JACET

J. ADV COMP ENG TECHNOL, 4(2) Spring 2018

Cost-aware Topology Customization of Mesh-based Networks-on-Chip

Ali Ramezanzad[1], Midia Reshadi[2]
1,2 Department of Computer Engineering Science and Research Branch, Islamic Azad University Tehran, Iran.
(a.ramezanzad@srbiau.ac.ir)

Received (2017-04-27) Accepted (2017-12-14)

Abstract: The small world network idea recently has been introduced in order to optimize the performance of the Networks-on-chip. Based on this method the architecture will be neither fully customized nor completely regular. Results have shown that by using the longrange links which optimized the network power and performance, the area consumption will exceed. We can derive from this that an acceptable bound on the area consumption should be considered. Based on the restriction of a designer, in this paper we want to present a methodology that will automatically optimize an architecture while at the same time considering the area consumption.

Keywords: Networks-on-chip, long-range link insertion, power and area consumption, average latency

How to cite this article:
Ramezanzad A, Reshdi M. Cost-aware Topology Customization of Mesh-based Networks-on-Chip. J. ADV COMP ENG TECHNOL, 4(2) Spring 2018 : 61-68

1. INTRODUCTION

Nowadays, the growing demand for supporting multiple applications causes to use multiple IPs onto the chip. In fact, finding truly scalable communication architecture will be a critical concern. To this end, the Networks-on-Chip (NoC) paradigm has emerged as a promising solution to on-chip communication challenges within the silicon-based electronics [1][3][4]. Many of today's NoC architectures are based on grid-like topologies which are also used in application-specific design [2][3][4]. We can see that in this way, the mapping algorithm plays an important role in improving the overall system cost and performance. Nevertheless, the cost and performance of mapping the applications onto the general-propose platforms are not as optimal as a fully customized topologies [5]. In recent years, small-world network idea [6] [7] has been proposed in order to improve the performance of the NoC with minimum effect on the regularity of the network. Ogras et.al [5] has presented an iterative long-range link insertion algorithm with evaluating the performance and power consumption of the NoC. We have explored this algorithm and simulation results show that by inserting long-range links the average latency and power consumption will be reduced. Moreover, we have also evaluated the area consumption of this approach. The results ensure us that we need to define the limitation regarding the expense of the growing of the area. Built on

This work is licensed under the Creative Commons Attribution 4.0 International Licence.
To view a copy o this licence, visit https://creativecommons.org/licenses/by/4.0/

these assumptions, we present an iterative long-range link insertion algorithm while considering the area cost.

2. RELATED WORKS

In this section, we review a number of related works which could clarify the scheme of the intended methodology.

In [1], a short representation to a contemporary paradigm for systems on chip (SoC) design has been represented. A component-based design methodology will be common among other methods in future in order to support plug-and-play reusability. Authors of [2] have introduced research problems in NoC design which have not been solved yet. In this paper, five main research problems are organized as application characterization, communication paradigm, communication infrastructure, analysis, and solution evaluation. In [3], a 3D modular and scalable NoC architecture which uses robust asynchronous logic is implemented. Authors of [4] have represented a processor array containing 12 memory modules and 1000 independent processors which have been fabricated in 32-nm partially depleted silicon on insulator CMOS. In [5], for synthesizing an architecture which is neither fully customized nor regular, a new method has been introduced. In fact, the communication architecture is a superposition of long-range links and a mesh network. Authors of [6] have introduced the small-world network model of Strogatz and Watts in order to mimic structure of social interaction networks. In [7], the book of DJ Watts which named, "the dynamics of networks between order and randomness", has been introduced. Authors of [8] have represented an accurate and fast NoC area and power model for design space exploration which named: ORION 2.0. In [9], a flow to invent analytical models for NoC switches in terms of power consumption and area occupation, and also strategies for coefficient characterization with different tradeoffs in accuracy and modeling activity has been represented. Authors of [10], based on the combination of both branch and bound concepts and the bandwidth-constrained have represented a performance and power aware mapping mechanism.

