Measurement 74 (2015) 208–220

Contents lists available at ScienceDirect

Measurement

journal homepage: www.elsevier.com/locate/measurement

OPAIC: An optimization technique to improve energy consumption and performance in application specific network on chips

Mehdi Taassori [a], Meysam Taassori [b], Sadegh Niroomand [c,*], Béla Vizvári [c], Sener Uysal [a], Abdollah Hadi-Vencheh [d]

[a] Department of Electrical and Electronic Engineering, Eastern Mediterranean University, Famagusta, Mersin 10, Turkey
[b] Department of Computer Engineering, Sharif University of Technology, Tehran, Iran
[c] Department of Industrial Engineering, Eastern Mediterranean University, Famagusta, Mersin 10, Turkey
[d] Department of Mathematics, Isfahan (Khorasgan) Branch, Islamic Azad University, Isfahan, Iran

ARTICLE INFO

Article history:
Received 30 November 2014
Received in revised form 12 June 2015
Accepted 13 July 2015
Available online 21 July 2015

Keywords:
Network on Chip
Application specific
Energy consumption
Optimization
Mapping
Latency

ABSTRACT

Network on Chip (NoC) is an appropriate and scalable solution for today's System on Chips (SoCs) with the high communication demands. Application specific NoCs is preferable since they can be customized to optimize all requirements of the specific applications. This paper presents an OPtimization technique for Application specifIC NoCs (OPAIC), which aims not only to decrease the energy consumption but also to improve the area of NoCs. OPAIC is composed of three stages to find the optimum NoC; in the first stage, it uses a linearized form of a Quadratic Assignment Problem (QAP) to map tasks on cores to minimize the energy. In the second stage, a Mixed Integer Linear Problem (MILP) is proposed to find the optimum number of the routers for the layout earned in previous stage. Finally, a Greedy Algorithm is applied to optimize the number of virtual channel for every link based on its traffic needs.

1. Introduction

Advances in Very Large Scale Integration (VLSI) technology have led researchers to create a system on a chip which is called System on Chip (SoC). It consists of cores connected to each other by an interconnection network. The interconnection in SoCs has a colossal effect on the energy consumption and this contribution is going to increase in new families of VLSI [1]. Researchers have utilized an infrastructure to improve these interconnections by borrowing the concept of networking from computer network field which is called Network on Chip (NoC) [2]. Many challenges in SoC can be solved by NoC architecture [2]. Although these days some commercial products are using the NoC infrastructure to enjoy its privileges, there are still many challenges in this kind of network which have significant effect on SoCs [3]. As technology shrinks and trend of digital market toward mobile systems such as tablets and smart phones, power consumption has become a key issue in many core chip architecture. By shrinking the transistors in new VLSI families, the power consumed in cores and logics would not be dominant portion of power any more. Instead, the dominant portion is going toward the interconnection between nodes and logics [1].

SoC architecture can be categorized into regular and irregular topologies. A general NoC usually needs to use regular topology since the designers have to assume that

* Corresponding author. Tel.: +90 392 630 1103.
E-mail address: sadegh.niroomand@cc.emu.edu.tr (S. Niroomand).

http://dx.doi.org/10.1016/j.measurement.2015.07.013
0263-2241/

the bandwidth among the different cores is same. Whereas, application specific NoC gives the opportunity to design custom NoCs which are the best choice for our application in terms of power consumption and performance [4–6].

Although, the first priority in nanoscale technologies is energy consumption, this paper focuses on the optimization technique not only to improve the energy consumption but also to boost the performance of the NoCs. The proposed method can be divided into three stages. The objective of the first stage is to construct an optimized mapping of the tasks onto the core regarding to the bandwidth, link length and latency. Linearization technique is used for Quadratic Assignment Problem to obtain optimal layout of NoCs. One of the most important constraints in mapping algorithms is link length which is estimated precisely in our assumed topology named weighted super mesh (WSM). In this topology, all the cores are located like the mesh but there is an extra route in diagonal of cubes in comparison with regular mesh which connects every two cores directly to each other. Every link has a weight estimating the distance between two adjacent cores. In other words, traversing between all cores of a cube costs just one hop with different weight. This new topology provides us with more paths between two cores whose distance would be used while mapping the tasks to the cores.

Number of the routers has a direct impact on the energy consumption, due to this fact the second contribution tries to obtain the optimum number of routers with a new algorithm. Another contribution in this paper is defining a problem to obtain the optimum number of virtual channels. Latency can be affected by the numbers of virtual channels and injection rate. However the number of virtual channels has made a great contribution to the power dissipation [6], the existing researches ignored it and did not consider the optimum number of virtual channels in their problems.

