Network-on-Chip Application Mapping based on Domain Knowledge Genetic Algorithm

Yin Zhen Tei*, N. Shaikh-Husin*, Yuan Wen Hau†, Eko Supriyanto†, M. N. Marsono*
*Department of Electronic and Computer Engineering, Faculty of Electrical Engineering
†IJN-UTM Cardiovascular Engineering Center, Faculty of Biosciences and Medical Engineering
Universiti Teknologi Malaysia
81310 Skudai, Johor, Malaysia
Email: yztei2@live.utm.my, nasirsh@fke.utm.my, {hauyuanwen, eko}@biomedical.utm.my, nadzir@fke.utm.my

Abstract—This paper addresses application mapping technique targeted for large-scale Network-on-chip (NoC). The increasing number of intellectual property (IP) cores in multi-processor System-on-Chips (MPSoCs) makes NoC application mapping more challenging to find optimum core-to-topology mapping. The factorial increase in possible mappings space requires a mapping algorithm to efficiently look for potential mapping space. This paper proposes an application mapping technique that incorporates domain knowledge into genetic algorithm (GA) to minimize the energy consumption of NoC communication. The GA is initialized with knowledge on network partition whereas the genetic crossover operator is guided with communication demands. The effects of domain knowledge GA on initial population and genetic operator are analysed in terms of the solution quality and convergence speed. Through simulation results on the MCSL real application traffic traces, the knowledge-based genetic operator gives 28% energy saving compared to application mapping using conventional GA. By combining both domain knowledge technique into GA, the solution quality can further save energy by 5% and convergence speed by 86% compared to knowledge-based crossover GA. Our experiment suggests that domain knowledge in GA initialization and crossover operation obtain low energy application mapping effectively for large-scale NoC-based MPSoC.

Index Terms—Application mapping, domain knowledge, genetic algorithm, Network-on-Chip, network partitioning

I. INTRODUCTION

Network-on-chip (NoC) has emerged as a promising on-chip communication architecture providing modularity and scalability for MPSoCs. Application mapping determines the placement of IP cores to the routers on the Network-on-Chip (NoC) tiles such that the performance or cost metrics of interest are optimized [1]. It is an NP-hard problem that exhaustive algorithm cannot be applied. Therefore, an effective mapping algorithm to reduce the large search space and to obtain optimum mapping is required. Optimization search with refinement such as simulated annealing (SA) [2] and genetic algorithm (GA) [3] have been used in application mapping in NoC. GA could provide near optimum solution with limited known information [4] but it relies on the initial population, genetic operators and selection mechanism. Domain-knowledge has been used in crossover and mutation to improve GA mapping and convergence [3] by checking each gene’s communicating distances among cores. These will cause computation time to increase drastically for highly communicating applications and large-scale NoCs.

Large-scale MPSoCs are mostly combinations of few subsystems. One IP core may only communicate with several cores in such a large system. Network partitioning (NP) decomposes a large system into several smaller subsystems in which highly communicating cores are grouped in the same partition. Based on this knowledge, NP may be done prior to the application mapping to narrow down the search space and to reduce the core mapping complexity. A good initial mapping can be loosely defined as one that increases the probability to reach near optimum mapping [5]. It is important to investigate the effect of domain knowledge GA for NoC application mapping in terms of final solution quality and convergence speed especially for large-scale NoCs.

This paper proposes an application mapping technique that incorporates domain knowledge into genetic algorithm (NP-DKGA) to minimize the energy consumption of NoC communication. The initial GA population is initialized with network partitioning knowledge whereas the genetic operator crossover is guided with the knowledge on cores communication demands. NP-DKGA application mapping technique operates in two phases: a k-way partitioning of application graph into assigned partitions as initial population; and using knowledge-based genetic algorithm (DKGA) to search for near optimum mapping. The authors have tested the effectiveness of NP-DKGA on several real benchmarks applications [6]. Based on our simulation results, our proposed VOPD core mapping obtained the best mapping identical to reference [7]. For large applications, our technique show an overall improvement in the final mapping quality and in term of convergence. All results are compared with the mapping using conventional genetic algorithm (CGA).