3. PROBLEM FORMULATION

Firstly we want to present a few definitions regarding the application and NoC architecture. Fig.1 shows the pictorial description of the problem formulation.

Definition1: 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.

Definition2: The NoC architecture is described by $Arch(T(N,L), R_p, \Omega(C))$, where,

The labeled directed graph *T(N,L)* represent the network topology where each $n_i \in N$ denotes a router 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 : $n_i \in N$

- $np_i(n_i)$ gives a number of the input ports of the router n_i,
- $np_o(n_i)$ gives a number of the output ports in the router n_i,
- $bd_j(n_i)$ gives the buffer depth of a channel j^{th} of the router n_i.

For every $l_{i,j} = (n_i, n_j) \in L$,

$bw_{i,j}$ represents the bandwidth available across the edge $l_{i,j}$,

$h(l_{i,j})$ gives the length of the link $l_{i,j}$. We have defined a basic link unit *bl* identical to the size of the regular links used in the mesh network,

$fw(l_{i,j})$ gives the flit width of the link $l_{i,j}$,

The routing protocol $R_p(n_i, n_{Src}, n_{Dest})$ that determines the outgoing channel at the router n_i for all packets travelling from the source router n_{Src} to the destination router n_{Dest}. In this paper

we have applied a routing algorithm which is proposed in [5].

A mapping function $\Omega : V \rightarrow N$ maps each core $v_i \in V$ of the core graph onto a node $n_i \in N$ of a topology graph such that $map(v_i) = n_j$. We have applied Elixir mapping which has been presented in [10].

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

$$D = \begin{cases} d^k : vl(d^k) = comm_{i,j}, k = 1,2,\dots,|E|, \forall e_{i,j} \in E, \\ with\ source(d^k) = map(v_i), dest(d^k) = map(v_j)x, \end{cases} \quad (1)$$

Then the problem statement is as follows:

The fast and automated algorithm has to customize the mesh network based on the specific application by adding the long-range links with a minimum distortion of regularity plus at the same time considering the area overhead constraints so that average latency and power consumption are minimized.

4. AREA MODEL

Here we want to investigate the Networks-on-chip from an area consumption point of view. In order to achieve this, we use the Orion2.0 the power and the area model [8] which is associated with the Meloni, et al's formulation [9].

In general, the total area consumption of the network can be expressed as:

$$Area_{Network} = Area_{router} + Area_{links} \quad (2)$$

$$Area_{router} = Area_{input\_buffers} + Area_{output\_buffers} + Area_{Arbitration} + Area_{crossbar} \quad (3)$$

