Elixir: A new bandwidth-constrained mapping for Networks-on-chip

Midia Reshadi1a), Ahmad Khademzadeh2, and Akram Reza1

1 Islamic Azad University, Science and Research Branch, Tehran, Iran
2 Iran Telecommunication Research Center, Tehran, Iran

a) reshadi@srbiau.ac.ir

Abstract: Nowadays, with technology shrinking and the huge demand for supporting multiple applications has led designers to use multiple IP cores within a single chip. Therefore, the designers have proposed Networks-on-chip to overcome the problems of future complex systems. Mapping IPs directly affects NoC design parameters such as latency and power consumption. In this paper we present a power and performance aware mapping technique based on the combination of both the bandwidth-constrained and branch and bound concepts. Results have shown improvements of the latency and power consumption of our technique when compared to other popular NoC mappings.

Keywords: Networks-on-chip, mapping, communication cost, bandwidth constrains, power consumption, average network latency

Classification: Integrated circuits

References

1 Introduction

The complexity of increasing the number of IPs will be a major problem in the future and also with the current SoCs. At the same time, the use of standard hardwired busses to interconnect these cores is not scalable [1].

Thus, in order to cope with these problems, Networks-on-chip (NoC) has been recently proposed as a potent solution. In Networks-on-chip all IP cores are connected via the packet based communication infrastructure. The NoC not only solves the scalability problem but has several advantages as well as better structure, performance and modularity [1].

Assigning the IP cores to the NoC platform and allocating the bandwidth across the links will greatly impact both the performance and power consumption of the network [2]. By allocating higher bandwidth across the links of the NoC, more energy is dissipated. Thus, it is important to balance the bandwidth needs across the different links. In this paper we present Elixir, a fast algorithm which computes the lowest communication cost map-
ping with minimum average latency and power consumption. Furthermore, our approach is not limited just to the mesh topology but we have chosen the mesh because it is a fair prototype when comparing the results of proposed mapping. The rest of this paper is organized as follows. Section 2 presents the mathematical formulation of mapping problem, section 3 covers Elixir mapping algorithm in detail. Section 4 presents the simulation results. Lastly, section 5 is our conclusion.

## 2 Problem Formulation

First, we want to present some definitions regarding the application in addition to NoC architecture below.

**Definition 1:** The core graph $G(V, E)$ is a directed graph where each $v_i \in V$ denotes a core and the directed edge $e_{i,j} = (v_i, v_j) \in E$ denotes the communication trace between the cores $v_i$ and $v_j$. For every $e_{i,j} = (v_i, v_j) \in E$, $comm_{i,j}$ represents the bandwidth of the communication from $v_i$ to $v_j$.

**Definition 2:** The NoC topology graph $T(N, L)$ is a directed graph, where each $n_i \in N$ denotes a tile and the directed edge $l_{i,j} = (n_i, n_j) \in L$ denotes a direct physical link between the tiles $n_i$ and $n_j$. For every $l_{i,j} = (n_i, n_j) \in L$, $bw_{i,j}$ represents the bandwidth available across the edge $l_{i,j}$.

In this paper we have limited our consideration to the mesh network with the XY routing. Thus the mapping function is defined as the following:

$$map: V \rightarrow N, \ s.t. \ \ map(v_i) = n_j, \ \forall v_i \in V, \ \exists n_j \in N \ (1)$$

In the core graph, $d^k$ is the communication between each pair of cores. $d^k$ is treated as a flow of single commodity and the value of $d^k$ is denoted by $vl(d^k)$. The set of all commodities is represented by $D$ and is defined as:

$$D = \left\{ d^k: vl(d^k) = comm_{i,j}, \ k = 1, 2, \ldots, |E|, \ \forall e_{i,j} \in E, \ with \ source(d^k) = map(v_i), \ dest(d^k) = map(v_j) \right\} \ (2)$$

The bandwidth constraints are represented by the inequality:

$$\forall \text{link } l_{i,j} \in L, \ \sum_{k=1}^{|E|} f(l_{i,j}, path(source(d^k), dest(d^k))) \leq BW(l_{i,j}) \ (3)$$

$$f(l_{i,j}, path(l, m)) = \begin{cases} vl(d^k) & \text{if } l_{i,j} \in path(l, m) \\ 0 & \text{otherwise} \end{cases} \ (4)$$

where the set $Path(l, m)$ represents the set of links between the mesh nodes $l$ and $m$ according to the XY routing. Likewise, the set path can be defined for other deadlock free and the shortest path routing algorithms.

If the bandwidth constraints are satisfied then the communication cost is given by:

$$\text{comm cost} = \sum_{k=1}^{|E|} vl(d^k) \text{hopcount}(source(d^k), dest(d^k)) \ (5)$$

Where $\text{hopcount}(a, b)$ is the minimum number of hops between nodes $a$ and $b$. 
After mapping and the bandwidth allocation, the allocated bandwidth of the network per each commodity is obtained by:

\[
\text{AllocatedBW} = \sum_{k=1,2,\ldots,|E|}^{\forall l} \frac{BW_{i,j}}{v_l(d_k)} \quad \forall l \in \text{path}(\text{source}(d_k), \text{dest}(d_k)) \tag{6}
\]

Then the problem statement is as follows.

