International Journal of Computer and Electrical Engineering, Vol. 3, No. 2, April, 2011
1793-8163

Network-on-Chip: Power Optimization Architecture Mapping based on Global Interconnection Links

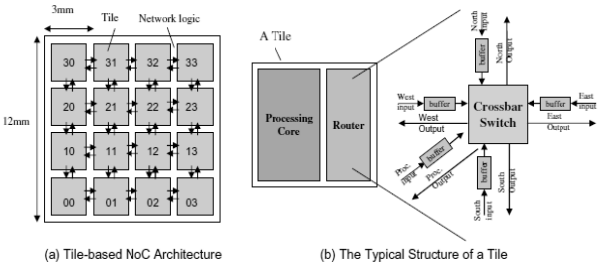
El Sayed M. Saad, Medhat H. Awadalla, Sameh A. Salem, and Ahmed M. Mostafa

***Abstract*—The increasing in communication demands of processing elements (PEs) on Systems on Chips (SoCs) necessitates the existence of Networks-on-chip (NoC) to interconnect the PEs. An important issue in the developing of NoCs is the mapping of the PEs onto the most suitable topology for a specific application. This paper analyzes the system power consumption of the global interconnection links in Networks-on-Chip (NoC)-based systems. Based on this analysis, this paper also proposes a new algorithm to optimize the power consumption on the global interconnection links of NoC-based systems at early design phases. The robustness and reliability of the proposed algorithm is verified in the context of MPEG4 video application. NoC application synthesis and mapping to nine standard topologies are explored. The achieved experimental results show that the proposed optimization algorithm achieve the optimum mapping for all nine topologies in large number of tasks with much less time compared with the exhaustive algorithm.**

***Index Terms*—Architecture mapping, Network-on-Chip, Power optimization, Standard topologies.**

I. Introduction

Complexity increasing of embedded systems is caused by the advancements in technology which enable the integration of hundreds of millions of transistors onto a single chip. Therefore the designers of System-on-Chips are able to integrate more processing elements (PEs) (like e.g. processor cores, DSP cores, memories and application specific hardware) on a single chip. However, the scalability of Systems-on-Chips (SoCs) declines with the number of processing elements connected to the on-chip buses. Furthermore, the increasing number of processing elements on SoC is accompanied by growing problems of deep sub micron effects like cross-talk and interference.



(a) Tile-based NoC Architecture (b) The Typical Structure of a Tile

Figure 1 Architectural overview of a NoC-based System

Energy inefficiency is another critical limitation of on-chip buses. In these architectures, every data transfer is broadcast, meaning that the data must reach each possible receiver at great energy cost. The recent integrated systems will contain tens to hundreds of units generating information that must be transferred. For such systems, a bus-based network would become a critical performance and power bottleneck [1], [2].

Networks-on-chip (NoC) are emerged as new communication paradigm for high-performance systems-on-chip with high information volumes to accommodate [1]. PEs communicate with one another by sending packets via the network instead of routing wires. The router embedded in each tile consists of input and output links, buffers and a crossbar switch. Architectural overview of a NoC-based System [3] is shown in Fig. 1.

However, on-chip systems have to balance the need for performance with power and area constraints, to be feasible [4]. Optimizing the power consumption of NoC-based designs has become more critical with the use of high speed, complex ICs in mobile and portable applications [5], [6]. Therefore, several approaches and methodologies have been proposed to address the high-power dissipation problem from both circuit and system perspectives [7]–[9].

At the circuit level, clock gating, voltage-islands, and multiple voltage thresholds are examples of the approaches proposed to achieve low-power designs: (1) clock gating reduces dynamic power by restricting clock distribution [10], (2) voltage-islands use adaptive or dynamic voltage scaling to optimize supply voltage at runtime and compensate for process and temperature variations [11], (3) multiple-voltage-threshold designs use high-voltage threshold cells to decrease leakage current, where performance is not critical [12]. At the system level, router design, first-in-first-out (FIFO) buffer resizing, and the mapping of PEs are introduced to address the high-power dissipation problem: (1) router design explores the optimum router design in terms of switching techniques, scheduling algorithms, and routing protocols [13], (2) FIFO resizing focuses on acquiring the optimum buffer size that achieves the lowest power consumption [14], (3) mapping of PEs depends on achieving the best matching between PEs' physical placement and their average communication traffic pattern [15].

One of the most effective approaches to address the high-power dissipation problem is to select the network topology that achieves the lowest power consumption [9]. In this paper, we concentrate on the power consumed on the global interconnection links. We propose a new optimization algorithm to select the network topology and the corresponding PEs mapping that achieves the lowest power consumption in terms of the communication traffic on the global interconnection links.

We analyze the power consumption of the global

interconnection links at the system level. The connectivity matrix is obtained from the traffic distribution graph (TDG) of the application and used in the system level analysis of interconnection links power consumption. We explore NoC application synthesis and mapping to 9 standard topologies. The target standard topologies are: Mesh [16], Torus [17], X Torus [18], Ring [19], Octagon (Oct) [20], Binary tree (BT), Butterfly Fat Tree (BFT) [21], Hypercube (Hcube), and Star. Based on this analysis, we develop a new optimization algorithm to find the network topology and the PEs mapping that matches the traffic characteristics of the NoC-based system and gives the lowest power consumption on the global interconnection links. The proposed algorithm reach the optimal solution in a very small time compared to the exhaustive search algorithm.