II. RELATED WORK

Many application mapping techniques have been proposed for optimizing different costs and performance metrics. The first mapping algorithm based on a modified bit energy model [8] was proposed by Hu and Marculescu using branch and bound technique such that energy consumption can be minimized with bandwidth reservation [9]. Reference [5] compares few application mapping algorithms using the bit energy...
model which is targeted for low energy systems. Simulated annealing (SA) [2] and genetic algorithm (GA) [3] were also proposed as application mapping techniques to optimize energy consumption using bit energy model.

Multi-objective evolutionary algorithm with random-based initial population mapping and crossover with hot spot remap was proposed to optimize execution time and power consumption [10]. The author claimed that more effective genetic operators have a great impact on the final mapping [10]. In reference [11], GA is initialized with a random initial mapping and the crossover is based on swapping communicating cores with neighbouring cores. There are few crossover techniques such as remap hotspot [10], [12] and cycle crossover [13]. All of these crossover techniques do not include useful NoC mapping knowledge. The convergence is slow especially for large-scale NoC. In the domain-knowledge evolutionary algorithm [3], mapping similarity crossover (MS) has been proposed that maintains the common characteristic between parents and the rest using greedy mapping. Mapping similarity is able to handle symmetric problem in mesh topology but the technique increases the computation time as the NoC size increases.

As the NoC size increases, the complexity of communication ranking placement can hardly obtain good mappings. Large SoC system can be divided into several clusters (partitions). Cluster-based application mapping techniques have been proposed in [14], [15]. The author in [14] proposed a cluster-based relaxation for integer linear programming formulation for application mapping in order to reach the optimum result within tolerable time limits. Authors in [13] proposed a partition-based application mapping with near-convex core placement for large NoC. However, these three techniques map cores without improving cross partition movement. Although they show shorter runtime, the final mapping quality is affected [13]. A mapping algorithm based on Kernighan-Lin (K-L) partitioning, called LMAP, has been proposed to explore search space via flipping the partitions and cores in a hierarchical fashion [16].

References [15] proposed a cluster-based initial mapping for simulated annealing (CSA) to speed up the convergence to near-optimal solutions. These work shows the runtime advantage without compromising the quality of solution compared to the pure SA approach. Given an random initial mapping, optimized simulated annealing (OSA) [2] improves SA by clustering communicating cores implicitly during swapping process. OSA shows better mapping quality compared to CSA. However, author in [3] has shown that evolutionary algorithm perform better than OSA. Particle Swarm Optimization (PSO) has been proposed with deterministic initial mapping to explore the search space [17]. The domain knowledge applied on initial mapping using greedy-based approach where IP cores are placed on the NoC topology based on the descending ranking of total communication cost defined in APCG. This initial mapping technique can hardly obtain good mapping as the NoC size increases.

### III. Domain-Knowledge Genetic Algorithm Application Mapping

This paper proposes an application mapping technique that incorporates domain knowledge into genetic algorithm (NP-DKGA) to minimize the energy consumption of NoC communication. The initial population of GA is initialized with network partitioning (NP) knowledge whereas the genetic operator crossover is guided with communication demands knowledge as shown in Fig. 1. The proposed two-phase application mapping is targeted for low power large-scale NoC. The first phase is shown in dashed-box in Fig. 1. All cores are mapped onto the assigned partitions on the mesh-based topology as the knowledge-based initial population. The second phase involves the optimization of energy consumption using domain-knowledge crossover genetic algorithm (DKGA) to search for optimum mapping. Some definitions used in this paper are listed next.

![Fig. 1: Overview of the proposed technique, NP-DKGA.](image)

### A. Problem Formulation

**Definition 1:** An application characteristic graph (APCG), $G = G(V,E)$ is a directed graph, where each vertex $v_i \in V$ represents an IP core and each directed edge $e_{(i,j)} \in E$ characterizes the total communication volume in bits from vertex $v_i$ to vertex $v_j$. Application tasks are assumed to be assigned to all vertices, $v_i$ and scheduled to each IP core.

