Wireless Personal Communications (2023) 128:2937–2952
https://doi.org/10.1007/s11277-022-10080-x

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

Aruru Sai Kumar[1] · **B. Naresh Kumar Reddy**[2]

Accepted: 28 September 2022 / Published online: 12 October 2022

Abstract
The Network on Chip architecture's performance metrics and inter-core communication are significantly impacted by the acceleration of the evolution of the components integrated on a single chip. Therefore, it is crucial to offer an effective mapping between the cores so that communication between them improves in order to solve such problems. Throughput and latency both have a higher impact on outperforming the network's performance in NoC. In this research paper, an efficient mapping strategy implemented on the real-time embedded applications named ERTEAM is presented. In this algorithm, based on the minimum core average distance 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 is evaluated on a set of embedded applications, which reveals a reduction in latency at 12.3% and 8.4%, the simulation time reduces at an average of 19% and 9.6%, the throughput increases at 14.5% and 7.8% and reduces the communication energy by 15.6% and 5.2% against Branch and Bound Based Mapping (BBPCR) and segmented brute-force mapping respectively. The proposed ERTEAM is simulated and tested on Xilinxs Zynq UltraScale+ MPSoC ZCU104 Evaluation Kit using Xilinx Vivado 2020.2 software platform. The obtained hardware implementation results outperformed the delay and area metrics.

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

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 System on Chip (SoC). As per International Technology Roadmap for Semiconductors, named ITRS 2.0,

✉ Aruru Sai Kumar
asaikumar.nitw@gmail.com

B. Naresh Kumar Reddy
naresh.nitg@gmail.com

[1] Department of ECE, VNR Vignana Jyothi Institute of Engineering and Technology, Hyderabad, Telangana, India

[2] Department of Electronics and Communication Engineering, National Institute of Technology, Tiruchirappalli, Tamil Nadu 620015, India

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. 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.

The NoC design paradigm's integrative research issues could be generally divided into 4 categories. From those mentioned four categories, the application core mapping on NoC platform is identified as one of the most critical and fundamental problems in NoC designs [3, 4]. The mapping designs are basically classifies into 2 types namely, static mapping and dynamic mapping. The static mapping doesn't depend on the networks current state and uses the static paths for transferring the data. The dynamic mapping completely depends on the networks current state and transfer of data occurs on the basis of traffic during the run time.

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 efficient real-time embedded application mapping (ERTEAM). 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 major contributions provided by the proposed ERTEAM algorithm are stated below:

- This mapping approach, is mostly used for real-time embedded systems, reduces the mapping area by using a minimum core average distance (CAD) and maps the PE's inside that region.
- A detailed analysis and explanation of the mathematical calculations for CAD as well as the communication energy were presented using an example.
- This proposed algorithm was used to evaluate each set of real-time embedded applications, improving performance.
- The performance of the proposed method, which outperformed the delay and area metrics, was tested on the Xilinx Zynq UltraScale + MPSoC ZCU104 Evaluation Kit using the Xilinx Vivado 2020.2 software.

The organisation of this research is determined as follows; Sect. 2 provides the related work and, Sect. 3 provides the Model Analysis of NoC architecture. Section 4 explains the details of the proposed mapping strategy and the experimented outcomes represented

in Sect. 5. Section 6 provides the hardware verification for the proposed algorithm. The research paper concludes with Sect. 6 and provides the future scope.

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.

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 et al. [6] implemented a mechanism for mapping the irregular IP's embedded on a regular 2D mesh topology for NoC architectures. The core principle is to break down each big IP into several smaller dummy IP's, each of which can move into a single tile, reducing energy consumption and avoiding congestion.

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. 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. 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.

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. 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, ($C = C_1, C_2, C_3, \ldots, C_n$) 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].

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

3.3 Measurement of Communication Energy

Communication energy is considered the same as the distance between two tiles or nodes [15, 16]. 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) [17].

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 [18–20]. 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 [21].

4 Proposed ERTEAM Technique

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.

$$\forall P_i \in P,$$
$$\forall C_i \in C,$$
$$\Omega(P_i) \in C,$$
$$P_i \neq P_j \Rightarrow \Omega(P_i) \neq \Omega(P_j)$$
$$\forall C_{ij} \in D$$

Let $a_{ij} \in A$ be mapped to some $P_{xy} \in$ C then $P_{xy} = \Omega(C_i) \in$ D.