To evaluate the power dissipation, the existing layout optimization methods consider just power of self-capacitance as a power of physical links. Some research activities [7,8] show that in new VLSI technologies coupling capacitances are as significant as an even more dominant rather than self-capacitance; therefore, in contrast to the existing method [5], we consider both the coupling and self-capacitance of the link to determine the link power consumption. Since the size of transistor is shrinking in every family of VLSI technology, the distance between two adjacent wires in chips keep decreasing significantly and in turn the contribution of coupling capacitance is getting more dominant rather than self-capacitance. As a result, ignoring this kind of capacitance in today's families of VLSI has a dramatic misleading in the estimation of power consumption in wires on chips.

2. Literature review

Previous works which applied optimization methods to find best mapping for NoCs could be classified in two separate categories; some of them start with a regular topology such as mesh [9–14] and the other group get started with irregular topology. Murali and De Micheli [10] and Murali et al. [11] introduced different mapping algorithms for mesh based NoCs. Although the proposed method in [10] is fast for mapping, only bandwidth constraint is considered in this method. Murali et al. [11] investigated a mapping and reconfiguration NoC which is able to satisfy communication constraints of different applications. This NoC can be reconfigured based on specifics characteristics of applications. In [12] authors proposed a branch and bound algorithm to map the tasks onto the mesh based NoC architecture and suggested a deadlock free deterministic routing algorithm. Srinivasan and Chatha [13], introduced a mapping and routing algorithm to decrease the energy consumption of mesh based NoCs. In this work, bandwidth and latency are considered as constraints to solve the problem. This technique gave a mesh topology and a communication task graph as inputs to map the cores onto the routers. Ref. [14] presented a discrete particle swarm optimization to map applications onto mesh based Network on Chip architecture.

Obviously, when a mesh topology is considered as a start point of algorithm, the distances between cores are defined based on mesh structure; as a result, unintentionally we are limited by constraints of mesh and we are not able to consider all distances which include an optimum solution. The presented method in this paper does not fall into this category because weighted super mesh (WSM) helps us to consider all possible routes among different cores; therefore, in the proposed method all the existing cases would be regarded and find the optimum solution. Second category considers an irregular topology to start. Some works show that custom topologies outperform the regular ones. Note that irregular structures are more appropriate for application specific systems. Benini [15] describes the challenges and method of design in application specific NoC architecture. According to this research, energy consumption in application specific NoCs can be managed more efficiently than the general purpose applications. A method of core mapping and physical planning along with quality of service has been proposed in [16]. Srinivasan et al. [17] presented a technique to generate the topology of targeted application on chip interconnection architecture. They introduced a method to generate an application specific on-chip interconnection, by using linear programming techniques to map a core to router. In the core to router mapping techniques, topology of NoC is specified before mapping. In spite of this method, OPAIC is not limited by any particular topology in mapping step. Srinivasan et al. [5] suggested a method for application specific NoC architecture by using integer linear programming. The aim of their research is power reduction regarding to the performance. They solve the optimization problem of mapping the core to router and also generate the optimal topology. These researchers show that in terms of the power dissipation and area of NoCs, custom NoC is preferable to regular architectures. Srinivasan et al. [18] investigate an optimal mixed integer linear programming (MILP) regarding to the performance constraints. Chatha et al. [19] presented a method to solve the problem of core to router mapping and generates optimal topology. These researchers divided optimization problem in two sections. Firstly, core to router mapping and secondly, generating

the topology and routing for custom NoC architecture. The first goal in this research is decreasing the power consumption and number of routers reduction is the second objective. In contrast with these algorithms, our method is trying to optimize all specifications of NoC regarding the characteristics of application. A tool for finding the best topology for a specific application is explored in [20]. In [20] the authors proposed a tool to map cores onto the some standard topologies and this tool picks the best option among those predefined topologies based on the constraints of a designer whereas our approach is not bound by any specific topology. OPAIC tries to analyze all distances between the adjacent cores regardless of the topology. In the presented super weighted mesh, cores are located like the mesh but an extra route in diagonal is defined to connect all cores to each other. Extra route enabled us to have a direct connection between each core. Applying this topology provides us with more paths between all the cores in the topology. As a result, the mapping of the tasks to the cores would be based on the accurate distance between the cores which is an important constraint in mapping problem. In [21] a mixed integer linear programming task scheduling and core mapping method for regular and irregular NoC architectures is proposed. The authors presented a graph model to evaluate energy dissipation and latency.

3. Motivation

To achieve the optimum NoC, we need to characterize the power consumption and performance of NoCs to figure out which characteristics are effective in contributing to power dissipation and performance. Apparently, network on chip consist of two main parts, physical links and routers. Since [22] believes that the length of links and bandwidth (generation rate) are two key parameters in power and performance of physical links, these two parameters are evaluated and their impact on power and performance are studied in Part 3.1 and 3.2. On the other hand [23] shows that virtual channel is one of the most important source of power consumption in router of NoC. Due to this fact, the effect of number of virtual channels in performance and power of NoCs are also evaluated.

