


Received January 9, 2018, accepted February 22, 2018, date of publication March 5, 2018, date of current version April 18, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2811716

An Efficient Algorithm for Mapping Real Time Embedded Applications on NoC Architecture

SARZAMIN KHAN¹, SHERAZ ANJUM², USMAN ALI GULZARI³,
MUHAMMAD KHALIL AFZAL¹ , (Senior Member, IEEE),
TARIQ UMER², (Senior Member, IEEE), AND FARRUH ISHMANOV⁴

¹Department of Electrical Engineering, COMSATS Institute of Information Technology, Wah Cantonment 47040, Pakistan

²Department of Computer Science, COMSATS Institute of Information Technology, Wah Cantonment 47040, Pakistan

³Department of Electrical Engineering, The University of Lahore, Islamabad 44000, Pakistan

⁴Department of Electronics and Communication Engineering, Kwangwoon University, Seoul 01897, South Korea

Corresponding author: Muhammad Khalil Afzal (khalilafzal@ciitwah.edu.pk)

ABSTRACT Network-on-chip (NoC) has appeared to be an impending substitute for the communication paradigm in modern very large scale integration embedded systems. Apart from many design challenges, application mapping on the NoC system is one of the most intractable and challenging optimization problems. In this paper, we propose a hybrid, branch & bound (BB)-based exact mapping (BEMAP) algorithm, for mapping real-time embedded applications on the NoC architecture. The BEMAP optimizes the latency and throughput of the NoC system and minimizes power consumption under the bandwidth constraint. This method utilizes the modular exact and systematic search optimization techniques to obtain a multi-objective optimized solution to the mapping problem of the NoC designs. The proposed algorithm exploits the state-of-the-art BB algorithm, in order to obtain optimized results against its competitors. Experimental results under the benchmarks of several real-time embedded applications show that the proposed algorithm achieves up to 19.93% savings in power consumption and 61.10% improvement in network latency for two dimension mesh and torus topologies.

INDEX TERMS Algorithm, branch & bound, mapping, network-on-chip, optimization.

I. INTRODUCTION

The growing demand for the complex embedded systems and shrinking feature size of the Very Large Scale Integration (VLSI) technology, introduced System-on-Chip (SoC) designs [1]. SoC integrates a large number of Intellectual Property (IP) blocks in a single silicon die to optimize power, latency, and area utilization of the system [2]. In the last decade, SoC has proven its importance in every aspect of life, ranging from industrial to consumer electronics. Due to its demanding position, SoC also stumbles upon some complex design challenges of scalability, modularity and fault tolerance, with the traditional shared bus designs. Network-on-Chip (NoC) has emerged as a potential elucidation for the communication bottleneck in the embedded systems. The Processing Elements (PEs) in a NoC system, communicate data packets through routers over a communication link under a specific network topology [3]–[6]. NoC inherits many features from macro computer networks by updating and incorporating indispensable changes, required for on-chip networks [7], [28].

In the design challenges of the NoC systems, application mapping is one of the structural design requirements to achieve optimal system performance. It is a Non Polynomial (NP)-hard combinatorial optimization problem, mitigating assigned tasks on the available NoC tiles of a specific NoC topology, with an efficient routing algorithm [8]. Application mapping optimization requires $n!$ computations for n number of tasks to map an application on the NoC platform. For example, to map an application of 100 tasks on the PEs of the NoC architecture with a 10×10 , 2D mesh topology, requires 9.33×10^{157} computations to find the optimal solution. It is impractical to solve large combinatorial optimization problem in polynomial time by linear programming methods with existing available commercial computation systems. For mapping smaller applications on NoC architecture, it is likely to solve the problem with the exact optimization model of Integer Linear Programming (ILP) and Mixed Integer Linear Programming (MILP) to acquire an exact optimal solution [10], [11], [27]. Fast search based optimization techniques and heuristics are generally used for such type of NP-hard

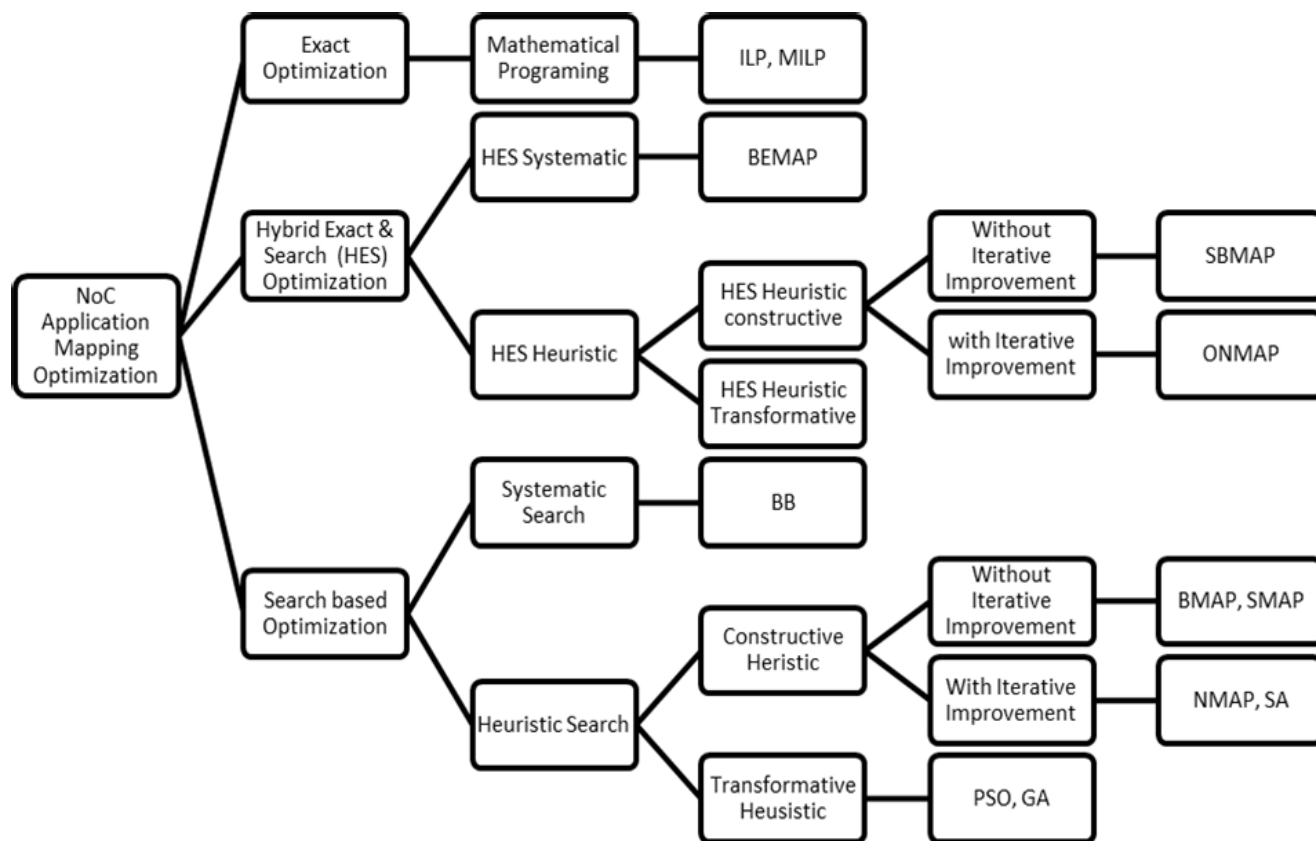


FIGURE 1. Application mapping optimization techniques.

problems [12]–[14]. These methods are fast, because they are based on intelligent search methods, however, they usually generate an infeasible solution due to trap in local minima. To address this dilemma, we have introduced a new hybrid optimization technique, composed of both the exact and search based optimization methods as shown in the middle branch of Fig. 1.

