware Task Mapping on Dynamically Reconfigurable Network-on-Chip Based Multiprocessor System-on-Chip," in IEEE Transactions on Computers, vol. 67, no. 12, pp. 1818-1834.
8. G. Jiang, Z. Li, F. Wang and S. Wei, (2015)"Mapping of Embedded Applications on Hybrid Networks-on-Chip with Multiple Switching Mechanisms," in IEEE Embedded Systems Letters, vol. 7, no. 2, pp. 59-62.
9. P. V. Bhanu, R. Govindan, P. Kattamuri, J. Soumya and L. R. Cenkeramaddi,(2021) "Flexible Spare Core Placement in Torus Topology Based NoCs and Its Validation on an FPGA," in IEEE Access, vol. 9, pp. 45935-45954.
10. S. Khan, S. Anjum, U. A. Gulzari, M. K. Afzal, T. Umer and F. Ishmanov, (2018)"An Efficient Algorithm for Mapping Real Time Embedded Applications on NoC Architecture," in IEEE Access, vol. 6, pp. 16324-16335.
11. L. Liu et al., (2015)"A Flexible Energy- and Reliability-Aware Application Mapping for NoC-Based Reconfigurable Architectures," IEEE Trans. on Very Large Scale Integration (VLSI) Systems, vol. 23, no. 11, pp. 2566-2580.
12. S. Khan, S. Anjum, U. A. Gulzari, T. Umer and B. Kim, (2018)"Bandwidth-Constrained Multi-Objective Segmented Brute-Force Algorithm for Efficient Mapping of Embedded Applications on NoC Architecture," in IEEE Access, vol. 6, pp. 11242-11254.
13. A. S. Kumar, T. V. K. Hanumantha Rao and B. N. Kumar Reddy, (2020)"Exact Formulas for Fault Aware Core Mapping on NoC Reliability", 2020 IEEE 17^{th} India Council International Conference (INDICON), New Delhi, India, pp. 1-5. DOI: 10.1109/INDICON49873.2020.9342427.
14. Beechu, N., Moodabettu Harishchandra, V. & Yernad Balachandra, N.,(2018) "Energy-Aware and Reliability-Aware Mapping for NoC-Based Architectures", Wireless Personal Communications, 100 (2), 213-225.
15. Naresh Kumar Reddy Beechu, Vasantha Moodabettu Harishchandra, and Nithin Kumar Yernad Balachandra.,(2017) "High-performance and energy-efficient fault-tolerance core mapping in NoC", Sustainable Computing: Informatics and Systems, Vol 16, pp. 1-10.

16. Naresh Kumar Reddy Beechu, Vasantha Moodabettu Harishchandra, Nithin Kumar Yernad Balachandra,(2017) "An energy-efficient fault-aware core mapping in mesh-based network on chip systems", Journal of Network and Computer Applications, Vol 105, pp. 79-87.
17. B. Naresh Kumar Reddy, M.H.Vasantha and Y.B.Nithin Kumar, (2016) "A Gracefully Degrading and Energy-Efficient Fault Tolerant NoC Using Spare core", IEEE Computer Society Annual Symposium on VLSI, pp. 146-151.
18. Beechu, N.K.R., Moodabettu Harishchandra, V. & Yernad Balachandra,, (2017)"Hardware implementation of fault tolerance NoC core mapping," Telecommunication Systems.
19. Naresh Kumar Reddy Becchu, Vasantha Moodabettu Harishchandra, Nithin Kumar Yernad Balachandra, (2017)"System level fault-tolerance core mapping and FPGA-based verification of NoC", Microelectronics Journal, Vol 70, pp. 16-26.
20. C. Wu et al., (2015) "An efficient application mapping approach for the co-optimization of reliability, energy, and performance in reconfigurable NoC architectures", IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 34(8), 12641277.
21. Task graphs for free (TGFF) Available: http://ziyang.eecs.umich.edu/˜dickrp/tgff/ .
22. Vincenzo Catania, Andrea Mineo, Salvatore Monteleone, Maurizio Palesi, and Davide Patti., (2015)"Noxim: An open, extensible and cycle-accurate network on chip simulator", In Proceedings of the 2015 IEEE 26^{th} International Conference on Application-specific Systems, Architectures and Processors (ASAP15), pp. 162-163.