We characterized this NoC in 65 nm technology. According to the International Technology Roadmap for Semiconductors [1], V_{dd} is equal to 1 V and the clock frequency is set to 500 MHz based on the critical path of the system. For this topology the length of the metal wires is 2 mm. The capacitance of the wire links and coupling capacitance are selected as 0.2 pF/mm and 0.6 pF/mm respectively. The transitions of wires are calculated by Modelsim (Synopsys and Modelsim are registered trademarks). In these simulations, we use a $4 * 4$ mesh topology NoC which has 2 virtual channels per physical channel and using X-Y as routing algorithm. Power of NoC is composed of the power of physical links and routers. Regarding the link power dissipation, we consider the self and coupling capacitances between adjacent wires. We use uniform distribution to send packets between the routers. This study is categorized as follow:

3.1. Power consumption

In this sub section, we study the effect of link length, generation rate, and number of virtual channels in power consumption of NoC.

3.1.1. Link length

Link length has a colossal effect on the link power consumption. The effect of link length on the link power is illustrated in Fig. 1. In this figure the generation rate is 0.035 packets/cycles.

It is obvious that with increasing in link length there is a linear increment in link power dissipation.

3.1.2. Generation rate

In Figs. 2 and 3 the total and link power consumption versus generation rate for a clock cycle of 14 ns are shown, respectively.

According to these simulations it can be concluded that both bandwidth and distance between the tasks affect the power consumption. Therefore, with the best mapping of the tasks to cores subject to bandwidth and distance, the power consumption is minimized.

3.1.3. Number of virtual channels

We investigate the impact of number of virtual channels on power consumption in NoC architecture in the presence of different routing algorithms.

The routing algorithms can fall in three categories; deterministic, partially adaptive and fully adaptive. We use X-Y as an example for deterministic algorithm, North-First, Odd-Even (NF/OE) and Duato as instances for partially and fully adaptive, respectively.

Fig. 4 illustrates power consumption of NoC versus number of virtual channel using X-Y routing algorithm.

Regarding to Fig. 4 the more number of virtual channels the more power would be consumed in NoCs.

Then, we examine the effect of partially adaptive routing such as OE (Odd-Even) and NF (North-First) in the specified NoC. Fig. 5 shows power consumption versus variety of number of virtual channel in OE/NF routing algorithm.

As shown in Fig. 5 by increment the number of virtual channels power will be increased. Finally, Duato algorithm

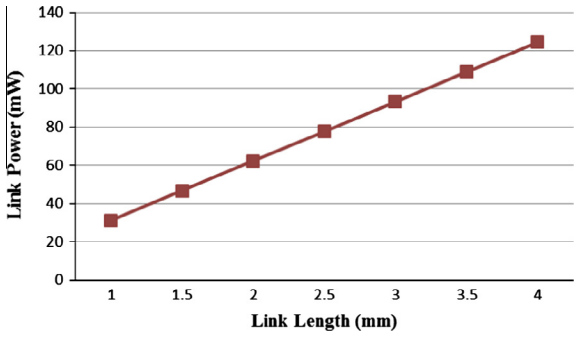

Fig. 1. The effect of link length onlink power consumption.

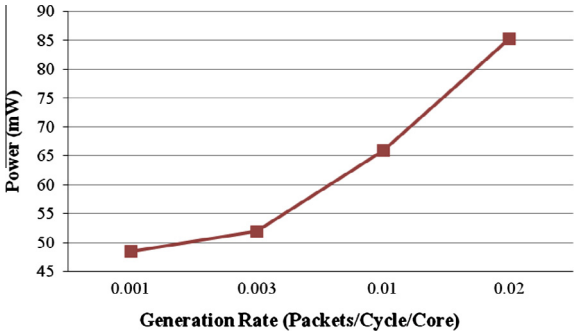

Fig. 2. The effect of generation rate on power consumption.

as an example of fully adaptive routing would be evaluated.

Fig. 6 plots the effect of different number of virtual channels on power consumption in Duato routing algorithm. As it can be observed by using more number of virtual channels the power consumption is increased.

3.2. Latency

Latency is the average delay time taken a packet to traverse between sender and receiver. As mentioned in the previous section, generation rate has an impact on the power consumption. On the other hand there is a relationship between generation rate and latency. Hence, we need to evaluate the effect of the generation rate and number of virtual channels on the latency as a sign of performance of NoCs.

3.2.1. Generation rate

Fig. 7 shows latency versus generation rate in the NoC architecture. The latency is almost constant until the load of network gets heavy enough; after the network is filled with packets in such a way that network is about to be congested, more generation rate leads in more latency and as a result the performance of NoCs goes down. It is clear that the routing algorithm is also effective in when network get congested.