$\mathrm{CE}(P_i, P_j) = \mathrm{W}(e_{ij}) \times |\,\mathrm{P(i)P(j)}\,|$ in terms of nodes.

$\mathrm{CE}(P_i, P_j) = \mathrm{W}(e_{ij}) \times C_{ij}$

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). $\mathrm{W}(e_{ij})$ 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 $\times CD$
  }
  if *min CE = Total CE ;*
  then Total Communication Energy < min Communication Energy
  end
  Core Mapping = min CE mapping ;
  Total CE (TCE) = $\sum BW_{(P_i,P_j)} \times CD_{(C_i,C_j)}$
while *Next Mapping*;
return Best Communicative Mapping with lowest Communication Energy ;
Calculate Core Mapping Execution Time ;
Calculate Latency and Throughput ;

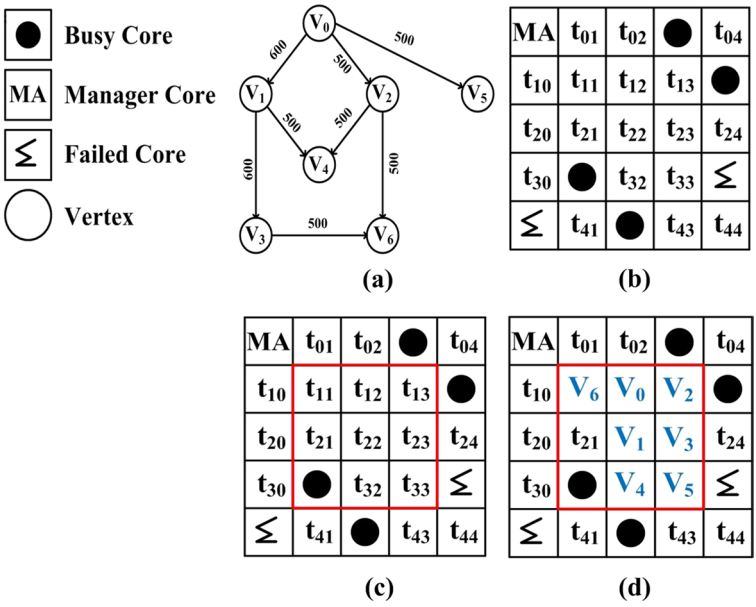

Fig. 1 ERTEAM Algorithm: **a** An example of application task graph, **b** 5×5 mesh NoC, **c** mapping region obtained through minimum CAD, and **d** efficient ERTEAM core mapping

The proposed core mapping algorithm is clearly explained in Algorithm 1. Network core graph (NCG) and NoC architecture graphs (NAG) acts as an input and NoC architecture mapping graph (NMG) acts as output. Initially, select the efficient mapping region using minimum core average distance (CAD) which reduces the mapping area. Then, processing element (PE's) in NCG are mapped on efficient mapping region in NoC according to the minimum communication energy. Figure 1 presents an obvious example with clear explanations. A simple network core graph has shown in Fig. 1a and 5×5 NoC Architecture Graph shown in Fig. 1b. As the number of vertices is 7 in the NCG, the efficient mapping region is selected based on CAD, preferably a size 3×3 region shown in Fig. 1c. Finally, NCG vertices are mapped on 3×3 region according to the minimum communication energy of the network shown in Fig. 1d.

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 [22] and the simulation configuration parameters of the ERTEAM are depicted in Table 2. The best mapping pattern found using a C++ program, the simulations carried out on Noxim simulator [23], and the time consumed can be obtained. .

Table 1 Specifications of embedded applications

Application	No. cores	Network size
H264 encoder (H264_enc)	36	6×6
MP3 decoder (MP3_dec)	16	4×4
Network processing (NP)	16	4×4
MPEG2 encoder (MPEG2)	16	4×4
Multimedia systems (MMS)	16	4×4
Video object plan decoder (VOPD)	16	4×4

Table 2 NoC simulator configuration parameters

Parameter	Value
Traffic	Synthetic
PIR(Injection rate)	0.02 Packets/Cycle/IP
Simulation time	2,00,000 Cycles
Saturation time	10,000 Cycles
Switching time	Wormhole
Flit length	32-Bits
Packet size	64 Flits(1 Header, 62 Payload, 1 Tail)

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).