Exact and search based optimization are the two significant search methods, in finding a feasible solution to an NP-hard mapping problem. Exact optimization is composed of mathematical modeling, and it is generally used for small search space to find a real solution to the predicament. To overcome the time convergence problem of the exact optimization, search based optimization techniques are used to find a near optimal solution to the application mapping problem [23]–[26]. Search based optimization is categorized into systematic search and heuristic search methods. Heuristic search method is either transformative or constructive with or without iterative improvement. We have presented a new branch of hybrid optimization, named as Hybrid Exact & Search (HES) optimization. The HES optimization is categorized into HES systematic and HES heuristic optimization methods. HES heuristic is further divided into HES constructive and HES transformative optimization techniques. HES constructive optimization is composed of HES

constructive with iterative improvement and HES constructive without iterative improvement. Our previous two mapping algorithms, namely Segmented Brute-force Mapping algorithm (SBMAP) and Optimized Near-optimal Mapping algorithm (ONMAP), respectively, fall into these two categories.

In this paper, we propose a hybrid, BB based Exact Mapping (BEMAP) algorithm. The BEMAP algorithm includes in the category of HES systematic optimization method. BEMAP utilizes both the properties of systematic search optimization and the exact method. To speed up the exact part of the algorithm, a modular partition based approach is adopted in this research work. The BEMAP algorithm is designed for the topological placement of IPs on the network nodes of the NoC platform. The algorithm utilizes the BB and the modular exact algorithms for mapping the embedded applications. The BB algorithm is fast, however, it usually generates a suboptimal solution to the mapping problem. We have combined this algorithm with our modular exact mapping algorithm to produce an efficient and optimized solution to the mapping problem. The algorithm segregates the initial solution and searches for the best solution, using network cost functions. The cost function is the product of bandwidth and hop counts from source to the destination node of the network. When the algorithm finds the best

solution of the segmented module, it retains the mapping and proceeds to the next module. The algorithm computes the cumulative best mapping in the final stage of the mapping technique. We have implemented the BEMAP algorithm in the NoCTweak simulator [22] to compute network performance parameters under the constraint of bandwidth reservation.

The rest of the paper is structured as follows. Section II presents background work on the NoC application mapping, whereas, Section III describes the problem formulation. In Section IV, we present the proposed BEMAP algorithm. Section V evaluates the mapping results of the BEMAP algorithm by taking real world embedded benchmark applications. Section VI presents concluding remarks and future extension of this research work.

II. RELATED RESEARCH WORK

Low power NoC based Multiprocessor System-on-Chip (MPSoC) designs are the most extensive requirements of the recent and future generation embedded systems. Apart from many design challenges, application mapping optimization problem is one of the key research areas in the field of NoC systems. Power consumption and the network traffic latency are heavily dependent on the mapping of the application on the target NoC architecture, which is an NP-hard optimization problem. By mapping heavily communicating tasks on topologically, near neighborhood IPs of the NoC platform, the average message delay and power consumption can significantly be reduced. To address this important research area, a number of mapping algorithms have been proposed in the literature.

Hu *et al.* [9] presented energy-aware mapping algorithm for NoC architecture, under the bandwidth constraint. The authors introduced BB algorithm to map the application tasks on the NoC tiles. The algorithm accelerated the run time of the mapping process, however, it produced low performance results. Lei and Kumar in [15] proposed a two-step Genetic Algorithm (GA) for mapping task graphs on NoC architecture. The authors minimized the execution time of the scheduling and mapping process of the tasks by reducing the delay between the messages without considering reduction in power and energy consumption. Murali and de Micheli [16] presented Near-optimal Mapping (NMAP) algorithm for the NoC application mapping with the constraint of bandwidth reservation. This algorithm followed the approach that the heavily communicating tasks would be the nearest neighbors in the network to minimize the communication delay. The authors performed a pairwise swapping to improve the results. The traffic is routed across different paths in the network to satisfy the bandwidth requirements. This approach performed better, however, it compromised the results optimality at the cost of simulation time of the architecture. To speed up the computation time of adhoc Simulated Annealing (SA), Lu *et al.* [17] proposed a fast Cluster based Simulated Annealing (CSA) algorithm at the cost of optimized solution.

Radu and Vintan [18] presented an Optimized Simulated Annealing (OSA) algorithm for mapping the embedded application on the 2D mesh topology. In this work, the authors optimized the annealing parameters to speed up the computation time, however, it compromised the performance results as compared to the adhoc SA algorithm. Ascia *et al.* [19] proposed Multi-objective Genetic Algorithm for mapping the IP cores on NoC tiles to optimize network performance and power consumption. Jena [20] proposed GA algorithm for the mapping of IP cores, using 2D mesh topology of the NoC architecture to optimize network power consumption, link bandwidth and network performance. GA is good heuristic for fast convergence time, however, it usually produces infeasible solution to the optimization problem. Sepulveda *et al.* [21] presented Multi Objective Adaptive Immune Algorithm (MAIA), for the optimization of network power and latency of the NoC applications. The research work in [30]–[33] also presented some novel ideas, however, the NoC application mapping is still an open problem for the research community due to its NP-hard nature.

Most of the above cited research work, mainly focused on the speed of simulation time and paid less attention to the outcome of the optimum feasible solution, which is the main objective of the application mapping, particularly for power and energy minimization. The proposed BEMAP algorithm optimizes, mainly the feasible solution of the application mapping for the performance parameters of the NoC systems, in addition to the optimization of simulation time.

III. PROBLEM FORMULATION

To formulate the mapping problem, we present the following definitions and mathematical models for the NoC systems.

Definition 1: A Network Task Graph (NTG), $N = N(T, C)$ is a directed acyclic graph in which each vertex of the graph represents a task ($T = T_1, T_2, T_3, \dots, T_n$) with the associated communication information and deadlines. The directed arc ($c_{i,j} \in C, i = 1, 2, 3, \dots, j = 1, 2, 3, \dots$) between the tasks represents data volume and interdependencies between the application nodes.

Definition 2: A Network Core Graph (NCG), $G = G(P, A)$ is a directed graph in which vertices of the graph represent the available PEs ($P = P_1, P_2, P_3, \dots, P_n$) for the task execution. The directed arc ($a_{i,j} \in A$) shows characteristic parameters and required bandwidth between the IP cores (p_i to p_j).

Definition 3: NoC Architecture Graph (NAG), $A = A(R, H)$ is a topology graph in which node of the graph, ($R = R_1, R_2, R_3, \dots, R_n$) shows a network router and the directed arc, ($h_{i,j} \in H$) represents the routing channel between the routers. The router transmits and receives the data packets of the associated IP core. The routing channel ($h_{i,j}$) provides the physical path for the transmission of packets from source to destination PEs. The routing channel is associated with data bandwidth requirements, $B_{ii,tj}$.

The mathematical models used in BEMAP algorithm are presented as follows:

According to the Bit Energy Model [9] the total energy consumption (E_T) of the NoC architecture is given by:

$$E_T = \sum_{i,j}^n [B_{ii,tj} \times (N_h \times E_S + (N_h - 1) \times E_L)], \quad (1)$$

the parameter, $B_{ii,tj}$ is the arc bandwidth from tile t_i to tile t_j , E_S is the Switch and E_L is the Link Energy. N_h is the Manhattan distance of the NoC architecture, whereas n represents the number of nodes in the architecture for the target application. The Manhattan distance from the source (x_i, y_i) to the destination node (x_j, y_j) of the NoC architecture is given by:

$$N_h = |x_i - x_j| + |y_i - y_j| \quad (2)$$

The communication cost is computed, using the following equation, which is also the performance indicator for application mapping.

$$Cost = \sum_{i,j}^n [B_{ii,tj} \times N_h] \quad (3)$$

For optimized solution the objective functions; Min $\{E_T\}$ or Min $\{Cost\}$ or both must be satisfied.

