KLSAT: An Application Mapping Algorithm Based on Kernighan–Lin Partition and Simulated Annealing for a Specific WK-Recursive NoC Architecture

Xiaojun Wang, Feng Shi, Hong Zhang

▶ To cite this version:

HAL Id: hal-03770573
https://inria.hal.science/hal-03770573

Submitted on 6 Sep 2022

KLSAT: An Application Mapping Algorithm Based on KernighanLin Partition and Simulated Annealing for a Specific WK-recursive NoC Architecture

XiaoJun Wang[1,2][0000−0002−1665−4629], Feng Shi[1][0000−0001−8870−5408], and Hong Zhang[2][0000−0002−8129−7728]

[1] Beijing Institute of Technology, Beijing 100081, China
[2] Henan University of Economics and Law, Zhengzhou Henan 450046, China
`wxjred9915@163.com, bitsf@bit.edu.cn, gracezxkl@126.com`

Abstract. Application mapping is a critical phase in NoC design because of the running time, the network latency and the power consumption. In order to reduce these problems of applications running on multicore architecture, we propose a novel application mapping algorithm, called KLSAT mapping algorithm. It is used for the triplet-based architecture (TriBA) topology which is WK-recursive based networks well conform to a modular design due to the properties of regularity and scalability. The KLSAT mapping algorithm exploits the advantage of both the KernighanLin partitioning algorithm and simulated annealing algorithm to reduce the overall power consumption and network latency. Compared to the random mapping algorithm, the experiment results reveal that the solutions generated by the proposed mapping algorithm reduce average power consumption and network latency by 6.4%, 12.2% in mapping 27 cores and 29.5%, 26.7% in mapping 81 cores respectively.

Keywords: WK-Recursive Network · KernighanLin Algorithm · Simulated Annealing Algorithm · Application Mapping · Network-on-Chip.

1 Introduction and Motivation

On-chip communication plays one of the crucial roles in multicore architecture topology design. Network-on-Chip (NoC) has been proposed to reduce the power consumption and the network latency limitations of bus-based on-chip multicore architecture [1, 2]. There are several factors affecting the NoC performance, such as the network topology, the routing algorithm, application mapping. So the network-on-chip (NoC) topology design is an important factor in the on-chip multicore architecture. Our team proposed the triplet-based multicore architecture (TriBA) on-chip network is a kind of the multicore WK-recursive network[3, 4], which has several advantages such as scalability, regularity, locality and hierarchy.

   Definition 1: Given a WK-recursive NoC topology with N^L ($L \geq 0$) cores (in here N=3), the core's ID number is encoded in the sequence $a_L a_{L-1} a_{L-2} \cdots a_1 a_0$,

where $a_i \in 1, 2, \cdots, N$ ($0 \leq i \leq L-1$) which contains the level number and the core number after partition at $level_i$ and the value of a_i means the position of the level number. The Fig.1 shows TriBA NoC topology as L=1, 2.
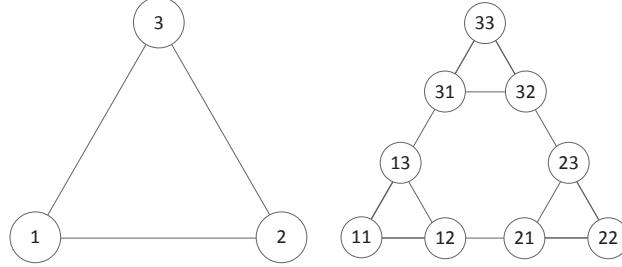

Fig. 1. TriBA multicore network topology with L=1, 2

TriBA network topology has smaller degree, bisection width, smaller network diameter and less number of total links than 2DMesh topology with the same number of cores. It indicates any of TriBA's cores can spend less time to send a data package to other cores. Meanwhile, TriBA has a less total links which means capacity of low power consumption. The researches [5, 6] elaborate that on-chip multicore processors such as the power consumption of Terascale and MIT RAW with respect to the whole on-chip power are 30%, 40% respectively. So another crucial challenge in NoC is how associate the IP cores implementing tasks of an application to reduce the power consumption. This is application mapping algorithm which is a crucial design decision to improve the performance of the overall multicore architecture at an early design phase. In this paper, we propose a mapping heuristic algorithm (KLSAT mapping algo-rithm) that is based on KernighanLin (KL) algorithm, simulated annealing (SA) algorithm and the WK-recursive multicore architecture TriBA to reduce the overall power consumption and network latency. KL algorithm can reduce the fact network communication cost by placing frequently communicating cores closely. SA algorithm is a kind of mapping algorithm for exploiting optimization and searching solutions that originates from the annealing in engineering. TriBA [7] (Triplet-based architecture) is a novelty WK-recursive on-chip multicore architecture with the characteristic of scalability and locality.