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

K = Total number of packets reaching their destination cores. L_n = The clock cycle latency for the nth node.

Table 3 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. 2.

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 4 illustrates the obtained simulation time of the proposed algorithm

Table 3 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

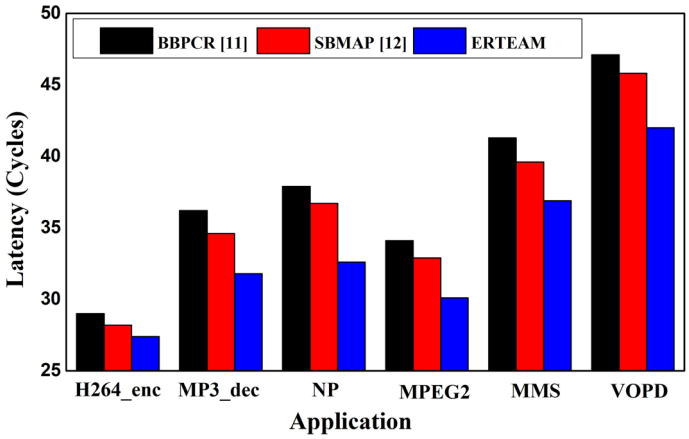

Fig. 2 Latency of the proposed algorithm ERTEAM (in terms of cycles) compared to the BBPCR [11] and SBMAP [12]

Table 4 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

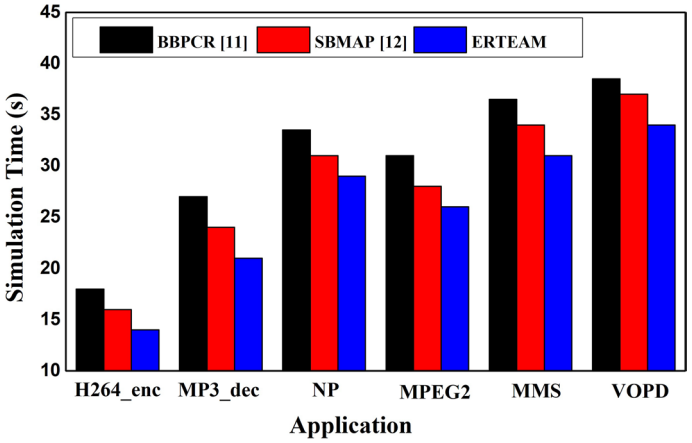

Fig. 3 Simulation Time of the proposed algorithm ERTEAM (in terms of seconds) compared to the BBPCR [11] and SBMAP [12]

ERTEAM (in terms of seconds) compared to the BBPCR [11] and SBMAP [12]. Therefore, the graphical representation of the simulation time depicted in Fig. 3.

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.

Table 5 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. 4.

Table 5 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

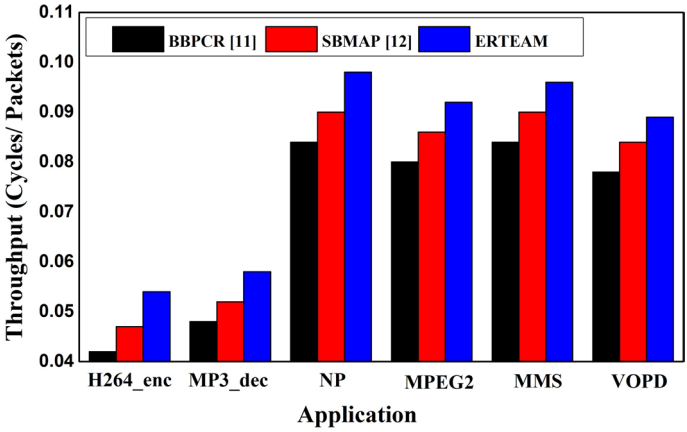

Fig. 4 Throughput of the proposed algorithm ERTEAM (in terms of cycles/packets) compared to the BBPCR [11] and SBMAP [12]

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 6 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. 5.