The rest of this paper is organized as follows. The related work to the theme of the paper is outlined in Section II. Section III presents a brief description of graph theory and power modeling of global interconnection links. The proposed optimization algorithm is described in Section IV along with a brief description of the exhaustive search algorithm. Experiments and discussions are illustrated in Section V. Finally, Section VI concludes the paper.

II. Related Work

Modeling NoC interconnects through abstract models is the first means to approach and understand the required architecture and the impact of the traffic within it [22], [23]. Hu et al. [24] present an energy estimation model based on the traffic flow in the NoC's building blocks (routers and interconnection wires). They make use of the bit energy concept [25], which represents the amount of energy consumed in the transmission of data bits throughout the NoC (in its routers and interconnection wires). This model evaluates the energy consumption in an end-to-end transmission only. Marcon et al. [26], [27] propose an energy estimation model based on the computation and communication dependencies in the cores of a NoC. They make use of the Hu et al. [28] assumption, which states that the energy consumption of an application can be reduced up to 60% by applying different application mapping algorithms in the network. Koohi et al. [29] present a NoC power and performance analysis with different traffic models, using analytical models. The authors targeted a NoC with a mesh topology. The employed traffic models are: (i) uniform, (ii) local, (iii) hot-spot and (iv) matrix transpose. Results were compared to Synopsys Power Compiler and Modelsim, showing an error of 2% for power estimation and 3% for throughput. Elmiligi et al. [9] propose a topology-based methodology that explores the impact of the NoC topology on system power dissipation. It uses a partitioning algorithm that aims to achieve minimum inter-partition traffic. This methodology uses graph-theoretic concepts to acquire the optimum NoC topology that reaches the lowest power dissipation for a given application. Various tools and algorithms have been developed to choose the optimal network topology design and mapping through network partitioning [30], long-range link insertion [31], and exploring various standard topologies [32].

III. Graph Theory and System Power Analysis of Global Interconnection Links

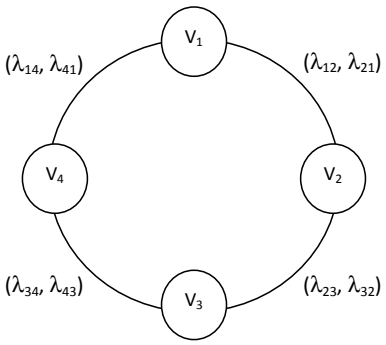
System-level modeling has been used to increase the design productivity of NoC-based systems. In this context, modeling and simulation at high abstraction levels are used to increase and simplify the development and the validation of these systems. In this section, a brief introduction to system analysis using graph-theoretic approach is given. Then, a system analysis of the power consumption on the global interconnection links is presented, in the following subsection.

A. A graph-theoretic approach for application modeling

The topological structure of any interconnection network can be represented by a graph. Then, using graph theory concepts, we could analyze the performance of the interconnection network from different perspectives. Each network topology can be represented using graph theory as a graph $G = (V,E,\psi)$, where each node $v_i \in V$ represents a PE. E is a set of edges that represent the logical communication channels between PEs. ψ is the graph mapping incident function $\psi : E \rightarrow V \times V$, which maps an edge onto a pair of vertices (v_i,v_j) [33].

Consequently, the traffic distribution figure of a system (intermodule communications between all PEs in number of packets per time step) can be represented by a graph. Fig. 2a shows an example of a system represented by a Traffic Distribution Graph (TDG) $G = (V,E,\psi)$. Each edge $e_{ij} \in E$ has a weigh factor λ_{ij} which represents the average number of packets per time step transmitted from v_i to v_j, $1 \leq i,j \leq n$; where n is the number of PEs [9]. This graph can also be represented in a traffic distribution matrix form (λ), as shown in Fig. 2b. As shown in the figure, if two vertices do not have a direct connection, a weigh of zero is given to the corresponding entry in the λ matrix.

For each network topology, a unique connectivity matrix (C) is generated [34]. This matrix represents the minimum number of links a packet goes through during its transition from the one node to another node [9]. For example, the connectivity matrix (C) of the ring network topology shown in Fig. 2a is shown in Fig. 2c.



$$\lambda = \begin{bmatrix} 0 & \lambda_{12} & \lambda_{13} & \lambda_{14} \\ \lambda_{21} & 0 & \lambda_{23} & \lambda_{24} \\ \lambda_{31} & \lambda_{32} & 0 & \lambda_{34} \\ \lambda_{41} & \lambda_{42} & \lambda_{43} & 0 \end{bmatrix} \qquad C = \begin{bmatrix} 0 & 1 & 2 & 1 \\ 1 & 0 & 1 & 2 \\ 2 & 1 & 0 & 1 \\ 1 & 2 & 1 & 0 \end{bmatrix}$$

(a) *(b)* *(c)*

Figure 2 (a) Traffic distribution graph (TDG) for a ring topology, (b) its corresponding traffic distribution matrix (λ), and (c) its corresponding connectivity matrix (C)

B. System analysis of the power consumption on the global interconnection link

Global interconnection links means the links among all routers in the network. Assume that one unit power is consumed when a packet is transmitted over a unit length. Then, the power consumption of global interconnection links depends on the target topology and is measured at an abstract level by calculating the number of power units consumed to transmit all data packets given in the TDG. For instance, to transmit a packet from node 4 to node 2 in the ring topology shown in Fig. 2a, two power units are to be consumed.