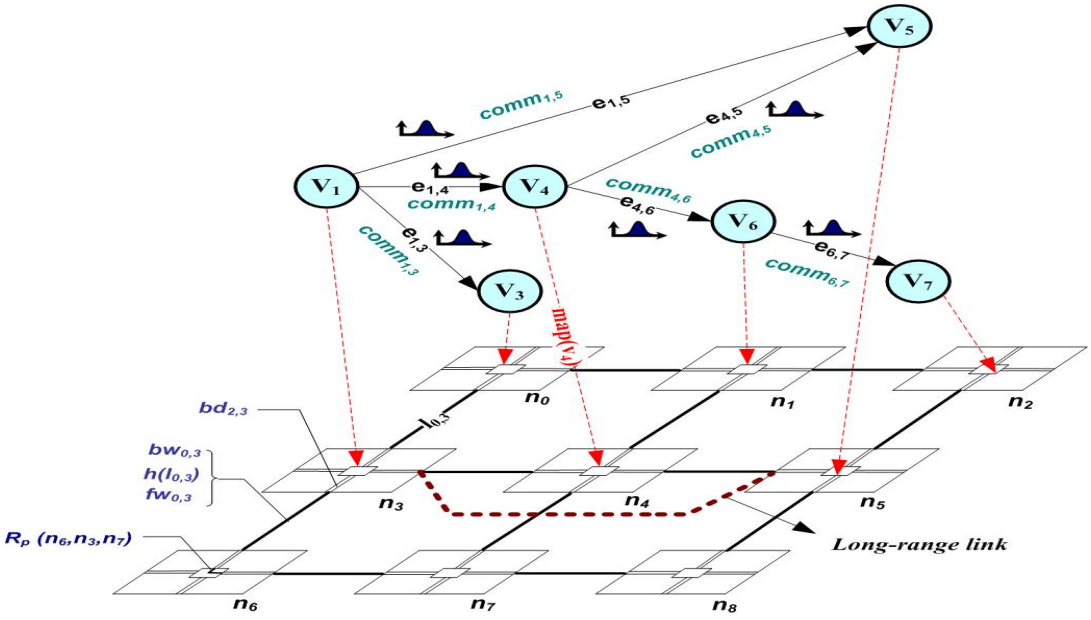

Fig.1. Pictorial description of problem formulation

where $Area_{Network}$ is the total area of the network and both $Area_{router}$ and $Area_{links}$ are the area of routers and links respectively. Modeled on both the Orion area model and [9], the area of the given router can be rendered as the sum of four contributions: *(I)* input buffers, *(II)* output buffers, *(III)* arbitration and flow control logic, *(IV)* crossbar switch. Thus the area of the router can be expressed as:

where $Area_{Arbitration}$, $Area_{crossbar}$, $Area_{input\_buffers}$ and $Area_{output\_buffers}$ are the area consumption of the arbitration unit, crossbar switch, input buffers and output buffers respectively. Each contribution depends on a combination of architectural parameters where have been introduced in section 3. According to [9] the area of the router can be calculated as:

$$Area_{router}=A_1.np_o.fw.bd+A_2.np_i.fw+A_3.np_o.np_i+A_4.fw.np_o.np_i \quad (4)$$

where A_1, A_2, A_3 and A_4 are the coefficients and np_i, np_o, fw, and bd are the number of input ports, number of output ports, flit width, and buffer depth of a given router.

According to the Orion2.0 model, the area of the link can be expressed as the combination of the following architectural parameters:

$$Area_{link}=A'_1.h.fw \quad (5)$$

where A'_1 is the coefficient and h and fw are the link length and the flit width of a given link. According to this, the total area consumption *before* inserting the long-range links can be expressed as:

$$Area_{Network}= A_1.np_o.fw.bd+A_2.np_i.fw+A_3.np_o.np_i+A_4.fw.np_o.np_i+ A'_1.h.fw \quad (6)$$

Identically, the total area consumption *after* the insertion of the long-range links can be calculated as:

$$Area_{Network}= A_1.np_o'.fw.bd+A_2.np_i'.fw+A_3.np_o'.np_i'+A_4.fw.np_o'.np_i'+ A''_1.h'.fw \quad (7)$$

Where h', np_i', np_o' totally represent the link length, the number of input buffer, as well as the number of output buffer after the insertion of the long-range links. In fact adding the long-range links will directly affect the area consumption of the network. Overall, as the area model shows, adding the long-range links has a serious impact on the area penalty.

5. LONG-RANGE LINK INSERTION ALGORITHM

In order to analyze the long-range link insertion algorithm which was proposed by Ogras et. al [5], we have used the SmallNoC[12] plus the worm_sim [13] tools. The results were obtained by defining the different resource constraints. On these results we can see, when the number of inserted long-range links exceeds a specific value, the latency and power consumption decrease abruptly while at the same time, the area consumption will grow linearly. Therefore our main objection is considering the area penalty in the process of adding long-range links. In order to achieve this end, for a given application we have performed the following steps:

- Firstly, the area constraint, the target application and the mesh topology dimension is taken as inputs.
- Secondly, based on the commodity ranking, the long-range links are inserted to the given pairs of tiles with a higher rank within a descending order. This ranking can be calculated as:

$$Ranking(d^k) = vl(d^k)hopcount(source(d^k), dest(d^k)); \forall hopcount(source(d^k), dest(d^k)) \geq 2$$

Where *hopcount(a,b)* is the minimum number of hops between nodes a and b.