2   Related Work

Several previous works have been proposed to use specially designed application mapping algorithms, for example Kernighan-Lin partitioning algorithm and simulated annealing algorithm, to improve the different NoC architectures performance or reduce power consumption and network latency.

Sahu et al. proposed several mapping algorithms which extends the basic Ker-nighan-Lin bi-partitioning algorithm to enhance the static and dynamic performances of three different NoC architectures [8]. Authors explored the opportunities in optimizing application mapping based on Kernighan-Lin algorithms for express channel-based on-chip network [9]. Manna et al. presented a KL bi-partitioning based approach to perform mapping the core graph of an application onto 2DMesh-based NoC architecture[10]. In [11], the authors proposed an application mapping algorithm for the mesh-of-tree network topology. Authors represented core mapping procedure based on the Kernighan-Lin graph bi-partitioning algorithm to select Through-Silicon-Via positions [12].

However, the KL mapping algorithm has its limitations and the resulted mappings generated by the KL algorithm may not be global optimal solution. It differs from KL algorithm, the SA algorithm has been observed to perform better application mappings. SA is one heuristic algorithm that has been used in a set of previous works for solving the application mapping problems [13-21]. Compared with KL mapping algorithm, the significant strength of SA is the ability of finding the global optimum solution.

Hu and Marculescu first used the SA algorithm in application mapping problem to evaluate the Branch and Bound application mapping algorithm on 2DMesh NoC [13]. In [14], the authors proposed algorithms using the simulated annealing and tabu search with communication-weighted model for obtaining low energy. The authors proposed an application mapping technique based on particle swarm optimization combined with simulated annealing for comparison of the performance of Zmesh with that of other NoC topologies [15]. In [16], the authors proposed two heuristics mapping algorithm based on the simulated annealing method for solving the capacitated version of the location-routing problem. The authors [17] used SA algorithm with two functions to map application onto multiprocessor system-on-chip (MPSoC). Bo et al. [18] proposed SA algorithm by using the Nelder-Mead simplex method for selecting a set of parameters applied. Tosun et al. [19] presented a mapping algorithm based on simulated annealing for energy- and communication-aware mapping problems of mesh-based NoC architecture. In [20], the authors proposed a heuristic algorithm CHMAP to solve the application mapping problem on the mesh topology to reduce energy consumption. In [21], an optimized mapping algorithm based on simulated annealing, which allocates tasks that have big communication volume to adjacent places on the mesh, was proposed for reducing the energy consumption of applications running on multicore architecture.

Based on the above mentioned reasons, the novelty of our proposed KLSAT mapping algorithm employs the advantages of KL algorithm and SA algorithm for mapping application onto TriBA multicore architecture. Firstly, we use with a Kernighan-Lin tri-partitioning algorithm which idea is come from the reference [22]. The modified Kernighan-Lin tri-partitioning algorithm which fits for the triplet-based characteristic of TriBA ensures the communication value among cores in the same partition is maximum value and the communication value among cores between partitions is minimum value. Secondly, we employ a

SA algorithm to find the final optimal mapping. To the best of our knowledge, the KLSAT mapping algorithm is the first work that employs the modified KL algorithm and SA algorithm onto TriBA, which satisfies the performance requirement of the application mapping and minimizes the average power consumption and network latency. Our experimental results show that, compared to the random mapping algorithm, the KLSAT mapping algorithm reduces the average power consumption and network latency by 6.4%, 12.2% in mapping 27 cores and 29.5%, 26.7% in mapping 81 cores respectively.

3   PROBLEM FORMULATIONS

In this section we focus on minimizing the power consumption associated to the application mapping.