Also, depending on the communication traffic between the PEs, different mappings on the same network topology will yield different power consumption on the global interconnection links. All links within standard topologies are assumed to have equal lengths. However, for some topologies this is not true (such as torus topology), there are some links do not have the same length as others. To address this problem, the set of entries corresponding to those links in the connectivity matrix C is multiplied by an adjustment factor to correct the length of those links [9]. At the system level, the total power consumed in the global links of a network topology can be estimated from [9]:

$$P_{sys} = \sum_{i=1}^{n} \sum_{j=1}^{n} \lambda_{ij} * C_{ij} * U_p \quad (1)$$

Where i and j are the source and destination node indexes respectively, λ_{ij} is the average number of packets per time step associated with each logical link. C_{ij} is the minimum number of links needed to transmit these packets from their source to destination on a certain topology. U_p represents a unit power. Assuming a unit power is consumed when a packet is transmitted over a unit length, the summation (P_{sys}) represents the total power consumption of global links.

IV. Proposed Algorithm

The first step to reduce the power consumption on the global interconnection links is to select an initial topology and perform an initial mapping. Then, using equation (1), the total power consumption on the links is calculated. The second step is to re-map the PEs to different locations in the selected topology and recalculate the total power consumption. In the exhaustive algorithm, all possible mappings are examined until the highest traffic rates are associated with shortest distances in the selected topology and reach the mapping with the lowest power consumption. In the proposed optimization algorithm, very less mappings are needed to reach the same total power mconsuption and thus taking much less time than in the exhaustive algorithm. This process is repeated with each one of the 9 topologies. Based on the obtained results, the topology which has the minimum power along with its corresponding mapping is selected. The exhaustive algorithm has been developed for the sake of comparisons with the proposed algorithm. In the following sub-section, the exhaustive algorithm is presented. Then, we discuss the proposed optimization algorithm.

A. The exhaustive algorithm

As mentioned above, in this algorithm all possible physical placements of the PEs onto the network locations are addressed in order to reach the best mapping in the specific network topology (i.e.: the complexity of this algorithm is O(N!) where N is the number of available locations in the current topology). Algorithm 1 shows the exhaustive algorithm.

Algorithm 1 Exhaustive algorithm

```
Let n the number of PEs
Let λ an n ×n traffic distribution matrix
Let C an n ×n connectivity matrix for a specific network topology
Let V an n elements vector which represent the mapping of PEs on the
network locations
Let Map an n elements vector which represent the final mapping
// initial mapping
for i = 1 to n do
   V[i] = i
   Map[i] = V[i]
end for
// finding the optimum mapping
P_sys = Calc_Psys( )
min_p = P_sys
for i = 2 to n! do
   V = get_next_mapping( )
   Restore_C(V)
   P_sys = Calc_Psys( )
   if P_sys < min_p then
      min_p = P_sys
      for k = 1 to n do
         Map[i] = V[i]
      end for
   end if
end for
```

In Algorithm 1, the V vector is initiated so that PE_i is

mapped onto location i in the target topology and the system power is calculated using (1). The algorithm iterates through all possible mappings (permutations of V) to find the optimum mapping which has the lowest system power consumption. In each mapping, the Connectivity matrix must be restored to reflect this mapping; this is done by swapping the appropriate rows and columns in the connectivity matrix. The exhaustive algorithm must explore all the possible mappings (n!) in order to find the optimum mapping, thus taking more time which will become unacceptable especially for large number of PEs.

B. The proposed optimization algorithm

The proposed algorithm to reduce the power consumption on the global interconnection links aims at reducing the time needed to find the optimum mapping. This is done by reducing the number of permutations for the mapping vector V which is needed to be explored to get the optimum mapping. Fig. 3 shows a flowchart of the proposed algorithm.