The fast and automated mapping algorithm has to map the IP cores \(v_i \in V\) onto the tiles \(n_i \in N\) of a topology according to the bandwidth constraint of each channel so that the communication cost, average latency and power consumption are minimized.

### 3 Elixir Mapping Algorithm

Our approach is based up branch and bound \[4\] and the bandwidth-constrained mapping \[3\] concepts. The algorithm has two steps which is as follows. Step 1 computes a set of mappings with the lowest communication cost based on the search tree. Then, step 2 returns a mapping with minimum average latency and power consumption.

As it is shown in Fig. 2(a), firstly in step 1 the core that has maximum communication demand is placed onto one of the topology tiles which has the same degree or maximum number of neighbors. As it is depicted in Fig. 1, four tiles are the candidate for the first IP. Afterwards, our algorithm searches the optimal solution by altering the following two operations:

- **Branch:** In this operation, an unexpanded node is selected from the search tree, and then the next unmapped IP which communicates mostly with the already mapped cores is selected. Then the selected IP is assigned to the set of remaining unoccupied tiles and after that the corresponding new child nodes are generated.

- **Bound:** The communication cost of each newly generated child nodes are calculated then, the lowest communication cost at each level of

![Fig. 1. Example Search tree](image-url)
the tree is specified. Then a node with higher communication cost is trimmed away without further expansion.

The following steps are repeated until all of the mapping candidates are generated. As previously stated, Elixir is a fast algorithm with low computational complexity. The worst case computational complexity of step 1 (i.e., computing the low communication cost candidate mappings) is as follows:

\[ O(|V| \cdot |N|) \]  

Where \(|V|\) is the number of IP cores in core graph and \(|N|\) is a number of tiles in topology graph. In step 2 all mapping scenarios are evaluated by a Polaris toolchain [5, 14] and the mapping with minimum power consumption and average network latency is selected. The flowchart of synthesis design process is given in Fig. 2b.

### 4 Simulation Results

As previously mentioned, we use the Polaris toolchain to estimate the power and average network latency of step 1’s candidate mappings. Finally, the best mapping is chosen along with minimum network latency and power consumption.
Fig. 3. (a) Communication cost of mapping algorithms, (b) Average network latency in 50 nm technology process (c) Power consumption in 50 nm technology process (d) Allocated bandwidth of the MPEG4 mapping (e) Allocated bandwidth of the VOPD mapping
Polaris has built upon three tools, Trident [6], Luna [7] and Orion [8, 9]. Instead of Trident, we have developed a tool named Atlas that automatically computes the Elixir mapping and constructs a synthetic traffic traces of the candidate mappings.

LUNA [7] is an analytical framework that captures the synthetic traffic of Atlas to estimate the levels of network resource utilization. Moreover Luna can analyzes up to 360x faster than other cycle accurate simulators.

Orion [8, 9] is a library of power, area, and router pipeline delay models [10] based on ITRS [11], BPTM [12] and CACTI [13] parameters which estimates and predicts both power and CMOS area in various technology processes. Fig. 3 (a), Fig. 3 (b) and Fig. 3 (c) show the communication cost, power consumption and average network latency of Elixir, NMAP [3], PBB [4], Onyx [15] and BMAP [16]. Note that all simulations were performed based on 50 nm technology process. Results show Elixir outperforms other mapping algorithms in the terms of the average latency and power consumption. In the absence of the congestion, the communication cost is a good indicator of power consumption and network latency. As it can be seen in Fig. 3 (a), Elixir and Onyx have the minimum communication cost when compared to the other mapping algorithms. The network contention is another factor that affects the network latency and power consumption [17]. In order to compare the network contention of mapping algorithms, we have calculated the allocated bandwidth. As seen from Fig. 3 (d) and Fig. 3 (e) Elixir gives the lowest allocated bandwidth compared to the other mapping algorithms.

5 Conclusion

In this paper we have presented a bandwidth-constrained mapping algorithm called Elixir for the fast and automated design of NoC architectures. Elixir automatically maps the IP cores onto the tiles of NoC so that the total communication cost, power consumption and average latency are minimized. Our approach can be extended to map cores onto various NoC topologies. However in this paper we have focused on the tile-based 2D mesh network with XY routing. This proposed mapping consists of two steps. The first step computes the mappings with minimum communication cost based on the search tree concept. Finally, by using polaris toolchain, the best mapping with minimum power consumption and latency is selected. Results show the improvements of power consumption and average network latency in comparison to other popular mapping algorithms.

6 Acknowledgement

The authors would like to thank Noel Eisley from Princeton University for his help on compiling the Polaris toolchain. We wish to thank Wein-Tsung Shen from National Taiwan University for providing the BMAP simulation source code and results.