For computing average latency, throughput, power and energy consumption of the network, the CMOS (Complementary Metal Oxide Semiconductor) Standard Cell Library Model is used for BEMAP algorithm. The CMOS Cell Library Model utilizes the standard CMOS cell library data for computation of the power estimation and performance parameters of the NoC system. In the NoCTweak simulator, RTL designs in Verilog of all the router components were synthesized with Synopsys Design Compiler. The RTL designs were placed and routed with Cadence SoC Encounter, using the CMOS standard cell library. Post-layout data of these components was fed to the simulator for the NoC performance estimation based on the activities of components while running a certain traffic pattern [22]. The average latency of the network under this model is given by:

$$Lt_{av} = \frac{1}{N} \sum_{i=1}^N \frac{1}{N_i} \sum_{j=1}^N Lt_{ij}, \quad (4)$$

where, Lt_{ij} is the packet latency of packet j , N_i is the number of packets received by processor, i after the warm-up time and N is the number of processors in the platform. The average throughput of the network is given by:

$$Th_{av} = \frac{1}{N(T_{sim} - T_{wrm})} \sum_{i=1}^N N_i, \quad (5)$$

where, T_{sim} is the simulation time and T_{wrm} is the warm-up time of the simulations. The average Power of the network is calculated by:

$$P_{Wav} = \frac{1}{N} \sum_{i=1}^N \sum_{j=1}^N [\alpha_{i,j} P_{Wact,j} + (1 - \alpha_{i,j}) P_{Winact,j}], \quad (6)$$

the parameter, $P_{W(act,j)}$ is the post-layout active power and $P_{W(inact,j)}$ is the post-layout inactive power of component, j . The component, $\alpha_{(i,j)}$ is active percentage of the component, j in the router, i (after T_{wrm}). Finally the average energy consumed by each packet in the network is given by:

$$En_p = \frac{(T_{sim} - T_{wrm})}{(NN_p)} \sum_{i=1}^N \sum_{j=1}^N [\alpha_{i,j} P_{Wact,j} + (1 - \alpha_{i,j}) P_{Winact,j}], \quad (7)$$

where, N_p is the total number of packets, traversed across the network of the NoC architecture.

Algorithm 1 General BB Algorithm

1. M (Search space set) $\leftarrow \{X$ (Feasible solution set)
 2. Let z^1 be a lower bound on $\max \{f(x) : x \in X\}$
 3. While $M \neq \emptyset$
 4. do
 5. Select a set X' (Subset) from M
 6. if $|X'| = 1$ then
 7. Let x be the single solution belonging in X'
 8. if $f(x) > z^1$ then
 9. $z^1 \leftarrow f(x)$
 10. end if
 11. else
 12. Let $U_{X'}$ be an upper bound on $\max \{f(x) : x \in X\}$
 13. if $U_{X'} > z^1$ then
 14. Let split up X' into m subsets X_1, X_2, \dots, X_m
 15. Add X_1, X_2, \dots, X_m into M
 16. end
-

IV. BEMAP ALGORITHM

The proposed BEMAP algorithm is composed of BB and Exact mapping algorithms for the topological placement of the PEs on the NoC architecture. The BB algorithm (algorithm 1) solves combinatorial optimization problems by enumerating the feasible solutions through the explorations of a search tree. The algorithm efficiently ambles through the exploration tree that represents the solution space. It forms a tree of the sub problems, as it advances through the solution space. It constructs upper and lower bounds for the root problem. The solution is feasible, if the boundary conditions are satisfied at a particular node; otherwise, the algorithm partitions the node to the child nodes. The search continues until the best solution is found by trimming all the nodes.

Hu and Marculescu [9] exploited the general BB algorithm for the application mapping of the NoC architecture. The author employed the speedup techniques and IP ordering to trim away the non-promising nodes. The algorithm sorted the PEs by their communication demands. NOCMAP, an open source mapping simulator was presented for NoC application mapping. The simulator implemented two mapping algorithms, BB and SA. It employed Bit Energy Model to compute the communication energy consumption by optimal application mapping. The BB mapping algorithm was

Algorithm 2 BEMAP Algorithm

```

1  INPUT1: G (P, A) or N = N (T, C)
2  OUTPUT: G (P, A), N (T, C) → A (R, H)
3  Select T (Topology) from S (Set of Topologies)
4  Run BB (Branch and Bound) Algorithm
5  INPUT2 ←  $M_i$  (BB initial application mapping)
6  N (Number of Segments) = Total tasks/10
7  Initialize task mapping, N=1,  $C_m$  (min Cost) =  $\infty$ 
8  do
9  {
10 Calculate  $N_h$  (Manhattan distance)
11 Calculate  $B_{i,tj}$  (Communication Bandwidth)
12 If the  $B_{i,tj}$  constraint is satisfied {
13 Compute  $C_t$  (total Cost) &  $C_m$  (min Cost)
14 If  $C_t < C_m$  then {
15  $C_t = C_m$ 
16  $M_t$  (Task mapping) ←  $M_m$  (min Cost mapping)} }
17 Compute  $B_l$  (min Link Bandwidth)
18 Compute  $E_T$  (Total Energy)
19 }
20 While (Next Mapping)
21 N++
22 Compute  $En_p$  (Energy),  $Pw_{av}$  (Power)
23 Compute  $Lt_{av}$  (Latency),  $Th_{av}$  (Throughput)
24 Return best  $M_c$  (segments cumulative mapping)
25 End

```

used for topological placement of PEs onto NoC platform to minimize energy consumption under the constraint of bandwidth reservation. The author implemented Energy and Performance Aware Mapping (EPAM) in the simulator with XY dimension ordered, Odd-Even (OE) and West-First (WF) routing algorithms. ReliableNoC [29] is an extended version of the NOCMAP simulator with the addition of the reliability algorithm for the NoC architecture.

Our proposed algorithm (algorithm 2) integrates the BB algorithm of the NOCMAP and the modular exact optimization method, to develop an efficient hybrid mapping algorithm named as BEMAP. As we assume one to one mapping, therefore the algorithm gets input data from Network Task Graph (NTG) and Network Core Graph (NCG) for mapping the application. The BB Algorithm generates the initial solution for mapping the application on the target NoC topology. This mapping contributes as an input source for the exact part of the algorithm. The algorithm splits this mapping into small modules, each containing at the most, ten IPs, depending on the speed and size of the search space. The value ten for each segment is chosen, because beyond this value the simulation speed drastically reduces, due to the NP-hard nature of the problem as shown in Fig. 2.

The proposed BEMAP as well as BB mapping algorithms are implemented in the open source NoCTweak simulator for analysis and evaluation of the results. NoCTweak is a SystemC based NoC simulator developed by A. T. Tran and B. M. Baas [22]. BEMAP is a multi-objective

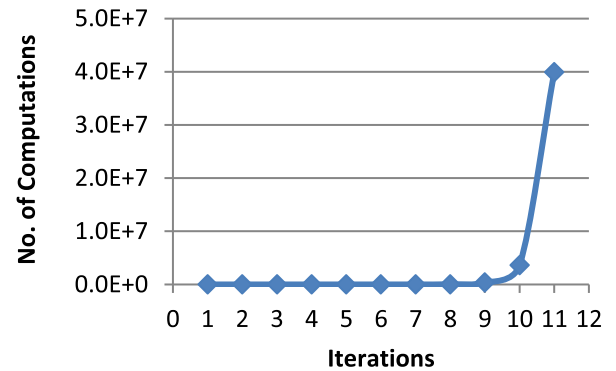


FIGURE 2. Reduction in simulation speed with system size.

mapping algorithm that computes and optimizes power, latency and throughput under the constraint of link bandwidth. Fig. 3 shows the flow of the algorithm, using an arbitrary example. The initial mapping is performed on the NTG/NCG as supplied in the input file by the user to the simulator. The algorithm then generates application mapping, using BB algorithm. The BEMAP algorithm splits the BB mapping into different modules and applies modular systematic mapping optimization method. The size of the module is user defined and can be changed in the user configuration settings.

Fig. 4 shows the mapping of Context Adaptive Variable Length Coding (CAVLC) benchmark through BEMAP algorithm. The vertices show the processing tasks and the arcs represent the communication bandwidth with the directional traffic flow. In this case, we take the number of tasks in the NTG, equivalent to the number of PEs of the NCG, therefore scheduling is not required. The NCG is, therefore mapped on 4 X 4, NAG of 2D mesh topology, which reveals that the heavily communicating tasks are placed close to each other by the algorithm for optimized performance results. For example, PE_{10} and PE_9 have a traffic volume of 1428 MB/S, therefore, they are placed close to each other on Tile (0, 0) and (0, 1), respectively, with one hop count i.e. with minimum communication cost. Similarly, PE_7 is mapped on Tile (1, 1) close to PE_9 and so on. The algorithm utilizes the Bit Energy Model to find the optimized mapping by the minimum cost computation method. The NoC performance parameters are calculated, using the CMOS Cell Library Model.