**Definition 2:** NoC mesh-based network, $T(R, Ch)$ is a labelled graph, where each $r_i \in R$ denotes a router and each $Ch_i \in Ch$ denotes a channel. All routers can have a maximum of 5 ports with 4 ports connected to neighbouring routers via channels and one connection to the processing core. $T$ is placed on a grid in the XY plane with unit distances between adjacent routers. $x(r_i)$ and $y(r_i)$ denote the x and y coordinates for a router $r_i \in R$.

**Definition 3:** Given an input APCG, network partitioning decomposes APCG into smaller subsystems according to size the mesh-based topology. APCG will be partitioned or divided into $m$ partitions, $P_1, P_2, ..., P_m$. Network partitioning is to find $P(N, \lambda)$ where $N$ is number of cores in each partition and $\lambda$ is inter-partition traffic. The objective of network partitioning is to reduce inter-partition traffic (min-cut partitioning), $\lambda$ subject to constraints, $Const(V)$ to obtain a balanced number of cores for all partitions.

**Definition 4:** The placement for the partitioned APCG involves partition placement and core placement. Assume a partitioned graph $P(N, \lambda)$ and topology $T(R, Ch)$. Partition placement, $\omega : P \rightarrow T$ assigns certain regions on the mesh-based topology, $T$ to a particular partition, $P_i$. For core
Network partitioning as knowledge-based GA initial mapping

Network partitioning (NP) decomposes a large NoC system into a few smaller partitions. In this proposed NoC application mapping, NP is implemented in two stages: mesh topology partitioning and application partitioning. In the first stage, mesh topology is assigned into a few smaller regions where each region represents one partition. The partitioning level will depend on the size of the topology. For the cases where mesh topology cannot be bipartitioned, such as 3 × 3 and 5 × 5, k-way partitioning can be implemented. Mesh topology is partitioned into k partitions with the same number of tiles for each partition.

In the second stage, the multilevel-KL (Kernighan-Lin) algorithm decomposes IP cores in APCG into halves and refines the partitions at each subsequent level. This algorithm is chosen due to its high-quality partitions and is scalable for large network [18]. The application is partitioned according to number of tiles available in each mesh partition. Each partition must have at least four available tiles. If the partition size is too small, the role of NP to group the highly communicating cores will be insignificant. The objective of NP is to achieve min-cut with the lowest inter-partition traffic. There is a single constraint to core balance each partition. Fig. 2 shows an example of 2-level partitioning on 4 × 4 mesh-topology and the VOPD application [19]. The dashed lines show the first-level partitioning while dashed-dot lines show the second-level partitioning for the VOPD application.

![Network partitioning diagram](image-url)

Fig. 2: VOPD application and 4 × 4 mesh-based topology partitioning. Partition and core mapping of VOPD application onto NP knowledge-based initial population and the associated integer chromosomes.

The outcome of the two-stage NP is used to generate an initial population for GA in the second phase. Thus, instead of detail hierarchical mapping for all partitions and cores, core placement are done randomly within the assigned region of mesh topology. The min-cut partitioning technique groups heavily communicating IP cores closer to each other to reduce communication energy consumption. It results in a better initial mapping compared to random-based mapping and increases the probability for GA to converge and to reach near optimum solution.

C. Knowledge-based genetic algorithm

Instead of utilizing conventional genetic algorithm (CGA), we propose a domain-knowledge genetic algorithm that applies NP as initial population and heuristic crossover (NP-DKGA). The detail of each GA components are presented next.

1) Problem Representation: Permutation chromosome is used to represent the application mapping problem. It consists of a series of genes with each gene corresponds to a tile in the mesh topology. Each gene is assigned an integer which represents an IP core in $APCG$ that is attached to the corresponding router in each tile. Fig. 2 shows examples of integer chromosome for a 4 × 4 mesh topology. A gene associated to a router is assigned null value if no IP core is assigned to the router. A valid permutation chromosome cannot have two genes with the same integer because it would represent a core connected to two routers.

