Application-Specific Topology Design Customization for STNoC

Gianluca Palermo† Cristina Silvano‡ Giovanni Mariani† Riccardo Locatelli♯ Marcello Coppola‡

†Politecnico di Milano - Dipartimento di Elettronica e Informazione
‡ALaRI - Faculty of informatics - University of Lugano
♯STMicroelectronics - Advanced System Technology

Abstract

Customized network-oriented communication architectures have recently become a must to support high bandwidth SoCs. To this end, a corresponding communication design flow is necessary to support the design space exploration of complex SoCs with tight design constraints. In order to exploit the benefits introduced by the NoC approach for the on-chip communication, the paper presents a Pareto Simulated Annealing (PSA) approach for the customization of the network topology. The proposed PSA approach has been applied to STNoC, the Network on-Chip developed by STMicroelectronics. Starting from the ring topology, the proposed application-specific design flow tries to find a set of customized topologies (optimized in terms of performance and area/energy overhead) by adding custom links up to the spidergon topology.

1 Introduction

Design-time specialization is an important factor for the Network-on-Chip (NoC) [1, 4] paradigm. In fact, most NoC architectures have been specifically developed either for one embedded application or as a platform for a small class of applications and consequently, the traffic characteristics cover an important role for the network customization [5–8].

Designing a NoC architecture, one of the first steps to be done is the topology selection. Due to the regularity characteristics and the success in macro-networks, standard topologies (such as 2D mesh and torus) are selected as basis for the on-chip network infrastructure. However, in the on-chip domain, the connectivity theoretically offered by standard topologies cannot be fully exploited due to the nature of communication traffic in real embedded applications. In other word, we can pay in area and power overhead more than the performance benefits that the topology can offer. In some application-specific scenarios, an approach based on the customization of simple standard topologies can be more effective in terms area/energy/performance trade-off with respect to more complex standard topologies.

In this direction, the proposed approach starts from a simple topology (such as ring) for the following application-specific topology customization process to boost the network performance without requiring the area and energy overhead of more complex networks. In particular, this paper presents an application-specific approach for the topology customization of STNoC [2, 10], the Network-on-Chip developed by STMicroelectronics. Starting from the ring topology, the proposed technique tries to find the optimal topology in the STNoC family of topologies (ranging from ring to spidergon) by adding a set of custom links to optimize the network performance on the target application. The main purpose of the paper is not to propose the STNoC architecture, that has already been presented in [2, 10], but to proposed a design methodology to customize STNoC for application-specific designs.

The paper is organized as follows. In Section 2 and 3, the problem description and the proposed approach are described respectively. The experimental results and conclusions are shown in Section 4 and 5.

2 Problem Definition

Before starting the description of the design flow used for the STNoC topology customization, let us introduce some theoretical concepts. First of all, we define two main graphs that are the object of our discussions: the core graph, used to describe the target application, and the topology graph, used to describe the network topology.

The core graph (also called application graph or communication graph) is a directed graph, \( G(V, E) \) where each vertex \( v_i \in V \) represents a core and each direct edge \( (v_i, v_j) \), denoted as \( e_{i,j} \in E \), represents the communication between the cores \( v_i \) and \( v_j \). The weight of the edge \( e_{i,j} \) represents the bandwidth of the communication from \( v_i \) and \( v_j \).

The NoC topology graph is a directed graph \( P(U, F) \) where each vertex \( u_i \in U \) represents a node in the topology and each direct edge \( (u_i, u_j) \), denoted as \( f_{i,j} \in F \), repre-
sents a direct communication between vertex $u_i$ and vertex $u_j$. The weight of the edge $f_{i,j}$, denoted by $bw_{i,j}$, represents the bandwidth available across the edge $f_{i,j}$.

The problem of mapping cores on to network tiles can be represented by the one-to-one mapping function that maps the core graph $G(V, E)$ on to the NoC topology graph $P(U, F)$ optimizing a set of target metrics. On the other hand, the problem of the application specific topology customization can be defined as the search of the best $F$ set in the topology graph $P(U, F)$ once the mapping has already been performed. In particular, for the STNoC customization, the basic idea is to start with a $F_{ring}$ set considering only the links of the ring topology, then to find the best $F_{cross}$ set of links to be added ($F = F' \cup F_{cross}$).

### 3 Proposed Approach

STNoC [10] is a flexible and scalable packet-based on-chip micro-network developed by STMicroelectronics and designed according to a layered methodology for the communication design. One of the most important features is the spidergon topology [2] that enables a cost-effective silicon implementation of the key components: routers and network interfaces. In fact, STNoC includes a flexible router architecture with up to 3 input/output ports for intra-router communication. This feature enables to optimize STNoC IP-blocks performance, improve their maintainability and reduce their verification effort, while keeping efficient application customization in a reasonable design time.