V. RESULTS AND ANALYSIS

NoCTweak simulator is utilized to evaluate the performance of BEMAP algorithm. NoCTweak has already embedded NMAP for application mapping of the NoC benchmarks. We have designed and implemented the BEMAP algorithm in the NoCTweak simulator along with BB algorithm of the NOCMAP simulator. Five real time embedded benchmarks are considered to evaluate the performance of the proposed BEMAP algorithm against its competitors. The Context Adaptive Variable Length Coding (CAVLC), Auto Industrial (AUTO-IND), TELECOM, Multi-Media System (MMS) and Video Object Plan Decoder (VOPD) benchmark

TABLE 1. Platform description for simulation environment.

Network type	2D MESH, TORUS
Platform type	EMBEDDED
Embedded applications	CAVLC, AUTO-IND, TELECOM, MMS, VOPD
Packet delivery type	WITHOUT ACK
Sending ACK policy	SEND ACK OPTIMALLY
Packet distribution	EXPONENTIAL
Fixed packet length	10 (flits)
Flit injection rate	0.5 (flits/cycle/node)
Mapping algorithm	BB, BEMAP, NMAP
Router type	WORMHOLE-PIPELINE
Routing algorithm	XY DIMENSION-ORDERED
Output channel selection	XY-ORDERED
Buffer size	8 (flits)
Inter-router link length	1000 (um)
Pipeline type	8
Pipeline stages	4
Input voltage	1 (V)
Input clock frequency	2000 (MHz)
Operating clock frequency	2000 (MHz)
Warmup time	20000 cycles

applications are collected from literature and used for the analysis and evaluation of BEMAP algorithm [9], [22].

A. 2D MESH TOPOLOGY

We have selected 2D mesh topology, wormhole XY dimension order routing with 2GHz clock frequency and 0.5 (flits/cycle/node) traffic injection rate, for configuration setting of the simulator as shown in Table 1. The absolute results for power, communication cost and latency of BB, BEMAP and NMAP algorithms for the listed applications are shown in Table 2. The percentage savings for these parameters are also shown in Table 3.

TABLE 2. Network power, communication cost and latency of 2D mesh topology.

Power (μ W)			
	BB	BEMAP	NMAP
CAVLC	72883.9	72284.5	73628.1
AUTO-IND	179770.7	157738.4	164007.1
TELECOM	149605.0	146922.6	156656.5
MMS	103168.5	102870.8	103015.8
VOPD	106742.6	106742.6	108237.8
Cost (Bw X Nh)			
CAVLC	6778	6701	6968
AUTO-IND	139	131	132
TELECOM	13660	13054	14834
MMS	671938	664636	667628
VOPD	4119	4119	4265
Latency (Cycles)			
CAVLC	20.5	20.3	20.6
AUTO-IND	4470.4	3072.5	4943.7
TELECOM	6953.2	7029.4	6845.5
MMS	60.2	52.7	45.4
VOPD	154.4	154.4	111.1

The absolute values in Table 2 show that BEMAP performs better as compared to BB and NMAP algorithms and produces improved results for the network power consumption and communication cost. BEMAP has 0.83%, 13.97%, 1.83%, and 0.29% improvement in power consumption than BB under CAVLC, AUTO-IND, TELECOM, MMS and VOPD applications, respectively as shown in Table 3. The

1. Initial Mapping

Tasks	T0	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12	T13	T14	T15
Tiles	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

2. Branch and Bound (BB) Mapping

Tasks	T0	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12	T13	T14	T15
Tiles	5	4	2	3	1	0	15	7	14	6	10	11	12	13	9	8

3. Splitting BB Mapping and applying Exact method

Tasks	T0	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12	T13	T14	T15
Tiles	5	4	2	3	1	0	15	7	14	6	10	11	12	13	9	8

4. BEMAP Results

Tasks	T0	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12	T13	T14	T15
Tiles	3	0	2	1	7	4	15	5	8	13	9	12	11	10	6	14

FIGURE 3. BEMAP illustration with example.

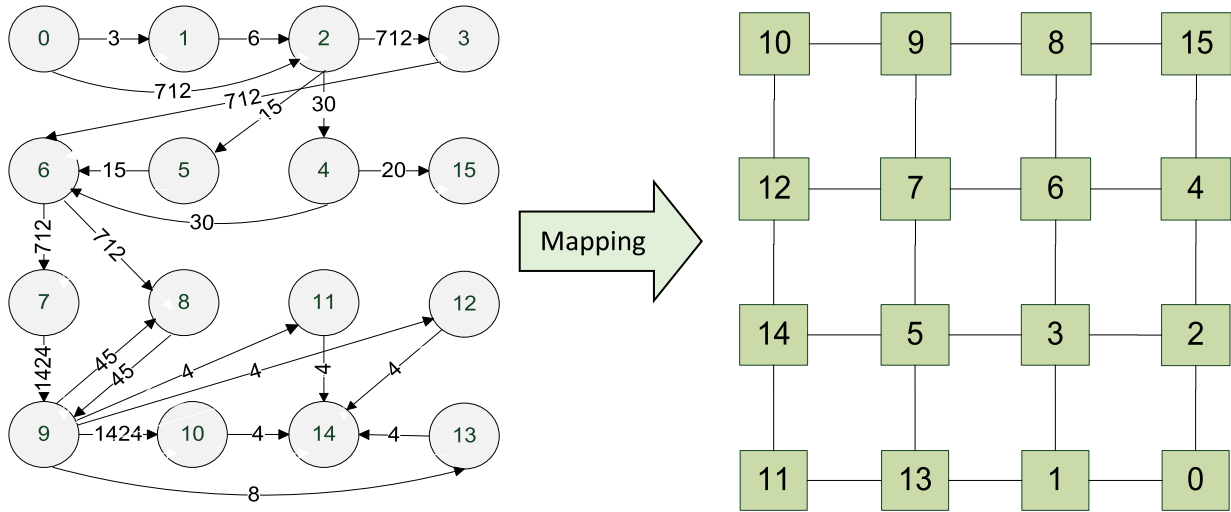


FIGURE 4. Mapping of Context Adaptive Variable Length Coding (CAVLC) application on NoC architecture, with BEMAP algorithm.