3.1   Power consumption model

Ye et al. [23] proposed a power consumption model for evaluating the power consumption of switch fabrics in network routers. For the on-chip multicore architecture, however, links between nodes should also be included in the power consumption model. So Hu and Marculescu [24] proposed a modified power consumption model for the on-chip multicore architecture. By evaluating the difference of the power consumption of various components on-chip multicore architecture, Hu and Marculescu found that the power consumed by buffering and internal wires is negligible compared with switch and link. Thus, the power consumption model can be reduced to:

$$E_{bit} = E_{Sbit} + E_{Lbit} \tag{1}$$

where E_{Sbit} and E_{Lbit} represent the energy consumed by switch and link respectively. So, the power consumption of sending one bit from node i to node j can be ex-pressed as following:

$$E_{bit}^{i,j} = n_{hops} \times E_{Sbit} + (n_{hops} - 1) \times E_{Lbit} \tag{2}$$

where n_{hops} is the number of routers the bit passes on its way along a path from node i to node j.

So the total power consumption of the NoC is the sum of weight value of all edges as following:

$$E_{total} = \sum_{i,j}^{E} \sum_{bit} E_{bit}^{i,j} \tag{3}$$

3.2   Defination of application mapping

The goal of application mapping algorithms is to assign a given task to a specific core in the NoC to match the certain requirement such as minimizing the network latency and power consumption.

Definition 2: The task core graph is a weighted edge graph, C(V, E). A vertex $v_i \in$ V represents a task and the weighted edge $e_{i,j} \in$ E represents the communication bandwidth between the cores v_i and v_j. $\text{Comm}_{i,j}$ denotes the weighted value of edge $e_{i,j}$, which indicates the bandwidth constraints of the communication from vertex v_i to vertex v_j.

Definition 3: The NoC topology graph is a multicore interconnects architecture graph T(U, F). A vertex $u_i \in$ U represents a node in multicore NoC topology and the directed edge $f_{i,j} \in$ F indicates a physical link for directed communicating between the vertices u_i and u_j. $\text{Bw}_{i,j}$ denotes the weighted value of the edge $f_{i,j}$, which shows the available communication bandwidth across the edge $f_{i,j}$.

The application mapping algorithm can be formulated as the following one-to-one mapping function:

Mapping algorithm: given a task core graph C(V, E) and the NoC topology graph T(U, F), find the function:

map: $V \rightarrow U$, such that, map(v_i)=u_j, $\forall\ v_i \in$ V, $\exists\ u_j \in$ U, $\mid$ V $\mid\ \leq\ \mid$ U $\mid$

$\forall\ v_i \in$ V, map(v_i)=U

$\forall\ v_i \neq v_j$, map(v_i)$\neq$map(v_j)

Number(V)$\leq$Number(U)

Minimam(E_{total}).

4   THE PROPOSED KLSAT MAPPING ALGORITHM

In this section, we present the proposed KLSAT mapping algorithm which includes the Kernighan-Lin partitioning algorithm and simulated annealing algorithm to minimize the overall communication cost among all of cores. The goal of KL partitioning algorithm is to partition a task graph into subsets recursively and get the minimum value of the communication costs between the subsets. So we use the KL partitioning algorithm to obtain the first stage optimal solution as the initial solution as the input of the next stage SA algorithm. The simulated annealing algorithm is an effective global optimization algorithm which simulates the physical annealing process of solid and solves large scale combinatorial optimization problems. Along with the Metropolis acceptance criterion is introduced to the optimization process, the result of the simulated annealing achieves an approximate global optimal solution. So, we apply the simulated annealing algorithm and obtain the optimal mapping solutions at the second stage.

The KL partitioning algorithm is applied to recursively partition the core graph. Firstly, all cores are in one partition group at level-0. At level-1, there are three partition subsets, naming partition number 1, 2 and 3, each partition containing one third the nodes of the core graph. At level-2, nine partitions are generated (three each from partition-1, partition-2 and partition-3 of level-1) having partition number 11, 12, 13, 21, 22, 23, 31, 32 and 33. This continues until there are 3 cores left in each partition for TriBA. Because the initial partitioning determines the KL algorithm partitioning results, in this paper, this algorithm runs several times for the best result with different randomly generated initial partitions which is used for subsequent mapping and iterative improvement. Fig.