- Thirdly, this resulting output network is then evaluated to find out the latency improvement which has been obtained from the previous configuration while considering the area constraint.

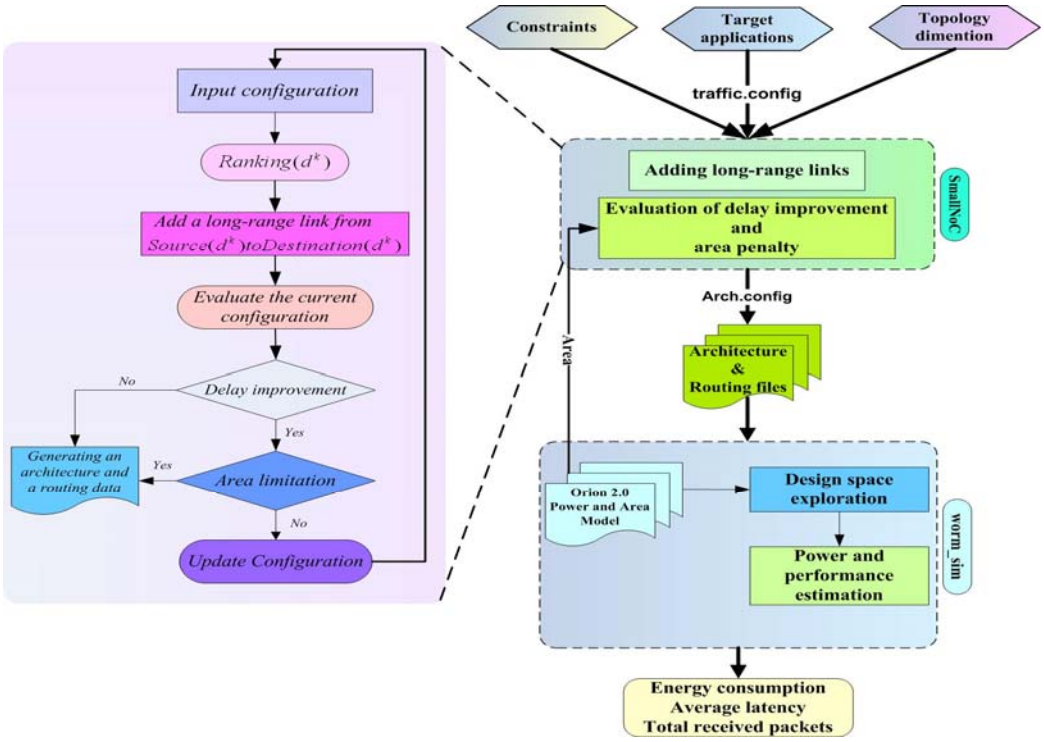

Fig.2. Long-range link insertion algorithm and the simulation toolchain

This process repeats iteratively until the most beneficial links have been founded along with considering the area overhead penalty (Fig. 2). Once this happens, the architecture file and routing data are generated for the precise evaluation. It is important to note that our proposed algorithm has a lower computational complexity (i.e., the link insertion and the routing table generation) because, instead of examining all of the pairs of tiles, the tiles with a higher rank are inspected. In order to verify this idea experimentally, we have modified the SmallNoC tool to implement the proposed algorithm. Furthermore, the Orion 2.0 power and area model [11] library have been added to both the SmallNoC and worm_sim tools so that we can estimate the area consumption of the network. This proposed algorithm gives same long-range links in a lower computational complexity. Fig.3a, Fig.3b, Fig.3e and Fig. 3f totally show the improvement of the average latency, power and area consumption for the MPEG4 and the VOPD video processing applications in 65nm process technology. In order to clear view by considering the area penalty, we have normalized all values in Fig.3d and Fig.3h. Our results show that for the MPEG4 application, growing of the area are 7.8%, 17.9% and 59.3% in conjunction with 1, 2 and 8 links respectively. Correspondingly, with the VOPD, growing of the area are 15.8%, 34.2% and 79.6% while at the same time 2, 4 and 8 links are added respectively. At this point, the designer can chooses more improved architecture based on the area overhead limitation.