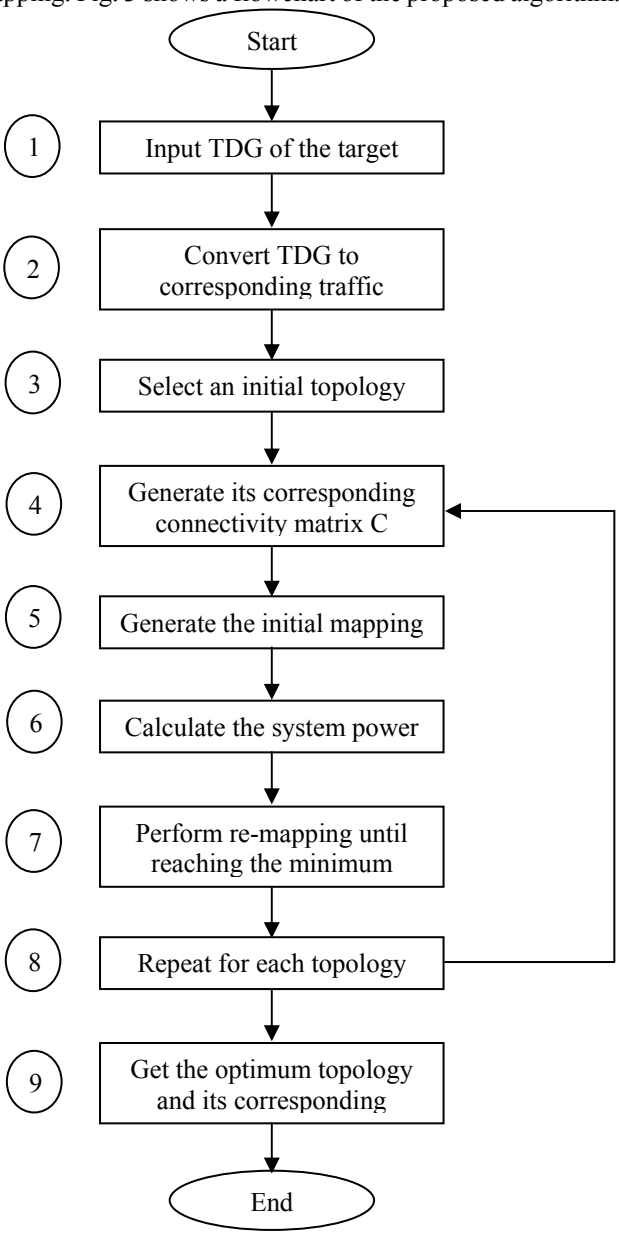


Figure 3 A Flowchart of the proposed optimization algorithm

Steps 1-4: Formal transformation of the TDG and the network topology

The first step of the proposed algorithm is to transform the TDG into a traffic distribution matrix and the target network topology into a connectivity matrix. Then, these representations will be used in the rest of the algorithm to find the optimum network topology and its corresponding mapping.

Step 5: Generate the initial mapping

A heuristic algorithm to find the initial mapping which is close to the optimum mapping is developed. Algorithm 2 shows the initial mapping algorithm. In this algorithm, the PE with the maximum traffic is placed onto the location with best connectivity (has minimum total connectivity from the connectivity matrix) and has number of neighbors that best match the number of PEs directly connected to it. Then the PE that has a maximum traffic with this placed PE is chosen to be placed in a neighbor location. This procedure is repeated until all PEs are placed.

Algorithm 2 Initial mapping algorithm

```
Let n the number of PEs
Let Max_Neighbor the maximum number of neighbors for any given location
Let P_Elements the set of all PEs
Let Mapped the set of mapped PEs
Let Not_Mapped the set of not mapped PEs
Let Loc the set of all locations
Let Neighbors an n elements vector which stores the number of neighbors to
each location in Loc
Sort locations in Loc according to their total connectivity
Sort P_Elements according to their total traffic
while P_Elements ≠ ∅ do
  PE_max = the PE which has the maximum total traffic
  num = the number of neighbors of PE_max
  if num > Max_Neighbor then
    num = Max_Neighbor
  end if
  for s = num to Max_Neighbor do
    Loc_max = find the first empty location in Loc with number of neighbors
equal to s
    Map(PE_max , Loc_max)
    Add PE_max to Mapped
    Remove PE_max from P_Elements
    for each neighbor b to Loc_max do
      decrement Neighbors[b] by 1
    end for
  end for
  if PE_max ∉ Mapped then
    for s = num-1 to 1 do
      Loc_max = find the first empty location in Loc with number of
neighbors equal to s
      Map(PE_max , Loc_max)
      Add PE_max to Mapped
      Remove PE_max from P_Elements
      for each neighbor b to Loc_max do
        decrement Neighbors[b] by 1
      end for
    end for
  end if
  if PE_max ∉ Mapped then
    Add PE_max to Not_Mapped
    Remove PE_max from P_Elements
  else
    for each neighbor b to Loc_max do
      P = find the PE with the maximum traffic with PE_max
      if P ∉ Mapped then
        L = the first free neighbor to Loc_max
        Map(P , L)
        Add P to Mapped
        Remove P from P_Elements
        for each neighbor b to L do
          decrement Neighbors[b] by 1
        end for
      end if
```

```
    end for
  end if
end while
for each PE ∈ Not_Mapped do
  Location = find a free location randomly
  Map(PE , Location)
end for
```

Step 6-7: Finding the optimum mapping

In step 6, the system power consumption for the initial mapping is calculated. Then, step 7 performs re-mapping until reaching the minimum power consumption. The re-mapping steps are shown in the flowchart in Fig. 4. As shown in the flowchart, the re-mapping process consists of three steps. The first step, Minimum-Maximum Iteration, is executed in two main sequential phases. The first phase (Minimum phase), as shown by the flowchart in Fig. 5, iterates on the TDG from the minimum traffic value until reaching the maximum traffic value. In each iteration, it re-locates the Source-Destination pair in different possible locations in the target topology (i.e.: the complexity of this step is $O(N^2)$ where N is the number of available locations in the current topology). In each re-location step, the system power is computed in order to reach to the minimum power consumption. For fully connected TDG, the number of traffic values is M(M-1)/2 where M is the total number of tasks. So, the complexity of the Minimum phase is $O(M^2N^2)$ where M ≤ N. In the worst case where M = N, the complexity of the Minimum phase is $O(N^4)$. Accordingly, the complexity of the proposed algorithm ($O(N^4)$) is less than that of the exhaustive algorithm (O(N!)) specially for N > 6 which coincides with real systems.