2 shows an example with N = 27 and how the IP-sets are merged. By merging three IP-sets, it finds the best contact between boundaries.
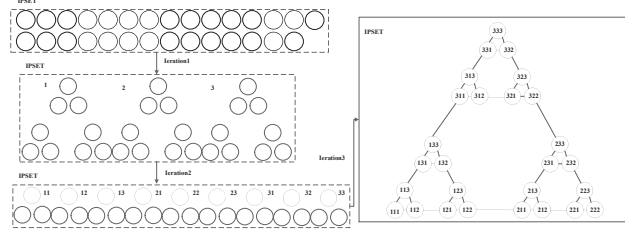

Fig. 2. An example of trinomial merging iteration (N = 27)

Algorithm 1 KL _Tri-Partitioning(C)

Input: Core graph C=(V, E)
Output: Partition number of each core at each level of partitioning
  if |V|≤3 **then**
    return
  end if
  best _tri-partiton = NULL
  best _cost = ∞
  for $i = 0$ to L **do**
    tri-partition = KL_Tri(C)
    if cost(tri-partition$l$ best_cost **then**
      best_cost = cost (tri-partiton)
      best_tri-partiton = tri-partition
    end if
  end for
  Generate graphs C1, C2 and C3 based on best_tri-partition
  KL _ Tri-Partitioning(C1)
  KL _ Tri-Partitioning(C2)
  KL _ Tri-Partitioning(C3)

Now the next stage, each of these 3-core subsets is assigned to the appropriate basic unit of the multicore architecture TriBA, L is the level of TriBA and the number of cores is 3^L. Although these 3-core subsets are attached to the nearby basic unit arbitrarily, it is still great opportunity to resolve an optimization solution by the proposed KLSAT mapping algorithm.

KLSAT Mapping Algorithm:

When the temperature initialization of the system is completed, the KLSAT mapping algorithm executes two nested loops. After the external loop with KL

partition algorithm reaching the global minima, the internal loop refines and finds the optimal local solution. The number of external loop iterations is limited to U^2 as suggested in [14]. The internal loop randomly selects two nodes in a L-level subset and swaps them to determine a new solution. Then the algorithm calculates whether the new solution is better than the old solution. If it is, the new solution replaces the current solution. Otherwise, the algorithm automatically generates a random variable γ ($0 \leq \gamma \leq 1$), and compares with the acceptance probability function $(-\Delta P)/\text{Temperature}$. If the value of the function result is higher than γ, the new solution is accepted. The acceptance probability is high at high temperatures. However, with the temperature of the system lowing, the acceptance probability decreases. We limit the iteration of the internal loop to L^2 consecutive rejects and the Temperature is more than 0.01. When each internal loop completed, temperature of the system decreases and the algorithm starts a new loop accepting the new solution as our initial solution for the next iteration.

Algorithm 2 Algorithm KLSAT mapping Task mapping algorithm based on simulated annealing

---

Input: Core Graph C= (V, E), Topology Graph T=(U, F), $U=3^L$, $M=400L^2$
  Partition_ID (partition number) for each core at each level of KL_Tripartition algorithm and simulated annealing algorithm
Output: Addressing number of each core in term of Q(level, C) S = KL_Tripartion(C)
  P=KL_Tri(C) S_best = S P_best = P Temperature = 1000L
  for i=0 to U^2 **do**
      R = 0
      while R<L^2 and Temperature>0.01 **do**
          S'=neighbor(S)
          P'=KL-Tri(S')
          Δ P=PP'
          Generates a random variable γ
          if ΔP$\leq$0 or $\gamma\leq$ e(-ΔP)/Temperature **then**
              P = P'
              R = 0
          else
              R++
          end if
          if R=0 and PlP$_{best}$ **then**
              S$_{best}$ = S
              P$_{best}$ = P
          end if
      end while
      Decrement Temperature
  end for
  Q=MAPPING(C)
  return Q

We produce a mapping by using MAPPING (G) algorithm. At each level of tri-partitioning, we assign a partition number 1, 2 and 3 to each subset by turn. These numbers have been utilized in the address assignment process in the MAPPING (G) algorithm. In the mapping algorithm, these 3-core subsets are assigned according to the output results generated by KLSAT mapping algorithm. After the mapping algorithm completed, each core has an assigned (level number, subset number) to identify its mapping position on the on-chip multicore TriBA.