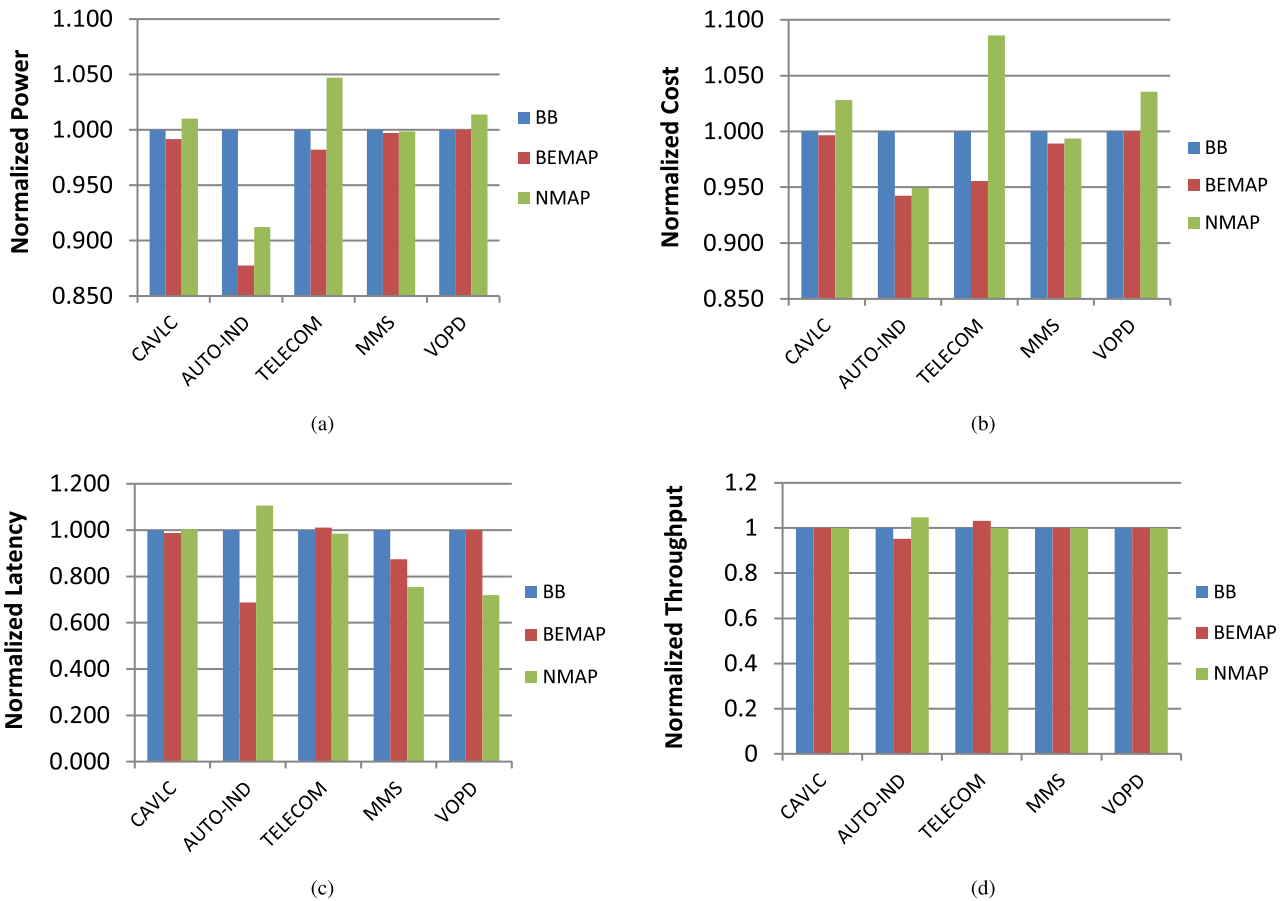


FIGURE 5. NoC performance parameters of 2D mesh topology, normalized to BB. (a) Network power consumption. (b) Network communication cost. (c) Network latency. (d) Network throughput.

power improvements of BEMAP as compared to NMAP algorithm are 1.86, 3.97, 6.63 and 0.14 percent, respectively for the above listed applications. The network cost savings of BEMAP algorithm are 1.15%, 6.11%, 4.64%, and 1.10% as compared to BB, and 3.98%, 0.76%, 13.64%, and 0.45% as compared to NMAP algorithm for CAVLC, AUTO-IND,

TELECOM and MMS applications, respectively. The cost and power consumption of BB and BEMAP are identical for VOPD application (Table 2), because this is the lowest possible communication cost for VOPD, which is also confirmed by the ILP method. The cost and power improvements of the BEMAP algorithm for VOPD are 3.54% and 1.38% as

TABLE 3. Network power, communication cost and latency savings of 2D mesh topology.

	CAVLC	AUTO-IND	TELECOM	MMS	VOPD
Power Savings (%)					
BB	0.83	13.97	1.83	0.29	0.00
NMAP	1.86	3.97	6.63	0.14	1.38
Cost Savings (%)					
BB	1.15	6.11	4.64	1.10	0.00
NMAP	3.98	0.76	13.64	0.45	3.54
Latency Savings (%)					
BB	0.99	45.50	-1.08	14.23	0.00
NMAP	1.48	60.90	-2.62	-13.85	-38.97

TABLE 4. Simulation time and network throughput of 2D mesh topology.

Simulation Time (S)			
	BB	BEMAP	NMAP
CAVLC	7.0	8.0	6.0
AUTO-IND	12.0	28.0	14.0
TELECOM	8.0	9.0	8.0
MMS	11.0	22.0	11.0
VOPD	7.0	8.0	7.0
Throughput (Cycles/S)			
CAVLC	0.015	0.015	0.015
AUTO-IND	0.021	0.022	0.022
TELECOM	0.031	0.032	0.031
MMS	0.012	0.012	0.012
VOPD	0.024	0.024	0.024

compared to NMAP algorithm. The normalized results of the performance parameters with respect to BB algorithm are also shown in Fig. 5. The power consumption and network cost of BEMAP algorithm is better for most of the applications as shown in Fig. 5 (a) and Fig. 5 (b).

The latency improvements of BEMAP algorithm are 0.99%, 45.50% and 14.23% for CAVLC, AUTO-IND, and MMS applications, respectively as compared to BB algorithm. BEMAP has 1.48% and 60.90% lower latency as compared to NMAP algorithm for CAVLC and AUTO-IND applications as shown in Table 3 and Fig. 5 (c). The latency of NMAP algorithm for some applications like VOPD is better,

TABLE 5. Network power, communication cost and latency of torus topology.

Power (μ W)			
	BB	BEMAP	NMAP
CAVLC	73408.4	73401.9	84844.7
AUTO-IND	179770.7	157738.4	177275.7
TELECOM	149605	146922.6	156656.5
MMS	103271.8	103168.5	123726.2
VOPD	111742.6	111662.1	115232.1
Cost ($B_w \times N_h$)			
	BB	BEMAP	NMAP
CAVLC	6705	6701	6758
AUTO-IND	135	131	135
TELECOM	13660	13054	13410
MMS	664727	658309	658551
VOPD	4119	4103	4119
Latency (Cycles)			
	BB	BEMAP	NMAP
CAVLC	20.4	20.3	26.7
AUTO-IND	4470.4	3072.5	4949.7
TELECOM	6953.2	7029.5	6845.5
MMS	60.3	48.6	74.0
VOPD	154.4	109.6	129.1

however, the communication cost and power consumption is worse for most of the applications.

The simulation time and throughput comparison of BEMAP, BB and NMAP algorithm are shown in Table 4. The simulation time of BEMAP algorithm is comparable with BB and NMAP algorithm and is less than half a minute for all of these benchmarks. The network has no congestion due to the uniform traffic injection rate, therefore; throughput is almost constant for these benchmark applications as shown in Fig. 5 (d). The comparative analysis in this Section, shows that the quality of the performance parameters of the proposed algorithm is better than BB and NMAP algorithms with comparable simulation time.

B. TORUS TOPOLOGY

We have also utilized torus topology, in order to get additional results for the quantitative analysis of the proposed algorithm under different topology. The absolute values of the performance parameters of BB, NMAP and BEMAP algorithms for CAVLC, AUTO-IND, TELECOM, MMS and VOPD applications are shown in Table 5. The proposed algorithm produces optimized results for most of the embedded application benchmarks as shown in Table 6 and Fig. 6.

The power savings of the proposed algorithm are up to 13.97% as compared to BB algorithm and 19.93% compared