Although spidergon already offers good characteristics for SoCs in terms of network diameter, connectivity and regularity, a customized topology could further satisfy the requirements imposed by the target embedded application. Considering that the application-specific STNoC topology should be in the range from ring to spidergon, our problem is to find the best solution in this topology design space. Since the complexity of on-chip interconnect could not be increased too much and a fully customized topology would not be feasible in a reasonable design time, we start from the simple ring topology by adding customized unidirectional links up to the maximum given by the spidergon (one more input and one more output port for each node of the ring topology).

The starting point for the topology customization phase is the basic ring topology, for which the mapping phase has already been performed. Considering a network with $N$ nodes, the design space for the topology customization phase has a cardinality of $\binom{N}{2}$, although the limitation of only one input/output added link. Even for a limited number of nodes, the problem cannot be solved by using the exhaustive approach in a reasonable design time, so heuristic approaches have been introduced. In this paper, we propose to extend the well-known simulated annealing algorithm to the case of multi-objective exploration. In fact, since we have to consider multiple cost functions, the on-chip topology customization represents a typical multi-objective optimization problem.

#### 3.1 Multi-Objective Exploration

The given optimization problem involves the minimization of multiple objectives (area, network latency, power consumption) making the definition of optimality not unique. In fact, a system which is the best from the performance point of view, can be the worst in terms of power consumption and vice-versa. To address this problem, some concepts from the theory of multi-objective optimization have been introduced hereafter.

When the generic multi-objective design space optimization problem includes a set of $m$ objective functions, the optimization goal consists of minimizing (or maximizing) the function: $\vec{g} = \vec{f}(\vec{x}) = (f_1(\vec{x}), f_2(\vec{x}), \ldots, f_m(\vec{x}))$ where $\vec{x}$ is the design vector and $\vec{f}$ is the objective function vector.

In single-objective optimization, the feasible set is totally ordered according to the objective function $f$. When several objectives are involved, the situation changes and the feasible set is partially ordered. Formally, the relations $=, <, \leq, >$ and $\geq$ can be extended to objective vectors in the following way.

For any two objective vectors of $k$ elements $\vec{g}$ and $\vec{h}$, we have:

$$\vec{g} = \vec{h} \iff \forall i \in \{1, 2, \ldots, k\} : g_i = h_i$$

$$\vec{g} \leq \vec{h} \iff \forall i \in \{1, 2, \ldots, k\} : g_i \leq h_i$$

$$\vec{g} < \vec{h} \iff (\vec{g} \leq \vec{h}) \land (\vec{g} \neq \vec{h})$$

The relations $\geq$ and $>$ are defined similarly.

The previous notions are useful to introduce the concept of Pareto dominance. In fact, for the minimization problem, for any two design vectors $\vec{a}$ and $\vec{b}$, the following concepts can be defined:

- **Dominance ($\prec$):**
  $$\vec{a} \prec \vec{b} \iff \vec{f}(\vec{a}) < \vec{f}(\vec{b})$$

- **Weak Dominance ($\preceq$):**
  $$\vec{a} \preceq \vec{b} \iff \vec{f}(\vec{a}) \leq \vec{f}(\vec{b})$$

- **Indifference ($\sim$):**
  $$\vec{a} \sim \vec{b} \iff (\vec{f}(\vec{a}) \leq \vec{f}(\vec{b})) \land (\vec{f}(\vec{b}) \leq \vec{f}(\vec{a}))$$

For any design vector $\vec{x} \in \Omega$ and a set $A \subseteq \Omega$, vector $\vec{x}$ is said to be non-dominated with respect to the set $A$ iff $\exists \vec{a} \in A : \vec{a} \succ \vec{x}$. Moreover, $\vec{x}$ is said to be Pareto Optimal if $\vec{x}$ is non-dominated with respect to $\Omega$. That is, $\vec{x}$ is optimal in the sense that it cannot be improved in any objective without causing a degradation of at least one of the other objectives.
configuration under analysis and the number of the neighbors which decreases as the time goes on. At each step, the annealing algorithm is called Pareto Simulated Annealing (PSA) [3] and the pseudo-code is outlined in Algorithm 1 where \( S \) represents the population, \( s \) is the current configuration under analysis and \( T \) is the temperature parameter which decreases as the time goes on. At each step, from each \( s \in S \) is generated a new point by using the \( \text{neighbors}(s) \) function and the new \( s' \) solution is evaluated. If the new point \( s' \) results better than the previous one (\( s' \) dominates \( s \)), \( s' \) is definitely accepted, while if \( s' \) is dominated by \( s \) it is accepted depending on the acceptance probability that decreases with the temperature value \( T \). If no one of \( s' \) and \( s \) dominates the other one, the new solution is accepted instead of the current one because moving in a non-dominated solution helps to increase the performance spread and it is useful also to escape from local minima.