Based on the above remarks, it is concluded that after network has enough load the more generation rate we have, the less performance can be gained. It should be mentioned that this simulation has been done on a

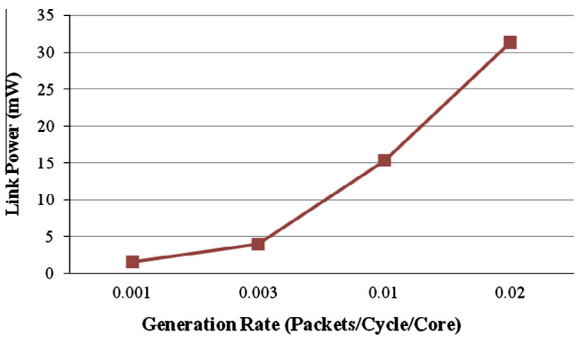

Fig. 3. The effect of generation rate on link power consumption.

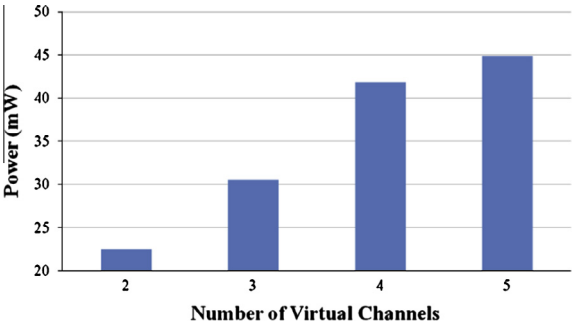

Fig. 4. The effect of variety of number of virtual channels versus power consumption in X-Y routing algorithm.

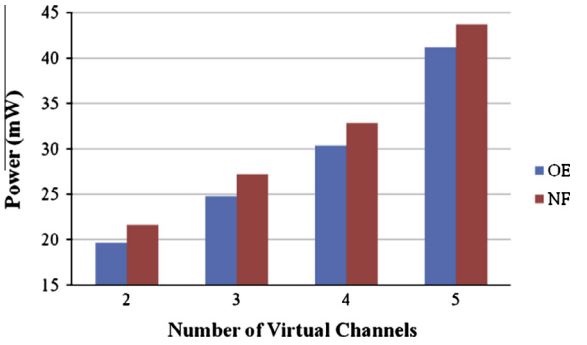

Fig. 5. The effect of variety of number of virtual channels versus power consumption in OE and NF routing algorithm.

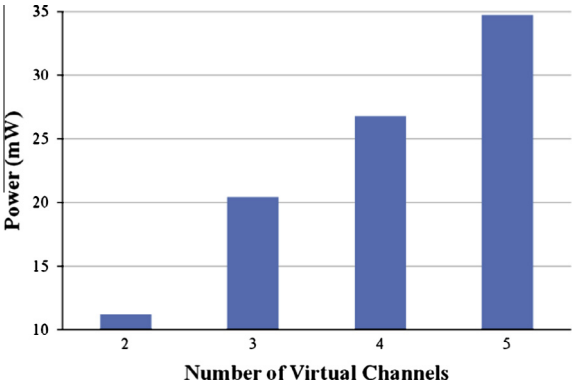

Fig. 6. The effect of variety of number of virtual channels versus power consumption in Duato routing algorithm.

topology having symmetric cores in terms of number of virtual channels. Obviously, the number of virtual channels has a significant impact on these parameters.

3.2.2. Number of virtual channels

We investigate the effect of different number of virtual channels on latency in NoC architecture in the presence of different routing algorithms.

Fig. 8 illustrates latency of NoC versus number of virtual channel using X-Y routing algorithm.

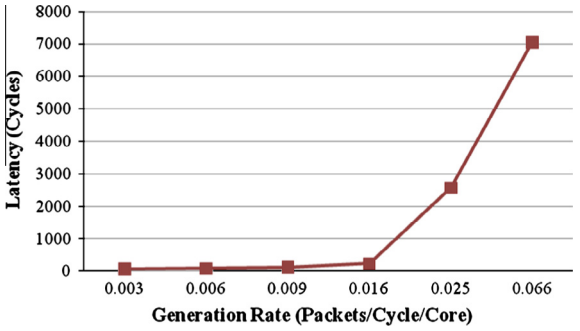

Fig. 7. The effect of generation rate on latency.

Clearly, by increasing the number of virtual channels the latency will be decreased. Putting in other words, with adding the virtual channels packets can traverse more smoothly and channel utilization is increase. Consequently, the performance of the network will be improved in X-Y deterministic routing algorithm.

As a result, when the number of virtual channels changes, there is a tradeoff between power dissipation and performance of NoC with X-Y deterministic routing.

Then, we examine the effect of partially adaptive routing such as OE and NF in the NoC specified above. Fig. 9 shows latency versus variety of number of virtual channel.

As shown in Fig. 9, by increment the number of virtual channels the bandwidth of link can be shared among more messages. This sharing can be done in a non-uniform distribution. Therefore, a message has to pass through several physical links with different degree of multiplexing which causes more latency. Latency will increase. It means that adding the number of virtual channels in partially adaptive cannot be useful nor in power dissipation neither in performance.