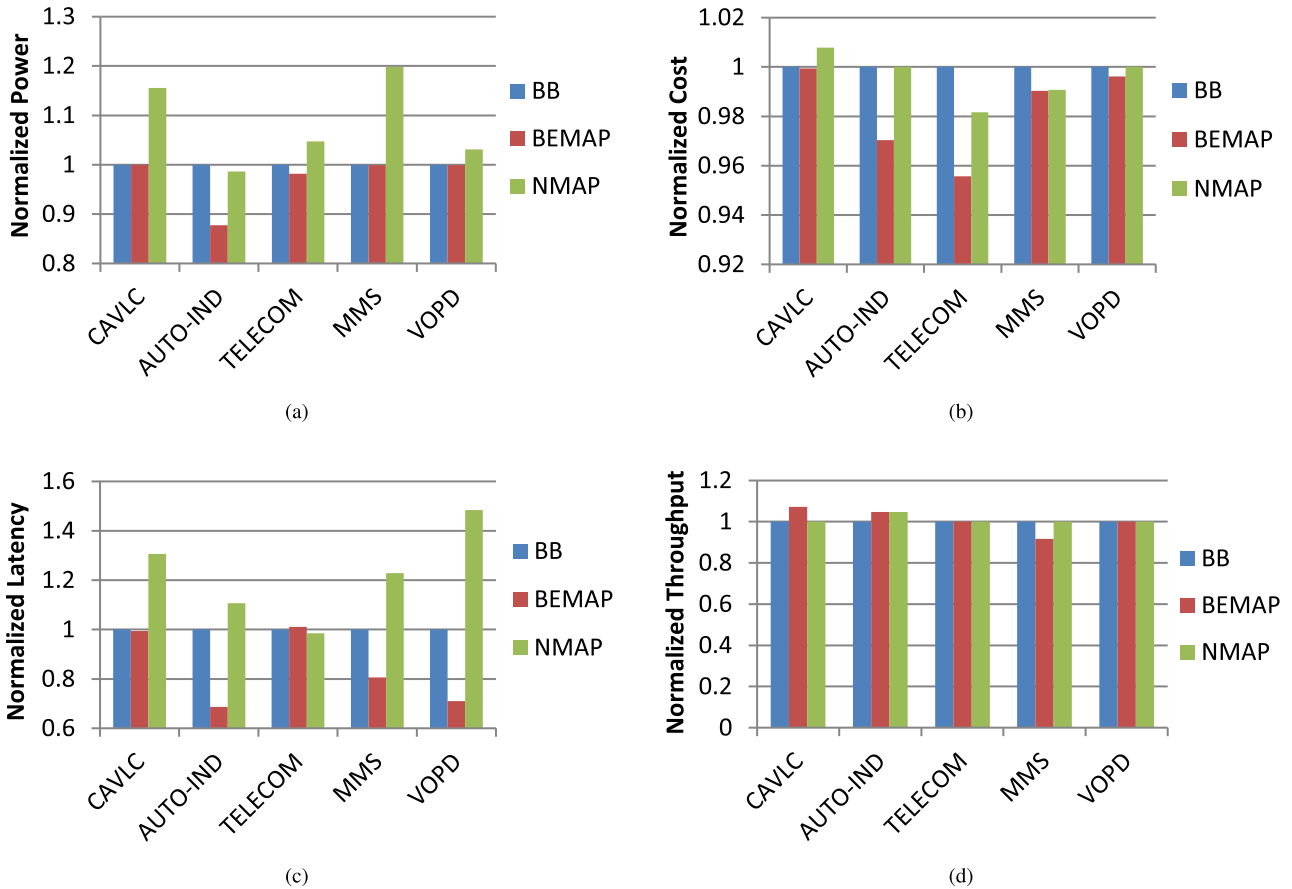


FIGURE 6. NoC performance parameters of torus topology, normalized to BB. (a) Network power consumption. (b) Network communication cost. (c) Network latency. (d) Network throughput.

TABLE 6. Network power, communication cost and latency savings of torus topology.

	CAVLC	AUTO-IND	TELECOM	MMS	VOPD
Power Savings (%)					
BB	0.01	13.97	1.83	0.10	0.07
NMAP	15.59	12.39	6.63	19.93	3.20
Cost Savings (%)					
BB	0.1	3.05	4.64	0.97	0.38
NMAP	0.79	3.05	2.73	0.04	0.38
Latency Savings (%)					
BB	0.1	45.50	-1.09	24.07	40.94
NMAP	30.67	61.10	-2.62	52.39	17.79

to the NMAP algorithm. The communication cost savings are up to 4.64% and 3.05% for BB and NMAP algorithms, respectively. The latency improvement by the application of

TABLE 7. Simulation time and network throughput of torus topology.

	Simulation Time (S)		
	BB	BEMAP	NMAP
CAVLC	6	9	8
AUTO-IND	12	22	12
TELECOM	7	9	8
MMS	11	17	11
VOPD	7	10	7
Throughput (Cycles/S)			
CAVLC	0.014	0.015	0.014
AUTO-IND	0.021	0.022	0.022
TELECOM	0.031	0.031	0.031
MMS	0.012	0.011	0.012
VOPD	0.024	0.024	0.024

BEMAP algorithm is very impressive that goes up to 45.50% and 61.10% as compared to BB and NMAP algorithms, respectively. The improvement in latency is due to the inherent characteristics of the bypass links in the torus topology.

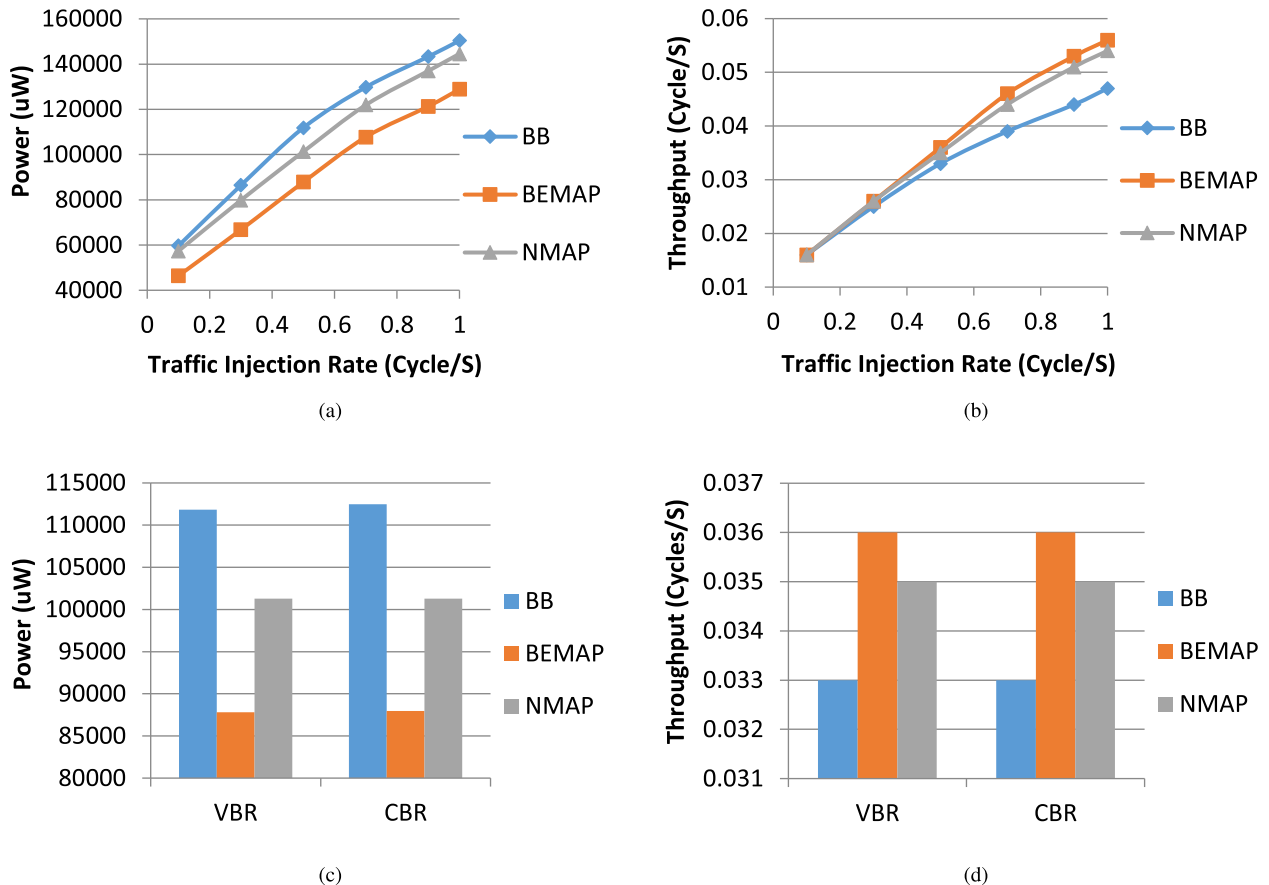


FIGURE 7. Performance comparison of BEMAP algorithm with traffic workloads. (a) Power versus traffic injection rate. (b) Throughput versus traffic injection rate. (c) Power versus CBR and VBR traffic. (d) Throughput versus CBR and VBR traffic.

In addition, the simulation time and throughput results are also comparable to BB and NMAP algorithms as shown in Table 7. The throughput produced by the proposed algorithm is either equal or better than BB and NMAP algorithms. The simulation results proved that the proposed algorithm outperforms for most of the performance parameters and can be used effectively, for better and optimized mapping results than its competitor algorithms.