At last the KLSAT mapping algorithm completed, we obtain the global optimal solution. All of task cores are mapped onto the corresponding position of TriBA multicore architecture.
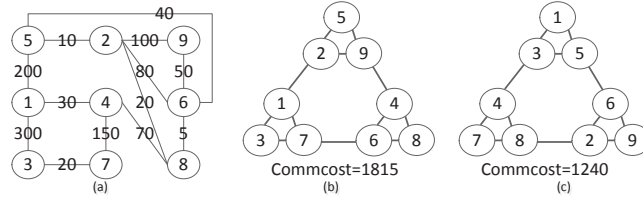

Fig. 3. An example for KLSAT mapping algorithm ((a) an example task graph, (b) communication cost of random mapping, (c) communication cost of KLSAT mapping)

In Fig. 3, we present an example of our KLSAT mapping algorithm. Fig. 3(a) shows an example of task graph with communication weighted between nodes. Fig. 3(b) and Fig. 3(c) shows communication cost with random mapping and KLSAT mapping. The communication cost of mapping with random mapping is calculated as Commcost = 1815. And with KLSAT mapping, the communication cost becomes Commcost = 1240, which is accepted as the new solution. The KLSAT mapping algorithm continues executing the iteration process until the predefined terminated condition value is reached.

5   EXPERIMENTATION AND RESULTS

5.1   Simulator and benchmarks

In this paper, we used Gem5 as our simulator to evaluate the KLSAT mapping algorithm, which is widely used as a configurable architecture simulator for multicore on-chip architecture-related research. In Gem5, the Orion[25] model is used to evaluate the power consumption of the various NoC topologies. Meanwhile, the benchmarks of PARSEC [26] are used in the following experiments. We use the WK-recursive NoC TriBA topology as the NoC topology, which is a regular topology with better NoC topology characteristics such as smaller network diameter, less total links and lower node degree than the 2DMesh topology.

We compare the KLSAT mapping algorithm with several other algorithms on the TriBA NoC architectures: (1) BL_TriBA (the baseline): which maps the tasks onto the TriBA NoC topology randomly; (2) KL_TriBA: KL mapping algorithm on the TriBA structure; (3) SA_TriBA: which is the conventional simulated annealing algorithm on TriBA NoC structure; (4) KLSAT: our proposed mapping algorithm on TriBA NoC structure.

5.2   Results and analysis

Based on the previous research experience, we set the initial parameters of the algorithm as follows: M =4000, temperture$_0$=5000, terminated temperature ε=0.01. We implement the algorithm in Matlab R2013b environment. Host CPU is Intel Core i7 3.40GHz, 8GB memory and operating system is Windows 7. Host has 8 processor cores and the sizes of the target machine are 27 and 81.

The network latency of TriBA multicore architecture normalized to the baseline case shows in Fig. 4 and Fig. 5. Due to the various communications characteristics of these benchmarks, the network latency of experimental results varies significantly. For 27 cores of TriBA, compared to the baseline case, KL_TriBA, SA_TriBA and KLSAT mapping algorithm decrease the network latency by the average 2.9, 6.1 and 12.2% respectively. In the experimental result of TriBA with 81cores, the differences between four mapping algorithms are more significant because the communication overheads among cores are dramatically increased. The KLSAT mapping algorithm decreases the network latency by an average of 26.7% compared to the baseline as shown in Fig. 5.
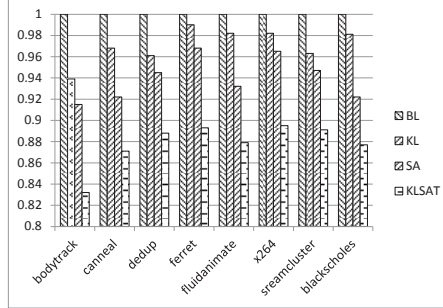