2) Population: The population is the main element of a genetic algorithm. Research has shown that the initial population may have effect on the best fitness function value and these effects may last for several generations [4]. For a large NoC, the possible mapping space is extremely large which could slow down the convergence. Hence, a good initial population may result in faster convergence. On the other hand, the population size influence is also another parameter to decide the coverage of mapping space. A population that is too large takes time to evolve whereas a population that is too small will lead to a local minima. Population size is proportional to the application size.

In this paper, the NP initial mapping that groups communicating cores within the same partition provides a potential low energy mapping. This enables GA to explore the reduced search space for low energy mapping with faster convergence instead of random exploration of the huge search space. Fig. 2 show NP initial population for 4 × 4 VOPD application after random island and core mapping.

3) Fitness Function: Fitness function represents the desired optimization goal. For NoC application mapping, the fitness function is closely related to the distance between the source and destination cores. According to literature, energy optimization is the primary goal for NoC. This paper targets energy minimization where bit energy model is utilized. Ye at al. [8] proposed a model to calculate power consumption in a switch fabric that accounts for each data bit goes through the network router. Then, Hu and Marculescu [9] modified the bit energy model such that it is suited to mesh-based network. Based on these previous works, this paper applies the bit energy model to optimize, $E_{bit}^{VOPD}$, the required energy for a bit of data from source core to destination core.

$$E_{bit}^{VOPD} = n_{hops}E_{bit} + (n_{hops} + 1)E_{Rbit} \tag{1}$$

where $n_{hops}$ is the number of hops for a path from the source core to the destination core (i.e., one hop is the distance
between two adjacent routers) with XY deterministic routing. \( E_{\text{Ltot}} \) is the energy consumption for a link between adjacent router and \( E_{\text{Rtot}} \) is the energy consumption for the router. The \( E_{\text{Rtot}} \) and \( E_{\text{Ltot}} \) are set to 0.43pJ and 5.445pJ respectively according to [8]. The overall energy consumption \( E_A \) is the summation of all energy bit consumed by all bit transmissions.

\[
E_A = \sum_{s,d} (E_{\text{tot}}^{s,d} \times e_{s,d})
\]

where \( e_{s,d} \) is the total communication traffic in bit from the source core to the destination core.

4) \textbf{Crossover}: Crossover is used to produce offspring, and fitter chromosomes are searched to form a new population. In this paper, knowledge-based is proposed as described in Algorithm 1. Crossover point are randomly set according to the nature randomization behaviour of GA. Two children chromosome are generated from two selected parents. After the crossover between parents, if the same integer is assigned to two genes, the latter gene in the resulting chromosome is labelled as \textit{InvalidGene}. Cores that are not assigned to any gene are labelled as \textit{UnmappedCores}. In CGA, all \textit{InvalidGenes} are randomly remapped with \textit{UnmappedCores}. However, in the proposed DKGA, we applied a heuristic crossover technique. The \textit{UnmappedCores} will determine its communication with the adjacent router of \textit{InvalidGene}. The \textit{UnmappedCores} will be remapped to \textit{InvalidGene} which has the highest communication with \textit{NeighborCore}. This crossover algorithm is done iteratively until the generated children chromosomes reach the population size. This implicit clustering approach aids GA to explore the mapping space efficiently for low power mapping.

\begin{algorithm}
\textbf{Algorithm 1} Knowledge-based Crossover Algorithm
\begin{algorithmic}
\STATE \textit{Population} is the population size
\STATE \textit{TotalParent} is total parent chromosomes
\STATE \( B \) is the length of chromosome
\FOR {\textit{i} = \text{TotalParent} + 1 \text{ to } \text{Population}}
\STATE Select parent chromosome using roulette wheel, \( P1 \) and \( P2 \).
\STATE Select random crossover point, \( C \in B \).
\STATE \( \text{Child}(i) \leftarrow \text{Crossover between } P1 \text{ and } P2 \).
\STATE Check \textit{InvalidGene}.
\STATE Check \textit{UnmappedCores}.
\STATE \textit{NeighborCore} = \text{GetAdjacentCore}(\textit{InvalidGene})
\STATE \textit{CommunicatingCore} = \text{GetCommCore}(\textit{NeighborCore}, \textit{UnmappedCores})
\STATE \textit{InvalidGene} \leftarrow \text{max}(\textit{CommunicatingCore})
\ENDFOR
\end{algorithmic}
\end{algorithm}