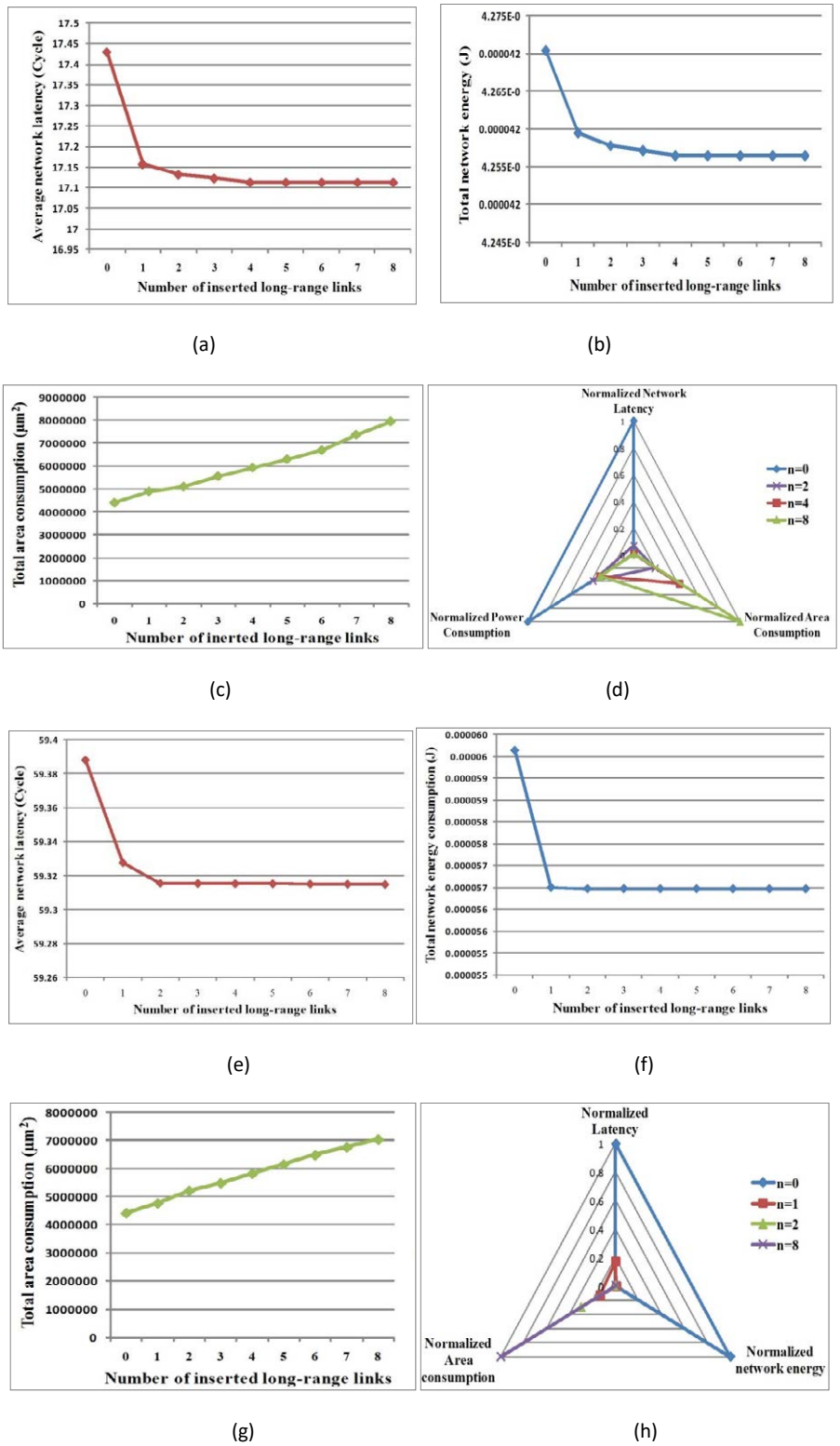

Fig.3(a)AverageNetworklatencyfortheVOPDapplicationwithin65nmprocesstechnology,(b)Networkenergyconsumption for the VOPD application within 65nm process technology, (c) Total area consumption for the VOPD application within 65nm process technology, (d) Normalized Network latency, power, and area consumption for the VOPD application. (e) Average Network latency for the MPEG4 application within 65nm process technology, (f) Network energy consumption for the MPEG4 application within 65nm process technology, (g) Total area consumption for the MPEG4 application within 65nm process technology, (h) Normalized Network latency, power, and area consumption for the MPEG4 application.

Finally, it's worth mentioning, one of the goals for writing this paper was comparing intended algorithm with other related algorithms. Also, we have sought in the latest researches for finding latest basic methods, which is relevant to the proposed technique. But, unfortunately, there is no other related algorithm in order to compare with the intended strategy.

6. CONCLUSION

In this paper we have experimentally shown an area penalty of long-range link insertion idea in the NoC. We have also demonstrated the area model based on the Orion 2.0 and the formulation of [9].As the part of the main contribution, we have proposed the fast and automatic application-specific long-range link insertion algorithm while considering the area consumption. Results show the improvement of the latency and performance based on the definition of the area limitation.

7. REFERENCES

1. Benini, L. and G.J.C.-I.C.S.-. De Micheli, Networks on chips: A new SoC paradigm. 2002. 35(EPFL-ARTICLE-165542): p. 70-78.