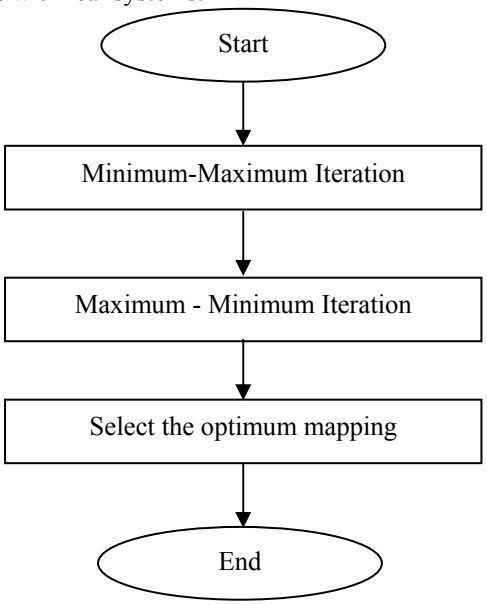


Figure 4 Flowchart shows the re-mapping steps

The second phase (Maximum phase), performs quite similar to the first phase, but it iterates on the TDG from the maximum traffic value until reaching the minimum traffic value.

The Minimum phase tries to optimize the PE mapping starting from the small traffic values. Then, the Maximum phase tries to re-optimize the PE mapping starting from the large traffic values. The execution of these two phases in this sequence cover all the cases that need to allocate PEs with small traffic values in best locations before allocating the PEs with larger traffic values in order to reach the optimum mapping.

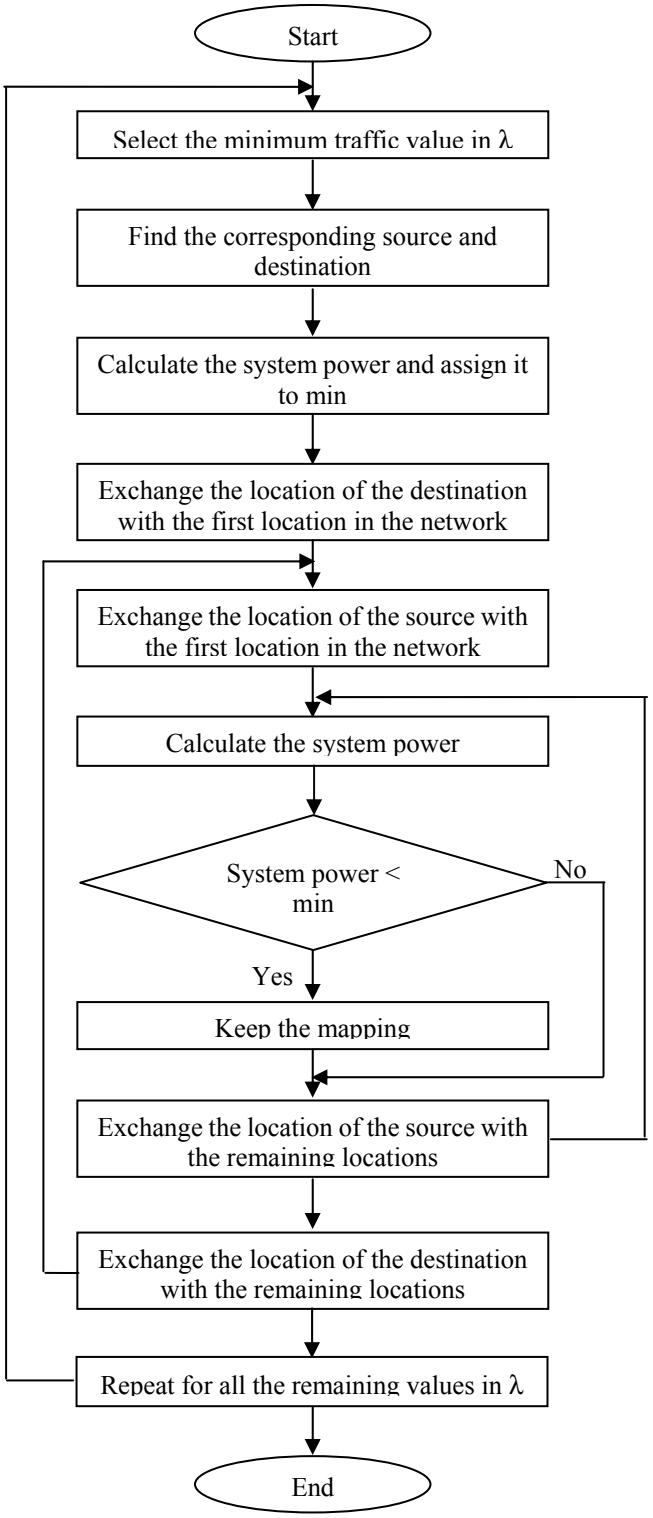


Figure 5 The Minimum phase of the re-mapping process

In other cases, this is not the case; we need to allocate PEs with large traffic values in best locations before allocating the PEs with smaller traffic values in order to reach the optimum mapping. To cope with these cases, the second step, Maximum-Minimum Iteration, in the re-mapping process is also executed. Finally, the results of the two steps are compared and the optimum solution is captured.

Step 8-9: Finding the optimum topology and its corresponding mapping

In step 8, another topology from the 9 standard topologies is examined until finally we get the optimum topology and its corresponding mapping in step 9.

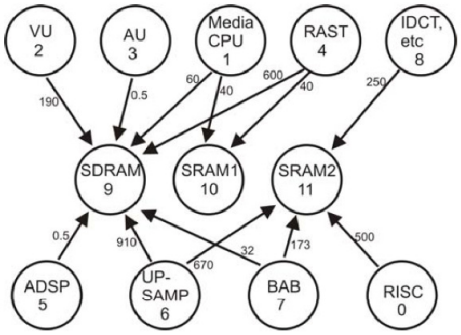
V. Experimental Results

The proposed optimization algorithm is validated through MPEG4 decoder application [35]. It is taken as an application to evaluate the proposed algorithm.