C. TRAFFIC ANALYSIS

To check the response of the proposed BEMAP algorithm for traffic variations, the traffic is injected at different rates and intervals to the network. We have mapped the Picture In Picture (PIP) real time application on the NoC architecture of 2D mesh topology, using BB, BEMAP and NMAP algorithms. The experimental results reveal that the power consumption of the BEMAP algorithm is lower and much better even at 100% traffic workload as shown in Fig. 7 (a). This is because the proposed algorithm intelligently distributes the traffic in the network and maps the heavily communicating IPs close to each other to avoid traffic congestion. The throughput of the BB, BEMAP and NMAP algorithms is almost uniform and identical at below 50% of the injected traffic. At higher traffic

rates the BEMAP algorithm performs better than BB and NMAP algorithms as shown in Fig. 7 (b). The improvement in throughput is due to the selection of lower Manhattan distance of the overloaded IPs for the packets traversal in the network.

The algorithm is also validated for different types of network traffics such as Constant Bit Rate (CBR) and Variable Bit Rate (VBR). The power consumption of the BEMAP algorithm is much lower and identical in both, CBR and VBR traffic for PIP application as shown in Fig. 7 (c). The BEMAP algorithm has also obtained better throughput at different streams of traffic as compared to its competitor algorithms (Fig. 7 (d)).

These results show that the algorithm possesses the Quality of Service (QoS) with Guaranteed Throughput (GT), because the proposed algorithm ensures no packet loss even at heavy traffic loads. We have evaluated the algorithm with XY dimension-ordered routing algorithm, however, the proposed algorithm with NoCTweak simulator can also be utilized, using Negative-First (NF), West-First (WF), North-Last (NL) and Odd-Even (OE) minimal adaptive routing algorithms that guarantee QoS and avoid packets collision. The BEMAP algorithm has deadlock and livelock free traffic flow with these adaptive routing algorithms. The BB algorithm has

the QoS, using Best Effort (BE) service, because at lower traffic injection rate, the algorithm performs better, however, at higher traffic workloads, its performance degrades both for power consumption and network throughput.

VI. CONCLUSION

In this paper, we proposed a hybrid BEMAP algorithm that is based on hybrid optimization techniques for mapping of embedded applications on the NoC system. BEMAP algorithm is composed of BB algorithm and modular exact systematic search method. In this technique, the fast Branch & Bound (BB) algorithm is used for the initial mapping, which is then optimized by the modular exact optimization method. The Bit Energy Model and CMOS Cell Library Model are used for the computation of the cost effective mapping of the NoC performance parameters under the constraint of link bandwidth reservation. The algorithm efficiently maps the selected real world application workloads on the available NoC tiles of the 2D mesh and torus topologies. BEMAP generates efficient mapping of the real benchmark applications than its competitors, NMAP and BB in terms of power, throughput and latency with comparable simulation time. The algorithm performs effectively even at higher traffic workloads with low power consumption and high throughput. In the future research work, we will utilize the proposed algorithm for mapping the real world benchmark applications on other NoC topologies such as cmesh, butterfly and binary tree.

ACKNOWLEDGMENT

The present research has been conducted by the Research Grant of Kwangwoon University in 2018.

REFERENCES

- [1] L. Benini and G. De Micheli, "Networks on chips: A new SoC paradigm," *IEEE Comput. Soc.*, vol. 35, no. 1, pp. 70–78, Jan. 2000.
- [2] W. J. Dally and B. Towles, "Route packets, not wires: On-chip interconnection networks," in *Proc. 38th DAC*, 2001, pp. 684–689.
- [3] H. Jantch, H. Tenhunen, and N. O. Chip, *Networks on Chips*. Norwell, MA, USA: Kluwer, 2003.
- [4] U. A. Gulzari, S. Khan, S. Aghaa, S. Anjum, and F. S. Torres, "Efficient and scalable cross-by-pass-mesh topology for networks-on-chip," *IET Comput. Digit. Techn.*, vol. 11, no. 4, pp. 140–148, Feb. 2017.
- [5] U. A. Gulzari, M. Sajid, S. Anjum, S. Agha, and F. S. Torres, "A new cross-by-pass-torus architecture based on CBP-mesh and torus interconnection for on-chip communication," *PLoS ONE*, vol. 11, no. 12, p. e0167590, Dec. 2016.
- [6] S. Anjum, J. Chen, P.-P. Yue, and J. Liu, "Delay optimized architecture for on-chip communication," *J. Electron. Sci. Technol.*, vol. 7, no. 2, pp. 104–109, Jun. 2009.
- [7] D. Atienza, F. Angiolini, S. Murali, A. Pullini, L. Benini, and G. De Micheli, "Network-on-chip design and synthesis outlook," *VLSI J. Integr.*, vol. 41, no. 3, pp. 340–359, May 2008.
- [8] R. Marculescu, U. Y. Ogras, L.-S. Peh, N. E. Jerger, and Y. Hoskote, "Outstanding research problems in NoC design: System, microarchitecture, and circuit perspectives," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 28, no. 1, pp. 3–21, Jan. 2009.
- [9] J. Hu and R. Marculescu, "Energy-aware mapping for tile-based NoC architectures under performance constraints," in *Proc. Asia South Pacific Design Autom. Conf.*, Jan. 2003, pp. 233–239.
- [10] S. Tosun, O. Ozturk, and M. Ozen, "An ILP formulation for application mapping onto Network-on-Chips," in *Proc. Int. Conf. Appl. Inf. Commun. Technol.*, 2009, pp. 1–5.
- [11] A. Bender, "MILP based task mapping for heterogeneous multiprocessor systems," in *Proc. Int. Conf. Design Autom. (EURO-DAC)*, 1996, pp. 190–197.
- [12] Y. Chen, L. Xie, and J. Li, "An energy-aware heuristic constructive mapping algorithm for network on chip," in *Proc. Int. Conf. ASIC (ASICON)*, 2009, pp. 101–104.
- [13] A. K. Singh, T. Srikanthan, A. Kumar, and W. Jigang, "Communication-aware heuristics for run-time task mapping on NoC-based MPSoC platforms," *J. Syst. Archit.*, vol. 56, no. 7, pp. 242–255, 2010.
- [14] S. Tosun, "New heuristic algorithms for energy aware application mapping and routing on mesh-based NoCs," *J. Syst. Archit.*, vol. 57, no. 1, pp. 69–78, Jan. 2011.
- [15] T. Lei and S. Kumar, "A two-step genetic algorithm for mapping task graphs to a network on chip architecture," in *Proc. IEEE Euromicro Symp. Digit. Syst. Design*, Sep. 2003, pp. 180–187.
- [16] S. Murali and G. De Micheli, "Bandwidth-constrained mapping of cores onto NoC architectures," in *Proc. IEEE/ACM Design Autom. Test Eur. Conf.*, vol. 2, Paris, France, Feb. 2004, pp. 896–901.
- [17] Z. Lu, L. Xia, and A. Jantsch, "Cluster-based simulated annealing for mapping cores onto 2D mesh networks on chip," in *Proc. 11th IEEE Comput. Soc. Workshop Design Diag. Electron. Circuits Syst.*, Bratislava, Slovakia, Apr. 2008, pp. 1–6.
- [18] C. Radu and L. Vințan, "Optimized simulated annealing for network-on-chip application mapping," in *Proc. 18th Int. Conf. Control Syst. Comput. Sci.*, Jun. 2011, pp. 452–459.
- [19] G. Ascia, V. Catania, and M. Palesi, "Multi-objective mapping for mesh-based NoC architectures," in *Proc. ICHSC/ICSS*, Sep. 2004, pp. 182–187.
- [20] R. K. Jena, "Application mapping of mesh based NoC using evolutionary algorithm," *J. Inf. Syst. Commun.*, vol. 3, no. 1, pp. 203–206, 2012.
- [21] M. J. Sepúlveda, M. Strum, W. J. Chau, and G. Gogniat, "A multi-objective approach for multi-application NoC mapping," in *Proc. IEEE Latin Amer. Symp. Circuits Syst. (LASCAS)*, 2011, pp. 1–4.
- [22] A. T. Tran and B. M. Baas, "NoCTweak: A highly parameterizable simulator for early exploration of performance and energy of networks on-chip," VLSI Comput. Lab, Dept. Elect. Comput. Eng., Univ. California, Davis, CA, USA, Tech. Rep. ECE-VCL-012-2, Jul. 2012.
- [23] T. Liu, S. Yin, J. Liu, and L. Teng, "Hybrid quantum genetic algorithm used for low-power mapping in network-on-chip," *J. Softw. Eng.*, vol. 11, no. 2, pp. 194–201, 2017.
- [24] C. Xu, Y. Liu, Z. Zhu, and Y. Yang, "An efficient energy and thermal-aware mapping for regular network-on-chip," *IEICE Electron. Exp.*, vol. 14, no. 17, pp. 1–11, 2017.
- [25] C. Wu et al., "An efficient application mapping approach for the co-optimization of reliability, energy, and performance in reconfigurable noc architectures," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 34, no. 8, pp. 1264–1277, Aug. 2015.
- [26] C. Wu et al., "A multi-objective model oriented mapping approach for NoC-based computing systems," *IEEE Trans. Parallel Distrib. Syst.*, vol. 28, no. 3, pp. 662–676, Mar. 2017.
- [27] Y. Hu, D. Müller-Gritschneider, G. Gogniat, U. Schlichtmann, and M. J. Sepúlveda, "Automatic ILP-based Firewall Insertion for Secure Application-Specific Networks-on-Chip," in *Proc. Interconnection Netw. Archit., On-Chip, Multi-Chip (INA-OCMC)*, 2015, pp. 9–12.
- [28] S. Khan, S. Anjum, U. A. Gulzari, and F. S. Torres, "Comparative analysis of network-on-chip simulation tools," *IET Comput. Digit. Techn.*, vol. 12, no. 1, pp. 30–38, Sep. 2017.
- [29] C. Ababei, H. S. Kia, O. P. Yadav, and J. Hu, "Energy and reliability oriented mapping for regular networks-on-chip," in *Proc. 5th ACM/IEEE Int. Symp. Netw. Chip (NOCS)*, May 2011, pp. 121–128.
- [30] L. Yang, W. Liu, P. Chen, N. Guan, and M. Li, "Task mapping on SMART NoC: Contention matters, not the distance," in *Proc. 54th ACM/EDAC/IEEE Design Autom. Conf. (DAC)*, Jun. 2017, pp. 1–6.
- [31] S. Tosun, V. B. Ajabshir, O. Mercanoglu, and O. Ozturk, "Fault-tolerant topology generation method for application-specific network-on-chips," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 34, no. 9, pp. 1495–1508, Sep. 2015.
- [32] D. Zhu, L. Chen, T. M. Pinkston, and M. Pedram, "TAPP: Temperature-aware application mapping for NoC-based many-core processors," in *Proc. Design, Autom. Test Eur. Conf. Exhib. (DATE)*, 2015, pp. 1241–1244.
- [33] C. Marcon, T. Webber, and A. A. Susin, "Models of computation for NoC mapping: Timing and energy saving awareness," *Microelectron. J.*, vol. 60, pp. 129–143, Feb. 2017.