2. Marculescu, R., et al., Outstanding research problems in NoC design: system, microarchitecture, and circuit perspectives. 2009. 28(1): p. 3-21.
3. Vivet, P., et al., A $4\times 4\times 2$ Homogeneous Scalable 3D Network-on-Chip Circuit With 326 MFlit/s 0.66 pJ/b Robust and Fault Tolerant Asynchronous 3D Links. 2017. 52(1): p. 33-49.
4. Bohnenstiehl, B., et al., KiloCore: A 32-nm 1000-processor computational array. 2017. 52(4): p. 891-902.
5. Ogras, U.Y. and R.J.I.T.o.v.l.s.i.s. Marculescu, " It's a small world after all": NoC performance optimization via long-range link insertion. 2006. 14(7): p. 693-706.
6. Newman, M.E. and D.J.J.P.r.E. Watts, Scaling and percolation in the small-world network model. 1999. 60(6): p. 7332.
7. Duncan, J.W.J.P.S.i.C., Small worlds: the dynamics of networks between order and randomness. 1999.
8. Kahng, A.B., et al. ORION 2.0: A fast and accurate NoC power and area model for early-stage design space exploration. in Proceedings of the conference on Design, Automation and Test in Europe. 2009. European Design and Automation Association.
9. Meloni, P., et al., Area and power modeling for networks-on-chip with layout awareness. 2007. 2007
10. Reshadi, M., A. Khademzadeh, and A.J.I.E.E. Reza, Elixir: a new bandwidth-constrained mapping for networks-on-chip. 2010. 7(2): p. 73-79.
11. Princeton University, Orion 2.0 software release, Available: http://www.princeton.edu/~peh/orion.html
12. Carnegie Mellon University, SLD: System Level Design Group, SmallNoC: Application-specific long-range link insertion tool, release 1.0, Available: http://www.ece.cmu.edu/~sld/software/SmallNoC.php
13. Carnegie Mellon University, SLD: System Level Design Group, Worm_sim Simulator: a cycle accurate simulator for Networks-on-Chip, release 4.2, Available: http://www.ece.cmu.edu/~sld/software/worm_sim.php