Fig. 6 shows the core task graph of the MPEG4 decoder and its corresponding traffic distribution matrix (λ).

The exhaustive algorithm and the proposed optimization algorithm are implemented using C# programming language. The two algorithms are executed in the cases of 7 to 10 tasks. Nine standard topologies (Mesh, Torus, X Torus, Ring, Octagon (Oct), Binary tree (BT), Butterfly Fat Tree (BFT), Hypercube (Hcube), and Star) are used. Table I through Table IV show the optimal mapping, power consumption on the global interconnection links, and the number of permutations used to find the optimal mapping for the exhaustive algorithm and the proposed optimization algorithm. These tables show the results for the cases of the first seven, eight, nine, and ten tasks of the MPEG4 decoder application respectively.

Based on these results, it can be concluded that the 2D Tours topology and the X Torus topology are the optimal topologies in terms of power consumed on the global interconnection links for all the test cases.



$$\lambda = \begin{bmatrix} 0 & 0 & 0 & 0 & 190 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 60 & 40 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 600 & 40 & 0 & 0 & 0 & 0 & 0 & 0 \\ 190 & 0.5 & 60 & 600 & 0 & 0 & 0 & 0 & 0.5 & 910 & 32 & 0 \\ 0 & 0 & 40 & 40 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 250 & 0 & 670 & 173 & 500 \\ 0 & 0 & 0 & 0 & 0 & 0 & 250 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 910 & 0 & 670 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 32 & 0 & 173 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 500 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

(a) *(b)*

Figure 6 (a) MPEG4 decoder core graph and (b) its corresponding traffic distribution matrix (λ)

TABLE I Experimental Results for the Case of the First Seven Tasks of the MPEG4 Decoder Application

Topology	Exhaustive Algorithm		Proposed Algorithm		Optimum system power
	Optimum Mapping	Permutations	Optimum Mapping	Permutations	
Ring	1 2 7 6 3 4 5	5040	7 2 1 5 4 3 6	865	1031
Star	5 2 3 4 1 6 7	5040	5 4 2 3 1 6 7	865	1010.5
Binary Tree	3 5 6 4 1 2 7	5040	4 5 6 1 3 2 7	865	1011.5
Butterfly Fat Tree	1 5 3 4 2 6 7	5040	1 4 5 3 6 2 7	865	2022
2D Mesh	7 2 8 4 5 1 6 3 9	362880	6 4 7 3 5 2 8 1 9	1537	930.5
2D Tours	1 5 3 8 2 9 7 4 6	362880	6 3 7 4 5 1 8 2 9	1537	930.5
X Tours	1 2 6 5 4 3 7 8 9	362880	5 4 2 3 6 1 7 8 9	1537	930.5
Hypercube	1 2 3 5 7 8 6 4	40320	5 4 2 1 3 6 8 7	1373	932
Octagon	1 2 6 4 8 7 3 5	40320	3 2 7 4 5 1 8 6	1373	932

TABLE II Experimental Results for the Case of the First Eight Tasks of the MPEG4 Decoder Application

Topology	Exhaustive Algorithm		Proposed Algorithm		Optimum system power
	Optimum Mapping	Permutations	Optimum Mapping	Permutations	
Ring	2 1 5 4 3 6 7 8	40320	6 3 4 5 1 2 8 7	1373	1281
Star	5 2 3 4 1 6 7 8	40320	5 4 2 3 1 6 7 8	1373	1510.5
Binary Tree	1 5 7 4 3 8 2 6	40320	4 5 8 3 1 7 2 6	1373	1261.5
Butterfly Fat Tree	1 5 3 4 2 6 7 8	40320	6 2 7 8 1 3 5 4	1373	2522
2D Mesh	1 2 7 5 4 8 3 6 9	362880	6 4 2 3 5 1 8 7 9	1793	1181
2D Tours	1 5 3 8 2 9 7 4 6	362880	6 3 9 4 5 1 7 2 8	1793	1180.5
X Tours	1 2 6 5 4 3 7 8 9	362880	5 4 1 2 7 3 8 6 9	1793	1180.5
Hypercube	1 2 3 5 8 7 6 4	40320	5 4 7 1 3 6 8 2	1373	1181
Octagon	1 2 6 4 8 7 3 5	40320	1 5 4 8 7 3 6 2	1373	1181

TABLE III Experimental Results for the Case of the First Nine Tasks of the MPEG4 Decoder Application

Topology	Exhaustive Algorithm		Proposed Algorithm		Optimum system power
	Optimum Mapping	Permutations	Optimum Mapping	Permutations	
Ring	1 2 9 8 7 6 3 4 5	362880	9 2 7 8 6 3 4 5 1	2049	1282.5

Star	5 2 3 4 1 6 7 8 9	362880	5 4 2 3 1 9 6 7 8	2049	1511
Binary Tree	1 5 7 4 3 8 2 6 9	362880	4 5 8 3 1 7 2 6 9	2049	1262.5
Butterfly Fat Tree	1 5 3 4 2 6 7 8 9	362880	6 2 7 8 1 3 5 4 9	2049	2524
2D Mesh	2 1 8 4 5 7 6 3 9	362880	6 4 2 3 5 1 8 7 9	2049	1182
2D Tours	1 6 3 8 7 2 9 4 5	362880	4 6 1 7 8 2 9 3 5	2049	1181
X Tours	1 2 6 5 4 3 7 8 9	362880	5 4 2 3 7 1 9 8 6	2049	1181
Hypercube	5 2 9 4 3 8 7 6 1	362880	5 4 7 2 3 6 8 9 1	2049	1181.5
Octagon	1 5 3 7 2 4 6 8 9	362880	2 5 4 8 7 3 6 9 1	2049	1181.5