Finally, Duato algorithm as an example for fully adaptive routing would be evaluated.

Fig. 10 plots the effect of different number of virtual channels on latency. As we see by using more number of virtual channels, the performance will diminish and power consumption increases as well.

3.2.3. Generation rate and number of virtual channels

We have evaluated the impact of number of virtual channel and generation rate on latency of NoCs separately

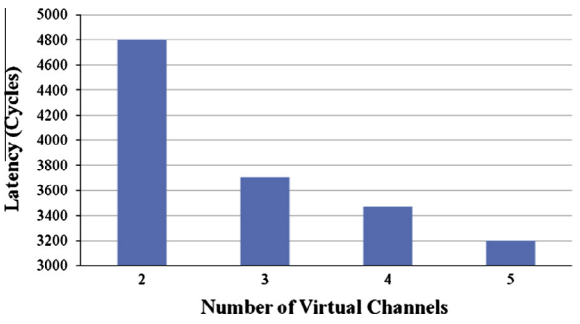

Fig. 8. The effect of variety of number of virtual channels versus latency in X-Y routing algorithm.

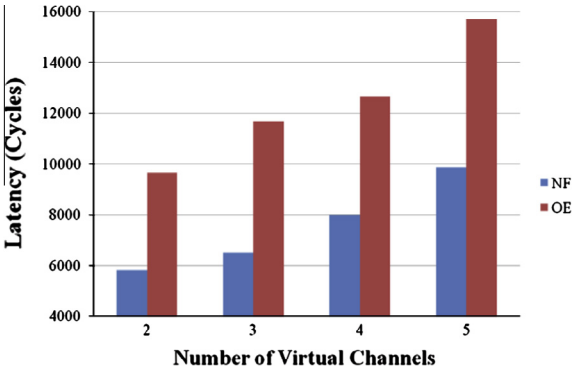

Fig. 9. The effect of variety of number of virtual channels versus latency in NF and OE routing algorithm.

so far. In this sub-section, we aim to study the effect of both these factors on NoCs because the impact of number of virtual channel varies on performance of system when generation rate of network gets changed. Obviously, combination of these factors is also worthy to be investigated. We monitor the effect of generation rate and different number of virtual channel on latency using two different routing algorithms, deterministic and fully adaptive algorithms.

As mentioned, the numbers of virtual channels and latency have direct effect on the power consumption. Splitting the physical line into several virtual channels increases the delay on the system. On the other hand, they can improve the throughput because of minimizing the chance of the congestion.

We study impact of number of virtual channel on the performance by considering the different kinds of routing algorithm, X-Y and Duato routing algorithms. In Figs. 11 and 12, the effect of generation rate and number of virtual channels on the latency in deterministic and fully adaptive routing algorithm is shown, respectively.

As shown in Fig. 11, in the deterministic routing algorithm there is a tradeoff between the throughput and the delay of the router by increasing the number of virtual channels. In fact, when each router has more virtual channels the time which takes packet to pass through that router will be increased. On the other hand, throughput of

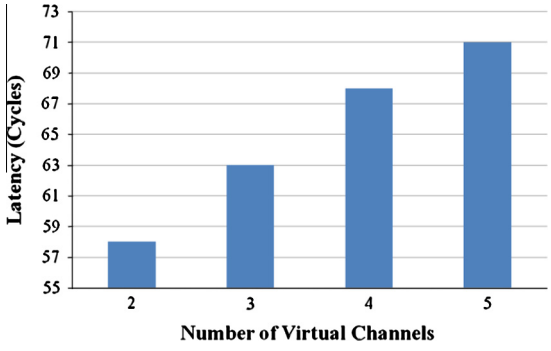

Fig. 10. The effect of variety of number of virtual channels versus latency in Duato routing algorithm.

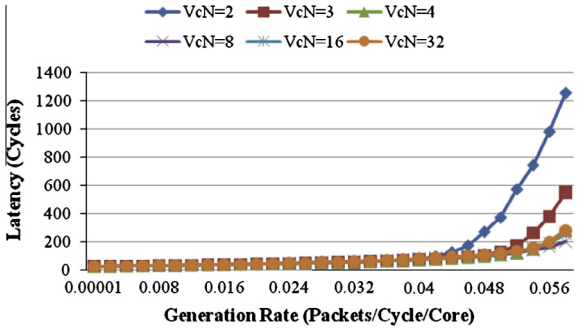

Fig. 11. The effect of generation rate and variety of number of virtual channels versus latency in X-Y routing algorithm.