Fig. 4. Network latency of the four algorithms with 27 cores

Fig. 6 shows the power consumption of TriBA with 27 cores. The power consumption is normalized to the BL_TriBA random mapping algorithm. As shown in the Fig. 6, the BL_TriBA random mapping algorithm consumes the highest power consumption while the KLSAT mapping algorithm has the least power, with an average of 6.4% than the random mapping. Fig. 7 shows the experimental results of TriBAs power consumption with 81 cores. For 81 cores
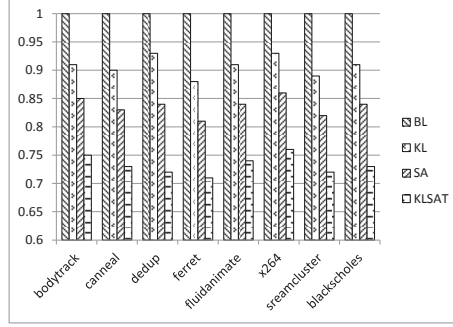
Fig. 5. Network latency of the four algorithms with 81 cores

of TriBA, compared to the baseline case, KL_TriBA, SA_TriBA and KLSAT mapping algorithm decrease the power consumption by the average 12.0, 23.1 and 29.5% respectively. In this experimental result, the power savings of KLSAT mapping algorithm in Fig. 7 is more significant than that in the 27 cores of TriBA architecture in Fig. 6. Overall, KLSAT mapping algorithm saves power consumption by an average of 29.5% compared to the baseline and achieves better performance compared to KL_TriBA and SA_TriBA.
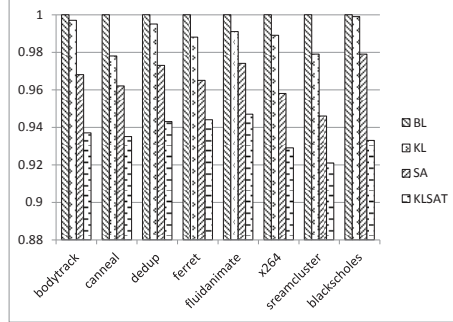

Fig. 6. Power consumption of the four algorithms with 27 cores

The reason is that KLSAT mapping algorithm has a smaller chance to get trapped in local optimum than random mapping algorithm because we add KL partition as the initial solution in the KLSAT mapping algorithm. Because KL partition algorithm combines the triplet-based characteristic of TriBA to make more communication transfer among three cores which have the characteristic of local full interconnect flavor. In consequence, the solution generated by the KL-SAT mapping algorithm has less network communication cost and lower power consumption than the other mapping algorithms.
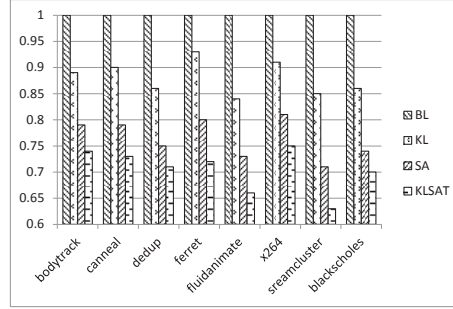
Fig. 7. Power consumption of the four algorithms with 81 cores

In general, the KLSAT mapping algorithm sharply decreases the number of iterations, the power consumption and network latency, compared with the random mapping algorithm. Our proposed KLSAT algorithm achieves better performance than both the KL algorithm and the simulated annealing algorithm.

6   Conclusion

One of the important research fields on NoC is the design of the application mapping algorithms. Several different mapping algorithms have been presented to reduce network latency, lower power consumption, satisfy bandwidth constraint or minimize on-chip area and so on. This paper focused on a new mapping algorithm based on KL partition algorithm and the simulated annealing algorithm in order to generate better performance in application mapping problems. We designed and implemented an application mapping algorithm on multicore architecture TriBA for performance simulation based on KL partition algorithm and simulated annealing, and verified the KLSAT mapping algorithm by experiments. Our experimental results show that the algorithm has significant reduction in the number of iterations, the network latency and the power consumption. It also shows that the algorithm can solve the large-scale problem.

References

1. Dally, W., Towles, B.: Route packets, not wires: On-chip interconnection networks. In: DAC01. 684-689 (Jun. 2001)
2. Benini, L., Micheli,G.:Networks on chips: A new SoC paradigm. IEEE Trans. Computers **35**(1), 70-78 (2002)
3. Farahabady, M., Sarbazi-Azad, H.:The WK-recursive pyramid: An efficient network to-pology. In:PAAN05. 6-11 (Dec. 2005)
4. Wang, Y., Juan, S.: Hamiltonicity of the basic WK-recursive pyramid with and without faulty nodes. J. Theoretical Computer Science **562**(C), 542-556 (2015)
5. Taylor,M, Lee,W., J., et al: Evaluation of the raw microprocessor: An exposed-wire-delay Architecture for ILP and streams. ACM SIGARCH Computer Architecture News **32**(2), 2-13 (2004)