Table 7 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 6 Communication energy of the proposed algorithm for various embedded applications

Communication energy (μ *J*)			
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

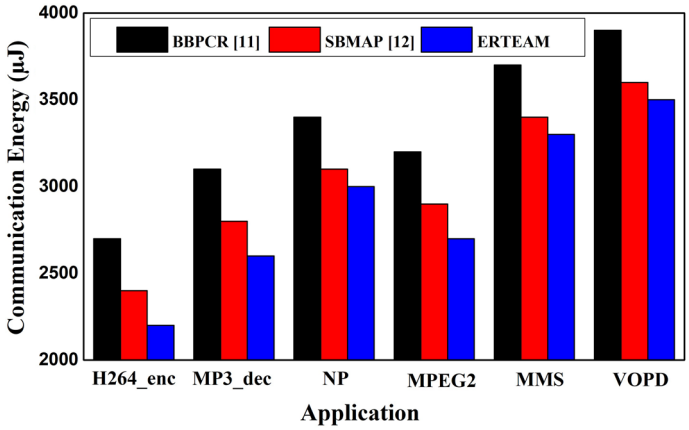

Fig. 5 Communication Energy of the proposed algorithm ERTEAM (in terms of μJ) compared to the BBPCR [11] and SBMAP [12]

Table 7 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 Hardware Implementation

The proposed ERTEAM algorithm is executed and verified on Zynq UltraScale- MPSoC ZCU104 Evaluation Kit by utilizing Xilinx Vivado 2018.3, software platform [24, 25]. The metrics such as delay and area in terms of LUTs for 4-bit, 8-bit, 16-bit, 32-bit, 64-bit were evaluated for the ERTEAM algorithm against BBPCR [11] and SBMAP [12]. The evaluation kit is based on a Zynq UltraScale + XCZU7EV MPSoC, that pairs programmable logic with a processing system based on a quad-core Arm Cortex-A53 application processor and a dual-core Arm Cortex-R5 real-time processor. Each metrics along with the comparative results were elucidated as follows:

6.1 Delay

Delay is considered as the most satisfying performance measure within systems. The overall measure of time required for a communication or packet to transit from its source towards its final destination is referred to as delay. The data transfer rate is indeed a probabilistic parameter. Table 8 illustrates the delay of the proposed algorithm ERTEAM (in

Table 8 Delay(ns) of the proposed ERTEAM algorithm for various bit sizes

Delay (ns)			
Bit sizes	BBPCR [11]	SBMAP [12]	ERTEAM
4	5.2	4.9	3.9
8	9.4	8.9	8.2
16	28.6	22.4	16.4
32	58.6	56.1	28.6
64	120	96	39

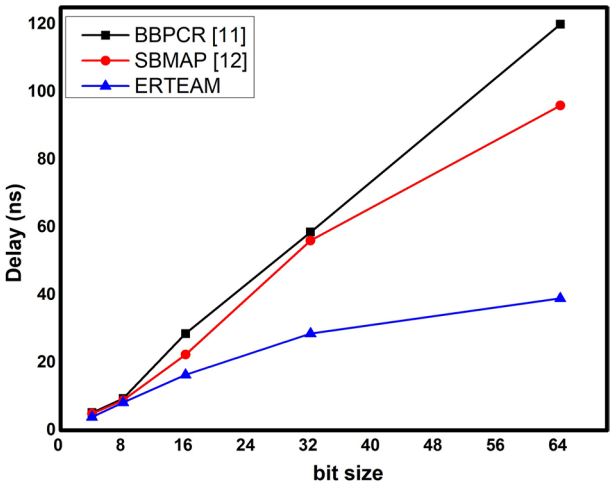

Fig. 6 Delay of the proposed ERTEAM algorithm for various bit sizes

terms of ns) for various bit lengths (4-bit, 8-bit, 16-bit, 32-bit, 64-bit). The Delay metric of ERTEAM reduced by an average of 15.6%, 14.8% compared to BBPCR [11] and SBMAP [12] algorithms. Figure 6 provides the graphical representation of delay of ERTEAM algorithm for various bit sizes.

6.2 Area

Area is also one of the major metrics in terms of performance improvement for NoC applications. Due to the exponential increase in the number of components on the chip, the area is getting increased. So, the proposed ERTEAM algorithm concentrated on this drawback and the Area (in terms of LUTs utilization in FPGA) of ERTEAM reduced by an average of 13%, 8% compared to BBPCR [11] and SBMAP [12] algorithms. Table 9 illustrates the Area (in terms of LUTs utilization in FPGA) of the proposed algorithm ERTEAM for various bit lengths (4-bit, 8-bit, 16-bit, 32-bit, 64-bit). Figure 7 provides the graphical representation of LUTs utilization of ERTEAM algorithm for various bit sizes.