SARZAMIN KHAN received the B.E degree in electronic engineering and the M.S. degree in electrical engineering from the NED University of Engineering and Technology, Karachi, Pakistan, in 2001 and 2003, respectively. He is currently pursuing the Ph.D. degree with the Department of Electrical Engineering, COMSATS Institute of Information Technology, Pakistan. He has over 10 years of industrial experience at different positions. His research interests include the design and analysis of analog and digital systems, communication systems, system-on-chip, network-on-chip, embedded systems, and optimization techniques.



SHERAZ ANJUM received the M.Sc. degree in electronics from Quaid-i-Azam University, Islamabad, Pakistan, in 1999, the M.Sc. degree in computer engineering from the University of Engineering and Technology at Taxila, Pakistan, in 2005, under the supervision of Dr. S. A. Khan, and the Ph.D. degree in microelectronics and solid-state electronics engineering from the Institute of Microelectronics, University of Chinese Academy of Sciences, Beijing, China, in 2008, under the supervision of Dr. J. Chen. From 1999 to 2001, he was with the Barani Institute of Information Technology as a Research Associate. From 2001 to 2005 and from 2008 to 2012, he was with the COMSATS Institute of Information Technology, Wah Cantonment, Pakistan, as a Lecturer and an Assistant Professor, respectively. He has over 18 years of teaching and research experience. He is currently an Associate Professor with the COMSATS Institute of Information Technology. His research interests include but not limited to digital system design, the design and analysis of networks-on-chip architectures and algorithms, reconfigurable architectures, multi-processor heterogeneous computing, and advance DSP Architectures. Since 2015, Prof. Anjum has been a member of IET and an Executive Member of the IET Islamabad Local Network. He is currently serving as a reviewer for many reputed international journals. From 2010 to 2017, he served in the TPC of the international conference FIT. He is registered as a Chartered Engineer with ECUK and as a Registered Engineer with Pakistan Engineering Council.



USMAN ALI GULZARI received the B.S. and M.S. degrees in computer engineering and the Ph.D. degree in networks-on-chip from the COMSATS Institute of Information and Technology, Islamabad, in 2000, 2003, and 2017, respectively. He has over 12 years of industrial and academic experience. He is currently a Faculty Member and an Assistant Professor with the Electrical Engineering Department, The University of Lahore, Islamabad, Pakistan. His research interests include computer architecture, embedded system, system on-chip, and network-on-chip.



MUHAMMAD KHALIL AFZAL (SM'16) received the B.S. and M.S. degrees in computer science from the COMSATS Institute of Information Technology, Wah Cantonment, Pakistan, in 2004 and 2007, respectively, and the Ph.D. degree from the Department of Information and Communication Engineering, Yeungnam University, South Korea, in 2014. From 2008 to 2009, he served as a Lecturer with Bahauddin Zakariya University, Multan, Pakistan. From 2009 to 2011, he was a Lecturer with King Khalid University, Abha, Saudi Arabia. He is currently an Assistant Professor with the Department of Computer Science, COMSATS Institute of Information Technology at Wah Cantonment. His research interests include wireless sensor networks, ad hoc networks, smart cities, and IoT. Prof. Afzal is serving as a Reviewer for the IEEE ACCESS, *Computers and Electrical Engineering* (Elsevier), the *Journal of Network and Computer Applications* (Elsevier), *Future Generation Computer Systems*, and the IEEE TRANSACTION ON VEHICULAR TECHNOLOGY, and a Guest Editor for *Future Generation Computer Systems* (Elsevier), the IEEE ACCESS, and the *Journal of Ambient Intelligence and Humanized Computing* (Springer).



TARIQ UMER (M'16–SM'16) received the Ph.D. degree in communication systems from the School of Computing and Communications, Lancaster University, U.K., in 2012, and the master's degree in computer science from Bahauddin Zakariya University, Multan, Pakistan, in 1997. He served for IT education sector in Pakistan for over 13 years. Since 2007, he has been an Assistant Professor with the CS Department, COMSATS Institute of Information Technology, Wah Cantonment. His research interests include cognitive radio ad hoc networks, Internet of Things, wireless sensor networks, telecommunication network design, vehicular adhoc networks, and Internet of Vehicles. Prof. Umer is an active member of the Pakistan Computer Society and the Internet Society Pakistan. He is currently serving as an Editorial Board Member for *Future Generation Computer System* (FGCS) (Elsevier) and an Associate Editor for the IEEE ACCESS journal. He served in the TPC for the international conference FIT 15, FIT 16, IEEE PGnet 2011, IEEE PGnet 2012, IEEE FMEC 2016, WPMC 2017, CSCN 2017, and ANTS 2017 conferences. He is currently serving as a Reviewer for the *IEEE Communications Letters*, the IEEE ACCESS, *Computers and Electrical Engineering* (Elsevier), the *Journal of Network and Computer Applications* (Elsevier), *Wireless Networks* (Springer) journal, and the *Journal of Communications and Networks*. He is also serving as a Guest Editor for FGCS (Elsevier) and the IEEE ACCESS.



FARRUH ISHMANOV received the B.S. degree in information systems from the Tashkent State University of Economics, Uzbekistan, in 2007, the M.S. and Ph.D. degrees from the Department of Information and Communication Engineering, Yeungnam University, South Korea, in 2009 and 2014, respectively. He was with the Multimedia Laboratory, Tashkent State University of Economics, during the undergraduate years. In 2015, he joined the Department of Electronics and Communication Engineering, Kwangwoon University, South Korea, where he is currently an Assistant Professor. His research interests include resource management, and security in wireless sensor networks and IoT. He received the Korean Government IITA Scholarship for pursuing the M.S. degree.

• • •