6. Hoskote,Y., Vangal,S., et al:A 5-GHz mesh interconnect for a teraflops processor. IEEE Micro **27**(5), 51-61 (2007)
7. Shi, F., Ji, W., et al:A triplet-based computer architecture supporting parallel object computting. In: IEEE ASSAP07. 192-197 (Jul. 2007)
8. Sahu, P., Manna, N., Shah, N., Chattopadhyay,S.: Extending KernighanLin partitioning heuristic for application mapping onto network-on-chip. J. of Syst. Archi. **60**(7), 562-578 (2014)
9. Zhu, D., Chen, L., Yue, S., Pedram,M.: Application mapping for express channel-based networks-on-chip. In: DATE14. 1-6 (Mar. 2014)
10. Manna, K., Choubey, V., et al:Thermal variance-aware application mapping for mesh based network-on-chip design using Kernighan-Lin partitioning. In: PDGC14. IEEE 274-279 (Dec. 2014)
11. Fang, J., Yu, L., et al: KLGA: an application mapping algorithm for mesh-of-tree (MoT) architecture in network-on-chip design. J. Supercomputing. **71**(11), 4056-4071 (2015)
12. Manna, K., Teja, V., Chattopadhyay, S., et al: TSV Placement and Core Mapping for 3D Mesh Based Network-on-Chip Design Using Extended Kernighan-Lin Partitioning. In: VLSI15. IEEE 392-397 (Jul. 2015)
13. Hu, J., Marculescu, R.: Energy- and performance-aware mapping for regular NoC architectures. IEEE Trans. Computer-Aided Design Integrated Circuits and Systems **24**(4), 551-562 (2005)
14. Marcon, C., Moreno, E., Calazans, L., Moraes, F.: Comparison of network-on-chip mapping algorithms targeting low energy consumption. IET Computers Digit. Tech. **2**(6), 471-482 (2008)
15. Prasad, N., Mukherjee, P., Chattopadhyay, S., et al: Design and evaluation of ZMesh to-pology for on-chip interconnection networks. J. Parall. Distrib. Computing **113**(2), 17-36 (2018)
16. Dong, Z., Yang, B., Hu, P., et al: An efficient global energy optimization approach for robust 3D plane segmentation of point clouds. ISPRS J. Photogra. Remote Sensing **137**(1), 112-133 (2018)
17. Orsila, H., Salminen, E., Timo, D.: Best practices for simulated annealing in multiprocessor task distribution problems. Tech. **4**(2), 197-198 (2008)
18. Yang, Bo., Liang, G., et al: Parameter-optimized simulated annealing for application mapping on networks-on-chip. In: LIO12. 307-322 (2012)
19. Tosun, S., Ozturk, O., Ozkan, E., Ozen, M.: Application mapping algorithms for mesh-based network-on-chip architectures. J. Supercomputing **71**(3), 995-1017 (2015)
20. Cheng, C., Chen, W.: Application mapping onto mesh-based network-on-chip using constructive heuristic algorithms. J. Supercomputing **72**(11), 1-14 (2016)
21. Zhong, L., Sheng, J., et al: An optimized mapping algorithm based on simulated annealing for regular NoC architecture. In: ASIC11. IEEE 389-392 (Oct. 2011)
22. Larsson, T., Jesper, F.: Direct graph k-partitioning with a Kernighan-Lin like heuristic. Operations Research Letters **34**(6), 621-629 (2006)
23. Ye, T., Benini, L., Micheli, G.: Analysis of power consumption on switch fabrics in network routers. In: DAC02. 524-529 (Jun. 2002)
24. Hu, J., Marculescu, R.: Energy-aware mapping for tile-based NoC architectures under performance constraints. In: ASPDAC03. IEEE 233-239 (2003)
25. Kahng, A., Li, B., Peh, L., Samadi, K.: ORION 2.0: A power-area simulator for interconnection networks. IEEE Transactions on Very Large Scale Integration Systems **20**(1), 191-196 (2012)

26. Bienia, C., Li,K.: PARSEC2.0: A new benchmark suite for chip-multiprocessors. In: AWMBS09. IEEE 1-9 (2009)