TABLE IV EXPERIMENTAL RESULTS FOR THE CASE OF THE FIRST TEN TASKS OF THE MPEG4 DECODER APPLICATION

Topology	**Exhaustive Algorithm**		**Proposed Algorithm**		**Optimum system power**
	Optimum Mapping	**Permutations**	**Optimum Mapping**	**Permutations**	
Ring	1 3 6 2 9 8 7 10 5 4	3628800	5 10 7 8 2 9 6 3 1 4	3241	3154.5
Star	5 2 3 4 1 6 7 8 9 10	3628800	5 10 4 2 3 1 9 6 7 8	3241	3761
Binary Tree	10 5 7 4 1 9 8 3 6 2	3628800	10 5 7 4 1 2 8 3 6 9	3241	2862.5
Butterfly Fat Tree	1 2 3 6 5 4 7 10 9 8	3628800	4 10 7 5 8 2 3 6 1 9	3241	6604
2D Mesh	2 1 7 8 4 5 10 9 6 3	3628800	2 10 7 8 4 5 1 9 6 3	3241	2762
2D Tours	1 2 8 9 5 3 7 10 4 6	3628800	2 10 7 8 4 5 3 6 9 1	3241	2762
X Tours	1 2 3 5 8 7 6 4 9 10	3628800	7 10 5 3 8 2 1 9 4 6	3241	2762
Hypercube	10 5 3 7 2 4 6 8 9 1	3628800	5 10 7 4 3 2 8 6 1 9	3241	2762
Octagon	10 5 3 8 2 4 6 7 1 9	3628800	3 5 10 2 6 4 7 8 1 9	3241	2762

From the above tables, it is noticed that the proposed optimization algorithm achieved the optimum mapping for all the nine topologies in all test cases. Also, the number of permutations used by the proposed algorithm is much less than the number of permutations used in the exhaustive algorithm, which is coincided with the complexity analysis mentioned in section IV. This means that, we can reach the optimal topology and its corresponding optimal mapping in a much less time than the exhaustive algorithm.

XI. CONCLUSION

A new optimization algorithm to find the network topology and the PEs mapping that matches the traffic characteristics of the NoC-based system and gives the lowest power consumption on the global interconnection links, is presented in this paper. The proposed optimization algorithm explores NoC application synthesis and mapping to nine standard topologies. It uses the graph-theoretic concepts to acquire the optimum NoC topology/mapping that reaches the lowest power dissipation on the global interconnection links for a given application. The efficiency of the proposed algorithm and its benefits over the exhaustive algorithm is verified through MPEG4 video application. Experimental results show that the proposed algorithm reach the optimal solution in a much less time compared to the exhaustive algorithm.

REFERENCES

[1] L. Benini and G. De Micheli, "Networks on chips: a new soc paradigm", IEEE Computer, vol. 35, Jan 2002, pp. 70–78.
[2] T. Schonwald, J. Zimmermann, O. Bringmann, and W. Rosenstiel, "Network-on-Chip Architecture Exploration Framework", in 12th Euromicro Conference on Digital System Design Architectures Methods and Tools, 2009, pp. 375-382.
[3] J. Hu and R. Marculescu, "Energy-aware mapping for tile-based noc architectures under performance constraints", IEEE ASP-DAC, January 2003.
[4] G. Khan, and V. Dumitriu, "Throughput-based network-on-chip topology generation and analysis", in Canadian Conference on Electrical and Computer Engineering, 2009, pp. 180-184.
[5] B.H. Meyer, et al., "Power-performance simulation and design strategies for single-chip heterogeneous multiprocessors", IEEE Transactions on Computers 54, 2005, pp. 684–697.
[6] N. Banerjee, P. Vellanki, and K.S. Chatha, "A power and performance model for network-on-chip architectures", in Proceedings of the Design, Automation and Test in Europe (DATE'04), Paris, France, 2004, pp. 21250–21256.
[7] T. Simunic, S. Boyd, and P. Glynn, "Managing power consumption in networks on chips", IEEE Transactions on Very Large Scale Integration (VLSI) Systems 12, 2004, pp. 96–107.
[8] S. Bhat, "Energy models for networks-on-chip components", Master's thesis, Technische University Eindhoven, Eindhoven, Netherlands, December 2005.
[9] Haytham Elmiligi, Ahmed A. Morgan, M. Watheq El-Kharashi and Fayez Gebali, "Power Optimization for Application-Specific Networks-on-Chips: A Topology-Based Approach", Journal of Microprocessors and Microsystems, Elsevier, Volume 33, Issues 5-6, August 2009, pp. 353-355.
[10] L. Benini, P. Siegel, and G.D. Micheli, "Saving power by synthesizing gated clocks for sequential circuits", IEEE Design and Test 11, 1994, pp. 32–41.
[11] L.-F. Leung and C.-Y. Tsui, "Energy-aware synthesis of networks-on-chip implemented with voltage islands", in Proceedings of the 44th ACM/IEEE Design Automation Conference (DAC'07), San Francisco, CA, 2007, pp. 128– 131.
[12] D. Shin and J. Kim, "Power-aware communication optimization for networks-on-chips with voltage scalable links", in Proceedings of the International Conference on Hardware/Software Co-design and System Synthesis (CODES+ISSS'2004), Stockholm, Sweden, 2004, pp. 170–175.
[13] U.Y. Ogras and R. Marculescu, "Analytical router modeling for networks-on-chip performance analysis", in Proceedings of Design, Automation and Test in Europe Conference (DATE'07), Nice, France, 2007, pp. 1096–1101.
[14] V. Chandra, A. Xu, and H. Schmit, "A low power approach to system level pipelined interconnect design", in Proceedings of the 2004 international workshop on System level interconnect prediction (SLIP'04), Paris, France, 2004, pp. 45–52.
[15] Y. Hu, H. Chen, Y. Zhu, A. Chien, and C.-K. Cheng, "Physical synthesis of energy efficient networks-on-chip through topology exploration and wire style optimization", in Proceedings of the IEEE International Conference on Computer Design: VLSI in Computers and Processors (ICCD'05), San Jose, CA, 2005, pp. 111–118.
[16] S. Kumar, et al., "A network on chip architecture and design methodology", in Proceedings of the IEEE Computer Society Annual Symposium on VLSI (ISVLSI'02), Pittsburgh, PA, 2002, pp. 105–112.
[17] W. Dally and B. Towles, "Route packets, not wires: On-chip interconnection networks", in Proceedings of the 38th Design Automation Conference (DAC'01), Las Vegas, NV, 2001, pp. 683–689.