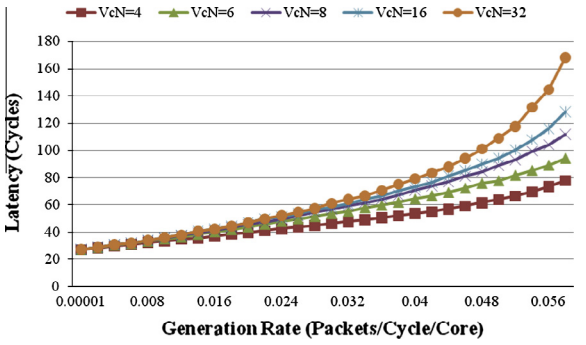

Fig. 12. The effect of generation rate and variety of number of virtual channels versus latency in Duato routing algorithm.

system is improved because in more virtual channels lead in less congestion in network and the packets can traverse more smoothly through network. Fig. 11 illustrates that in the deterministic routing algorithm by increasing the number of virtual channels the throughput increases and the latency slightly goes up. By adding the number of virtual channels messages are allowed to pass more smoothly and it leads higher channel utilization and throughput. On the other hand, increasing the number of virtual channels cannot increase routing flexibility significantly and also it has negative effect on latency. If the number of virtual channels in the deterministic routing algorithm exceeds 8, it cannot be useful any more, neither for performance nor for latency of the NoC. We study the effect of number of virtual channel respecting the generation rate and latency in Duato algorithm in Fig. 12.

According to the Fig. 12, increasing the number of virtual channel cannot be helpful in Duato algorithm. Thus, it can be concluded that there is no benefit in both throughput and delay when the number of virtual channel in fully adaptive routing algorithm increases.

According to the above remarks, it can be concluded that the power consumption depends on the link length, bandwidth between the cores and the number of virtual channels, while performance is affected by bandwidth and the number of virtual channels. Therefore, the power dissipation and performance of NoC can be improved by obtaining the optimum topology based on the distance and bandwidth of the cores with the optimized number of virtual channels.

4. Mathematical modeling

The main goal of the optimization is not only the minimization of the power and energy consumption in custom NoC architecture but also improvement in the overall performance. The proposed method is divided into three sections. First of all, the objective is to extract the optimal layout for application specific design by applying the method of integer programming. In this section we assign the task to core with respect to distance between cores and bandwidth between the tasks. Secondly, we suggest a method to find the optimum number of routers in our layout considering performance of network to get improvement in energy consumed in NoC. The last step is about finding the optimum number of virtual channels.

4.1. Task to core mapping

The input of this step is a communication task graph depicting how tasks are connected to each other as well as the bandwidth of each connection. To find the best mapping for this task graph, Quadratic Assignment Problem (QAP) is used. QAP is a combinatorial optimization problem that assigns a limited number of facilities to a limited number of fixed locations. This method is used to map the tasks to appropriate cores to minimize the total distance of tasks weighted by their related bandwidth. A general QAP denoted by P is defined as follows,

P)

$$\min \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n}\sum_{l=1}^{n} f_{ij} d_{kl} x_{ik} x_{jl} \tag{1}$$

subject to

$$\forall k \in \{1,\ldots,n\} : \sum_{i=1}^{n} x_{ik} = 1 \tag{2}$$

$$\forall i \in \{1,\ldots,n\} : \sum_{k=1}^{n} x_{ik} = 1 \tag{3}$$

$$\forall i,k \in \{1,\ldots,n\} : x_{ik} \in \{0,1\} \tag{4}$$

The variables and parameters of the model are defined according to our mapping problem (with n tasks and n cores) as follows.

- f_{ij} is the bandwidth between tasks i and j,
- d_{kl} is the distance between cores k and l,
- x_{ik} is a binary variable that determines the mapping of task i to core k. The value of the variable is 1 if task i is assigned to core k. 0, otherwise.

The constraints set (2) claims that only one task can be assigned to each core while the constraints set (3) guarantees that only one core can be assigned to each task. The objective function of the model minimizes the total distance of tasks that is weighted by the bandwidth of links. Note that in our benchmarks the Euclidian distance of cores is considered.

Generally, QAP is categorized as NP-hard type problem that cannot be solved optimally in a polynomial CPU running time. Several meta-heuristic algorithms e.g. genetic algorithm, simulated annealing, grasp method, etc. were introduced to solve QAP in order to obtain a good feasible solution near to the optimal [24–26]. On the other hand, the quadratic terms of objective function can be linearized using some well-known linearization techniques. The linearized version of QAP is generally easier to be solved optimally by optimization solvers. The full explanation of these linearization techniques is presented by Chaovalitwongse et al. [27] and He et al. [28]. A linearized version $\overline{P}$ of the above-mentioned problem P is defined as follow,