Table 9 LUTs utilization in FPGA of ERTEAM algorithm for various bit sizes

LUTs			
Bit sizes	BBPCR [11]	SBMAP [12]	ERTEAM
4	6	6	6
8	106	98	56
16	522	485	196
32	1987	1724	1156
64	3820	2900	1700

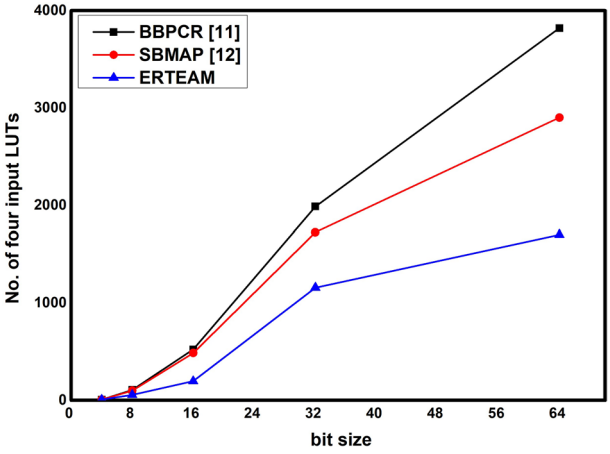

Fig. 7 LUTs utilization in FPGA of the proposed ERTEAM algorithm for various bit sizes

7 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 PE's 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. The hardware verification is carried out through Xilinx Zynq UltraScale + MPSoC ZCU104 Evaluation Kit using Xilinx Vivado 2020.2 software platform, where the test results shows a significant improvement and outperforms the delay and area metrics.

In future, we would like to extend this work by implementing a efficient mapping technique through ML (machine learning) concepts and provide a fault aware core mapping using a spare core replacement methodology. This improves the reliability and performance metrics of the NoC system.

Acknowledgements The author express their sincere gratitude to anonymous reviewers and the editor for their comments and suggestions towards improvement of this manuscript. Thanks to T. V. K. Hanumantha Rao for encouraging research.

Funding Not applicable.

Data Availability The referred papers will be available on request.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

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.

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.

References

1. Benini, L., & Micheli, G. D. (2002). Networks on Chips: A new SoC paradigm. *IEEE Computer, 35*(1), 70–78.
2. Jagadheesh, S., et al. (2022). Reinforcement learning based fault-tolerant routing algorithm for mesh based NoC and its FPGA implementation. *IEEE Access, 10,* 44724–44737. https://doi.org/10.1109/ACCESS.2022.3168992.
3. Sai Kumar, A., & Hanumantha Rao, T. V. K. (2019). Efficient core mapping on customization of NoC platforms. In *2019 IEEE international symposium on smart electronic systems (iSES) (Formerly iNiS)* (pp. 57–62). India: Rourkela.
4. Kumar, A. S., & Hanumantha Rao, T. V. K. (2020). Scalable benchmark synthesis for performance evaluation of NoC core mapping. *Microprocessors and Microsystems, 79,* 103272.
5. Li, B., Wang, X., Singh, A. K., & Mak, T. (2019). On runtime communication and thermal-aware application mapping and defragmentation in 3D NoC systems. *IEEE Transactions on Parallel and Distributed Systems, 30*(12), 2775–2789.
6. Li, G., Wu, J., & Ma, G. (2007). Mapping of irregular IP onto NoC architecture with optimal energy consumption. *Tsinghua Science and Technology, 12*(S1), 146–149.
7. Liu, W., et al. (2018). Thermal-aware task mapping on dynamically reconfigurable Network-on-Chip based multiprocessor System-on-Chip. *IEEE Transactions on Computers, 67*(12), 1818–1834.
8. Jiang, G., Li, Z., Wang, F., & Wei, S. (2015). Mapping of embedded applications on hybrid Networks-on-Chip with multiple switching mechanisms. *IEEE Embedded Systems Letters, 7*(2), 59–62.
9. Bhanu, P. V., Govindan, R., Kattamuri, P., Soumya, J., & Cenkeramaddi, L. R. (2021). Flexible spare core placement in torus topology based NoCs and its validation on an FPGA. *IEEE Access, 9,* 45935–45954.
10. Khan, S., Anjum, S., Gulzari, U. A., Afzal, M. K., Umer, T., & Ishmanov, F. (2018). An efficient algorithm for mapping real time embedded applications on NoC architecture. In *IEEE access*, vol. 6 (pp. 16324–16335).
11. Liu, L., et al. (2015). A flexible energy- and reliability-aware application mapping for NoC-based reconfigurable architectures. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 23*(11), 2566–2580.
12. Khan, S., Anjum, S., Gulzari, U. A., Umer, T., & Kim, B. (2018). Bandwidth-constrained multi-objective segmented brute-force algorithm for efficient mapping of embedded applications on NoC architecture. *IEEE Access, 6,* 11242–11254.
13. Kumar, A. S., Hanumantha Rao, T. V. K., & Kumar Reddy, B. N. (2020) Exact formulas for fault aware core mapping on NoC reliability. In *2020 IEEE 17th India Council international conference (INDICON), New Delhi, India* (pp. 1–5). https://doi.org/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. Beechu, N. K. R., Harishchandra, V. M., & Balachandra, N. K. Y. (2017). High-performance and energy-efficient fault-tolerance core mapping in NoC. *Sustainable Computing: Informatics and Systems, 16,* 1–10.