5) \textbf{Mutation}: Mutation is the operation that uses only one parent and creates one child by applying some kind of randomized change to the chromosome [20]. Probability of mutation is the probability for each gene in every chromosome to undergo mutation [21]. Since NoC application mapping consists of a large size chromosome, checking the probability for each gene is time consuming. Thus, population-based probability mutation is applied which refers to probability of one chromosome to undergo mutation in a population. For the selected chromosome, only two genes are randomly selected and mutated. Otherwise, the gene remains in the chromosome.

\section{Simulation Methodology and Result Discussion}

This section discusses the benchmarks, parameter setting and simulation results for NP-DKGA. A realistic traffic benchmark suite MCSL [6] that supports several NoC architectures is used to generate real traffic traces in this experiment. Six real applications were included in MSCL: FFT, FPPPP, SPARSE, ROBOT, RSenc and RSdec. A 12 × 12 mesh-based architecture is chosen for assessing the scalability of the proposed algorithm and additionally, we also implement VOPD (video object plane decoder) [19] for 4 × 4 mesh-based network. This application mapping is evaluated on mesh-based NoC and XY deterministic routing. All tasks in each applications are scheduled and mapped into the IP cores available on meshed network. For all the benchmarks, network partitioning is implemented using Chaco [18] tool before the application mapping stage. Chaco performs bisection partitioning by grouping highly communicating cores in the same partition and at the same time, performs the min-cut operation.

This paper studies the effectiveness knowledge-based crossover and effectiveness network partitioning as initial mapping for GA optimization in term of the solution quality and convergence. Several parameters in GA are fixed with probability for crossover of 0.8, population-based mutation rate of 0.3, and population size of 100 for 12 × 12 network size and 50 for the 4 × 4 network. This work does not analyse the optimal parameters for DKGA but rather to assess the effectiveness of the knowledge-based in initial population and genetic operator. The termination of GA is set to 1000 generations for MCSL applications and 300 generations for VOPD application. In order to obtain highly accurate results for GA with random-based optimization method, statistical analysis based on different energy level; the highest, the lowest and the average energy consumption for each application benchmarks.

\begin{table}[h]
\centering
\begin{tabular}{|c|c|}
\hline
\textbf{Benchmark} & \textbf{Range of connectivity degree} \\
\hline
RSenc & 0-14 \\
ROBOT & 0-15 \\
FFT & 60-116 \\
RSdec & 0-43 \\
SPARSE & 0-9 \\
FPPPP & 0-80 \\
VOPD & 1-4 \\
\hline
\end{tabular}
\caption{Connectivity degree for all benchmarks.}
\end{table}

TABLE I shows the connectivity degree for all benchmarks used in the experiments. The connectivity degree is defined as the total incoming and outgoing communication for each core in the benchmark. The FFT has a high connectivity degree between 60 and 116. On the other hand, other benchmarks contain IP cores that are not communicating to other cores. The relationship between the connectivity degree with the GA convergence will be analysed next.

We analysed the convergence speed of GA with knowledge-based crossover and NP initial mapping for all benchmarks.
TABLE II: The convergence speed improvement for R-DKGA and NP-DKGA compared to conventional GA (CGA).