$\overline{P}$)

$$\min \sum_{u=1}^{N} \theta_u \tag{5}$$

Subject to the constraints which are equivalent to constraints (2) and (3)

$$\forall k \in \{1,\ldots,n\}: \ \sum_{u=1}^{N} \omega_{ku}\varphi_u = 1 \tag{6}$$

$$\forall i \in \{1,\ldots,n\}: \ \sum_{u=1}^{N} \beta_{iu}\varphi_u = 1 \tag{7}$$

$$\forall u \in \{1,\ldots,N\}: \ \left(\sum_{v=1}^{N} \alpha_{uv}\varphi_v\right) - \gamma_u - \theta_u = 0 \tag{8}$$

$$\forall u \in \{1,\ldots,N\}: \ \gamma_u\varphi_u = 0 \tag{9}$$

$$\forall u \in \{1,\ldots,N\}: \ \gamma_u, \theta_u \geqslant 0 \tag{10}$$

$$\forall u \in \{1,\ldots,N\}: \ \varphi_u \in \{0,1\} \tag{11}$$

The variables and parameters of the model $\overline{P}$ are defined according to our mapping problem (with n tasks and n cores) as follow,

- $N = n^2$
- β_{iu} is a coefficient such that,

$$\forall i,u | i \in \{1,\ldots,n\} \& u \in \{1,\ldots,N\}: \quad \beta_{iu} = \begin{cases} 1, & \text{if } (i-1)n+1 \leqslant u \leqslant in \\ 0, & \text{Otherwise} \end{cases}$$

- ω_{ku} is a coefficient such that,

$$\forall k,u | k \in \{1,\ldots,n\} \& u \in \{1,\ldots,N\}: \quad \omega_{ku} = \begin{cases} 1, & \text{if } \exists z \in \{1,\ldots,n\}: \ u = (z-1)n + k \\ 0, & \text{Otherwise} \end{cases}$$

- α_{uv} is a coefficient such that,

$$\forall u,v \in \{1,\ldots,N\}: \ \alpha_{uv} = f_{ij}d_{kl}$$

$$i = \begin{cases} \lfloor u/n \rfloor + 1, & \text{if } u/n \text{ is not integer} \\ u/n, & \text{if } u/n \text{ is integer} \end{cases}$$

$$j = \begin{cases} \lfloor v/n \rfloor + 1, & \text{if } v/n \text{ is not integer} \\ v/n, & \text{if } v/n \text{ is integer} \end{cases}$$

where $\lfloor v/n \rfloor$ shows the lower integer limit of $\frac{v}{n}$.

$$k = \begin{cases} u \bmod n, & \text{if } u/n \text{ is not integer} \\ n, & \text{if } u/n \text{ is integer} \end{cases}$$

$$l = \begin{cases} v \bmod n, & \text{if } v/n \text{ is not integer} \\ n, & \text{if } v/n \text{ is integer} \end{cases}$$

- φ is a binary variable presented by an N dimensional vector such that, ith$n$ variables of the vector shows the assignment of task i ($i \in \{1,\ldots,n\}$), therefore, decision variable x_{ik} in P is exactly equivalent with the decision variable φ_u in $\overline{P}$ such that $u = (i-1)n + k$ or $k = u \bmod n$. This equivalency is guaranteed by constraints (6) and (7).
- γ and θ are continuous nonnegative variables.
- Based on the definition of α and φ:

$$\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n}\sum_{l=1}^{n} f_{ij}d_{kl}x_{ik}x_{jl} = \sum_{u=1}^{N}\sum_{v=1}^{N} \alpha_{uv}\varphi_u\varphi_v$$

Now, the following theorem shows the equivalency of P and $\overline{P}$.

Theorem. *Problems P and $\overline{P}$ are equivalent such that, $\forall i,k \in \{1,\ldots,n\}: \ x^*_{ik}$ is the optimal solution of P if and only if $\forall u \in \{1,\ldots,N\}: \ (\varphi^*_u, \gamma^*_u, \theta^*_u)$ is the optimal solution of $\overline{P}$ under the assumptions $x^*_{ik} = \varphi^*_u$ and $u = (i-1)n + k$.*

Proof. *Necessity:* Let $\forall i,k \in \{1,\ldots,n\}: \ x^*_{ik}$ be an optimal solution of P. As all elements of α_{uv} are nonnegative, it is clear that,

$$\forall u \in \{1,\ldots,N\}: \ \exists \gamma_u, \theta_u | \gamma_u, \quad \theta_u \geqslant 0$$

such that

$$\forall u \in \{1,\ldots,N\}: \ \left(\sum_{v=1}^{N} \alpha_{uv}\varphi_v\right) - \gamma_u - \theta_u = 0 \tag{12}$$

$$\forall u \in \{1,\ldots,N\}: \ \gamma_u\varphi_u = 0 \tag{13}$$

Now based on (12) and (13), $\forall u \in \{1,\ldots,N\}: \ \gamma^*_u, \theta^*_u$ are selected to minimize $\sum_{u=1}^{N} \theta^*_u$. Then, the theorem is proved if we prove that $\forall u \in \{1,\ldots,N\}: \ (\varphi^*_u, \gamma^*_u, \theta^*_u)$ is the optimal solution of $\overline{P}$.