More in detail, several design choices characterize the proposed implementation of the PSA:

- The representation of the different solutions in the design space is a string composed of \( n \)-elements, where \( n \) is the number of STNoC nodes in the topology (and also the number of links that can be added to the basic ring topology). The value of the \( k \)-th element represents the target node of the link starting from the node \( k \). If its value is equal to zero it means that the link does not exist. The only number that can have multiple occurrences in the string is zero.

- The neighbor function is implemented in two different ways selected with a uniform probability. The first one consists of swapping two elements, while the second one consists of setting/resetting an element randomly chosen in the string. In particular, if the value is equal to zero, then it is set to a possible value, else (if it is \( \neq 0 \)) it is reset to zero.

- The transition probability \( P_t \) is equal to:
  \[
P_t(s, s', T) = \exp \left( -\frac{\text{Dist}(s, s')}{T} \right)
\]
  where
  \[
  \text{Dist}(s, s') = \sum_{i=1}^{M} \frac{f_i(s') - f_i(s)}{f_i(s)}
  \]
  and \( f_i \) and \( M \) are respectively the \( i \)-th objective function and the number of the objective functions.

- The annealing scheme reflects the typical geometric cooling:
  \[
  T_k = \alpha^k \times T_0
  \]
  where \( 0 < \alpha < 1 \) is the cooling rate.

- The function \( \text{paretoFilter}(S) \) provides the set of non-dominated solutions in \( S \):
  \[
  \text{paretoFilter}(S) = \{ s \in S \mid \nexists s' \in S : s' \succ s \}
  \]
  The set \( \text{paretoFilter}(S) \) is the non-dominated set with respect to \( S \).

- The terminating condition is characterized by the evaluation of a fixed number of solutions. This choice has been done to control the time required by the topology synthesis.

Algorithm 1 Pseudo-Code of the PSA Algorithm

```plaintext
S ← \{\}
T ← T_0
repeat
    while |S| ≠ Population do
        S ← S ∪ \text{rand}()
    end while
    Snext ← \{\}
    for all s ∈ S do
        s' ← neighbors(s)
        if s' ≥ s then
            Snext ← Snext ∪ s'
        else if s ≥ s' then
            if \( P_t(s, s', T) \geq \text{threshold} \) then
                Snext ← Snext ∪ s'
            else
                Snext ← Snext ∪ s
            end if
        else if s ∼ s' then
            Snext ← Snext ∪ s'
        end if
    end for
    Snext ← \text{paretoFilter}(Snext)
    T ← \text{annealing}(T)
    S ← Snext
until end condition
```

3.2 Pareto Simulated Annealing

The simulated annealing is a Monte Carlo approach for minimizing multivariate functions. The term simulated annealing derives from the analogy with the physical process of heating and then slowly cooling a substance to obtain a strong crystalline structure. In the simulated annealing algorithm, a new configuration is constructed by imposing a random displacement. If the cost function of this new state is less than the previous one, the change is accepted unconditionally and the system is updated. If the cost function is greater, the new configuration is accepted probabilistically: the acceptance possibility decreases with the temperature. This procedure allows the system to move consistently towards lower cost function states, thus 'jumping' out of local minima due to the probabilistic acceptance of some upward moves.

The proposed multi-objective extension of the simulated annealing algorithm is called Pareto Simulated Annealing (PSA) [3] and the pseudo-code is outlined in Algorithm 1 where \( S \) represents the population, \( s \) is the current configuration under analysis and \( T \) is the temperature parameter which decreases as the time goes on. At each step, from each \( s \in S \) is generated a new point by using the \( \text{neighbors}(s) \) function and the new \( s' \) solution is evaluated.
Figure 1. Area-Power-Latency Pareto points with respect to the explored configurations

Although in the paper the proposed PSA algorithm has been applied to the STNoC family of topologies, it could be easily applied to a more general case of the customization of other standard topologies or the fully customization of a network topology.

4 Experimental Results

In this section, we discuss the results obtained by applying the proposed methodology to the MPEG4 case study.

Figure 1 shows the explored STNoC configurations (light gray) and the Pareto points (dark gray) from the Area-Power-Latency point of view, respectively in the Latency-Area (Figure 1(a)) and Power-Area (Figure 1(b)) spaces. In both figures, there are some points marked as Pareto that do not seem to be Pareto point. This is only due to the projection of these points from a three dimensional space into a bidimensional space. The results are obtained by exploring $10^5$ different configurations measured in terms of network area, average power consumption and average network latency [9].

5 Conclusions and Future Works

In this paper, an application-specific topology customization flow has been proposed for STNoC. Starting from ring topology (the basis of the STNoC family of topologies) the proposed customization approach is based on the Pareto Simulated Annealing algorithm, trying to add custom links to reduce the network distance between the mapped nodes (increasing the performance) without exceeding the maximum number of predefined STNoC router ports (controlling the area and power overhead).

Acknowledgements

This work has been carried out under the MEDEA+ LoMoSA Project and it is partially funded by the KTI - The Swiss Innovation Promotion Agency - Project Nr. 7945.1 NMPP-NM.

References