<table>
<thead>
<tr>
<th>Benchmarks</th>
<th>Convergence speed improvement (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>RSenc</td>
<td>R-CCA 65% R-DKGA 62%</td>
</tr>
<tr>
<td>ROBOT</td>
<td>R-CCA 56% R-DKGA 65%</td>
</tr>
<tr>
<td>FFT</td>
<td>R-CCA 14% R-DKGA 67%</td>
</tr>
<tr>
<td>RSdec</td>
<td>R-CCA 5% R-DKGA 86%</td>
</tr>
<tr>
<td>SPARSE</td>
<td>R-CCA 33% R-DKGA 44%</td>
</tr>
<tr>
<td>FPPPP</td>
<td>R-CCA 38% R-DKGA 47%</td>
</tr>
<tr>
<td>VOPD</td>
<td>R-CCA 69% R-DKGA 55%</td>
</tr>
</tbody>
</table>

TABLE II shows the percentage of convergence speed improvement using knowledge-based crossover and NP knowledge-based initial mapping with the speed convergence defined in (3). The knowledge-based crossover (R-DKGA) shows improvements in all benchmark applications. FFT is the largest communicating application among all benchmarks. For highly communicating applications: FFT, FPPPP and RSdec, knowledge domain on initial mapping helps DKGA converges faster. These applications show that advanced NP initial mapping helps highly communicating applications to converge faster and assists the GA in obtaining high quality mapping. The improvement in term of quality mapping solution will be discussed in detail next.

TABLE III shows the percentage of energy minimization after several simulation runs for the domain knowledge crossover GA (R-DKGA) and the domain knowledge in initial mapping (NP-DKGA) compared to conventional GA (CGA). It is easily observed that all applications improve significantly with domain knowledge applied GA compared to CGA. With the knowledge-based crossover, energy saved up to 28% for the highest communicating application (FFT). Besides, with the NP knowledge-based initial mapping, all applications show NoC communication energy saving up to 29%. TABLE III shows that the NP initial population assists GA to obtain better mapping quality. The NP-DKGA gives better energy saving compared to R-DKGA. Furthermore, the VOPD application optimized using R-DKGA and NP-DKGA achieved the global minimum identical to reference [7].

Fig. 3 shows the energy consumption for 10 simulations in comparison to R-CGA, R-DKGA, and NP-DKGA with CGA used as the reference. The error-bar shows the lowest, highest and average energy consumption of all 10 simulation runs for R-CGA, R-DKGA, and NP-DKGA. The result shows that knowledge-based crossover and initial mapping have significantly improved the quality of the final mapping. In term of convergence and quality mapping, NP knowledge-based initial mapping and knowledge-based crossover need to be added into GA in order to obtain better mapping quality and faster convergence speed. The knowledge-based crossover gives better final mapping regardless if the NP initial mapping is applied. The results show that NP could further facilitate DKGA to improve the application mapping quality and speed up the convergence when targeted for low power large-scale NoC.

V. CONCLUSION

This paper presented the NP-DKGA technique that uses network partitioning knowledge as initial mapping and knowledge-based crossover in GA in order to optimize NoC application mapping. This algorithm is targeted for large-scale low energy NoC. We performed analysis on the effectiveness of domain knowledge applied in initial population and genetic operator of GA based on several real benchmarks. The NP knowledge-based initial mapping shows significant energy saving compared to random initial mapping. Moreover, the convergence speed can be improved up to 86%. The effectiveness of the knowledge-based crossover gives significant effect in energy reduction compared to the NP initial mapping GA. For less communicating application, knowledge-based crossover GA (DKGA) could converged well. However, for highly communicating application, our experiment shows that the knowledge-based initial mapping in DKGA can improve both the application mapping quality and speed up the convergence. NoC is meant for heavy communication and large scale MPSoC. Hence, domain knowledge GA in initial population and crossover operation are needed in order to obtain low energy large-scale NoC-based MPSoC.

In a future work, we plan to consider for multi-objective environment. Thermal balanced in an issue to reduce faults in NoC and increase reliability of NoC. For energy and thermal balanced, network partitioning need to be done with balanced load and min-cut. This work can be extended into more accurate evaluation using cycle accurate NoC simulator.

REFERENCES

Fig. 3: Normalized energy consumption for all benchmarks. Y-errorbar shows minimum, average and maximum energy consumption.