16. Beechu, N. K. R., Harishchandra, V. M., & Balachandra, N. K. Y. (2017). An energy-efficient fault-aware core mapping in mesh-based network on chip systems. *Journal of Network and Computer Applications, 105,* 79–87.
17. 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* (pp. 146–151).
18. Beechu, N. K. R., Moodabettu Harishchandra, V., & Balachandra, Y. (2017). Hardware implementation of fault tolerance NoC core mapping, Telecommunication Systems.
19. Becchu, N. K. R., et al. (2017). System level fault-tolerance core mapping and FPGA-based verification of NoC. *Microelectronics Journal, 70,* 16–26.
20. Kumar, A. S., Rao, T. H., & Reddy, B. N. K. (2022). Performance and communication energy constrained embedded benchmark for fault tolerant core mapping onto NoC architectures. *International Journal of Ad Hoc and Ubiquitous Computing, 41*(2), 108–117.
21. Wu, C., 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), 1264–1277.
22. Task graphs for free (TGFF). http://ziyang.eecs.umich.edu/~dickrp/tgff/.
23. Catania, V., et al. (2015). Noxim: An open, extensible and cycle-accurate network on chip simulator. In *Proceedings of the 2015 IEEE 26th international conference on application-specific systems, architectures and processors (ASAP15)* (pp. 162–163).
24. Reddy, B., Vasantha, M. H., & Nithin Kumar, Y. B. (2018). Hardware implementation of fault tolerance NoC core mapping. *Telecommunication Systems (TELS), 68,* 621–630.
25. Zynq UltraScale + MPSoC ZCU104 Evaluation Kit. https://www.xilinx.com/products/boards-and-kits/zcu104.html.

Aruru Sai Kumar received B.Tech in Electronics and Communications in 2008 from JNTU Hyderabad, India and M.Tech in VLSI Design in 2010 from V.I.T University, Vellore, India. He is currently pursuing Ph.D. in NIT Warangal, India. His research interests focus on Network-on-Chip architecture design, application mapping in 2-D and 3-D environments and fault tolerant Network-on-Chip.

B. Naresh Kumar Reddy received B.Tech. (in Electronics & Communication Engineering) from Sri Venkateswara University in the year 2010 and M. Tech. (in Embedded Systems) from K.L.University in the year 2012 respectively. His higher education includes full-time Ph.D. (Engg.) research supported by Visvesvaraya Ph.D. Scheme, Government of India at National Institute of Technology, Goa, which got completed in the year 2018. He has also spent with Intel Technology India Pvt. Ltd., Bangalore, as the Graduate Intern Technical. He has been with the Indian Institute of Technology Delhi as a Post-Doctoral Fellow (2020-2021). Dr. B. Naresh Kumar Reddy is currently involved in various teaching/research-related areas like networks-on-chip, design methodologies for system-on chip, VLSI and embedded systems. He is a senior member of IEEE and ACM.