[18] H. Gu, Q. Xie, K. Wang, J. Zhang and Y. Li," X–Torus: A Variation of Torus Topology with Lower Diameter and Larger Bisection Width", ICCSA 2006, LNCS (3984),2006, pp.149 –157.
[19] J. Xu, "Topological Structure and Analysis of Interconnection Networks", 1st ed., Kluwer Academic Publishers, 2001.
[20] F. Karim, A. Nguyen, and S. Dey, "An interconnect architecture for networking systems on chips", IEEE Micro 22, 2002, pp. 36–45.
[21] P. Pande, C. Grecu, A. Ivanov, and R. Saleh, "Design of a switch for network on chip applications", in Proceedings of the 2003 International Symposium on Circuits and Systems (ISCAS'03), Bangkok, Thailand, 2003, pp. 217–220.
[22] P. Pande, C. Grecu, M. Jones, A. Ivanov, and R. Saleh, "Performance Evaluation and Design Trade-Offs for Network-on-Chip Interconnect Architectures", IEEE Computer, vol. 54-8, August 2005, pp. 1025-1040.
[23] T. Bjerregaard and S. Mahadevan, "A Survey of Research and Practices of Network-on-Chip", ACM Computing Surveys, vol. 38-1, March 2006, pp. 1-51.
[24] J. Hu and R. Marculescu, "Energy-aware mapping for tile-based NoC architectures under performance constraints", in Asia South Pacific Design Automation Conference (ASP-DAC'03), 2003, pp. 233-239.
[25] T. Ye, L. Benini, and G. De Micheli, "Analysis of Power Consumption on Switch Fabrics in Network Routers", in Design Automation and Conference (DAC'02), 2002, pp. 524–529.
[26] C. A. M. Marcon, et al., "Exploring NoC Mapping Strategies: An Energy and Timing Aware Technique", in Design Automation and Test on Europe (DATE'05), 2005, pp. 502-507.
[27] C. A. M. Marcon, "Modelos para o Mapeamento de Aplicações em Infra-estruturas de Comunicação Intrachip". Tese de Doutorado, PPGC - UFRGS, 2005, 192 p. (in Portuguese).
[28] J. Hu and R. Marculescu, "Energy- and Performance-Aware Mapping for Regular NoC Architectures", IEEE Transaction on Computer-Aided Design of Integrated Circuits and Systems, vol. 24-4, April 2005, pp. 551-562.
[29] S. Koohi, M. Mirza-Aghatabar, S. Hessabi, and M. Pedram, "High-Level Modeling Approach for Analyzing the Effects of Traffic Models on Power and Throughput in Mesh-Based NoCs", in International Conference on VLSI Design (VLSID'08), 2008, pp. 415-420.
[30] T. Ahonen, D.A. Siguenza-Tortosa, H. Bin, and J. Nurmi, "Topology optimization for application-specific networks-on-chip", in Proceedings of the 2004 International Workshop on System Level Interconnect Prediction (SLIP'04), Paris, France, 2004, pp. 53–60.
[31] U.Y. Ogras and R. Marculescu, "Application-specific network-on-chip architecture customization via long-range link insertion", in Proceedings of the IEEE/ACM International Conference on Computer-Aided Design (ICCAD'05), San Jose, CA, 2005, pp. 246–253.
[32] S. Murali and G. De Micheli, "SUNMAP: A tool for automatic topology selection and generation for NoCs", in Proceedings of the 41st Design Automation Conference (DAC'04), San Diego, USA, 2004, pp. 914–919.
[33] G. Agnarsson and R. Greenlaw, "Graph Theory: Modeling, Applications, and Algorithms", Prentice-Hall, 2006.
[34] J.L. Gross and J. Yellen, "Graph Theory and Its Applications", second ed., CRC Press, 2005.
[35] D. Bertozzi and L. Benini, "Xpipes: a network-on-chip architecture for gigascale systems-on-chip", IEEE Circuits and Systems Magazine, vol. 4, no. 2, 2004, pp. 18–31.