Eq. (12) is multiplied by φ_u. Therefore,

$$\forall u \in \{1,\ldots,N\}: \ \left(\varphi^*_u \sum_{v=1}^{N} \alpha_{uv}\varphi^*_v\right) - \varphi^*_u\gamma^*_u - \varphi^*_u\theta^*_u = 0 \tag{14}$$

The set of Eq. (14) by applying Eq. (13) turns out to,

$$\forall u \in \{1,\ldots,N\}: \ \left(\varphi^*_u \sum_{v=1}^{N} \alpha_{uv}\varphi^*_v\right) = \varphi^*_u\theta^*_u \tag{15}$$

that implies,

$$\sum_{u=1}^{N} \varphi^*_u \left(\sum_{v=1}^{N} \alpha_{uv}\varphi^*_v\right) = \sum_{u=1}^{N} \varphi^*_u\theta^*_u \tag{16}$$

If we prove that $\sum_{u=1}^{N}\varphi_u^*\theta_u^* = \sum_{u=1}^{N}\theta_u^*$, then, $\forall u \in \{1,\ldots,N\}:\ (\varphi_u^*, \gamma_u^*, \theta_u^*)$ is the optimal solution of $\overline{P}$.

To prove this equation it is enough to prove that for any u if $\varphi_u^* = 0$ then $\theta_u^* = 0$. The proof is done by the following contradiction.

Define set A such that $A \subset \{1,\ldots,N\}$. Consider $\forall t \in A:\ \varphi_t^* = 0$ and $\theta_t^* > 0$ such that $\forall u \in \{1,\ldots,N\}:$ γ_u^*, θ_u^* minimizes$\sum_{u=1}^{N}\theta_u^*$. We also define $\tilde{\gamma}$ and $\tilde{\theta}$ as $\forall t \in A:\ \widetilde{\gamma_t} = \gamma_t^* + \theta_t^*, \widetilde{\theta_t} = 0$ such that $\forall v \notin A:\ \widetilde{\gamma_v} = \gamma_v^*$, $\widetilde{\theta_v} = \theta_v^*$. Clearly, $\forall u \in \{1,\ldots,N\}:\ \varphi_u^*, \widetilde{\gamma_u}, \widetilde{\theta_u}$ satisfy (12) and (13) such that,

$$\sum_{u=1}^{N}\widetilde{\theta_u} < \sum_{u=1}^{N}\theta_u^*$$

This contradicts with the assumption that $\forall u \in \{1,\ldots,N\}:$ γ_u^*, θ_u^* minimizes $\sum_{u=1}^{N}\theta_u^*$.

Sufficiency: Assume that the vectors φ^*, γ^*, and θ^* are optimal in problem $\overline{P}$ then in a similar way it can be seen that $\sum_{u=1}^{N}\varphi_u^*\theta_u^* = \sum_{u=1}^{N}\theta_u^*$. If φ^* is not optimal in problem P then there is an even better solution of problem $\overline{P}$ what can be seen again in a similar way, and it is a contradiction.□

In the nonlinear constraint $\forall u \in \{1,\ldots,N\}:\ \gamma_u\varphi_u = 0$, for each u, if $\varphi_u = 1$, then $\gamma_u = 0$, but if $\varphi_u = 0$ the value of γ_u is not depended on this constraint. Based on constraint (8) the upper bound of γ_u is calculated as,

$$M = \max_u \sum_{v=1}^{N}|\alpha_{uv}|$$

Therefore, in $\overline{P}$ the constraint $\forall u \in \{1,\ldots,N\}$: $\gamma_u\varphi_u = 0$ is replaced by the following constraint which has the same restriction.

$$\forall u \in \{1,\ldots,N\}:\ \gamma_u \leqslant M(1-\varphi_u)$$

Finally, $\overline{P}$ is reformulated as,

$$\min \sum_{u=1}^{N}\theta_u \tag{17}$$

subject to

$$\forall k \in \{1,\ldots,n\}:\ \sum_{u=1}^{N}\omega_{ku}\varphi_u = 1 \tag{18}$$

$$\forall i \in \{1,\ldots,n\}:\ \sum_{u=1}^{N}\beta_{iu}\varphi_u = 1 \tag{19}$$

$$\forall u \in \{1,\ldots,N\}:\ \left(\sum_{v=1}^{N}\alpha_{uv}\varphi_v\right) - \gamma_u - \theta_u = 0 \tag{20}$$

$$\forall u \in \{1,\ldots,N\}:\ \gamma_u \leqslant M(1-\varphi_u) \tag{21}$$

$$\forall u \in \{1,\ldots,N\}:\ \gamma_u, \theta_u \geqslant 0 \tag{22}$$

$$\forall u \in \{1,\ldots,N\}:\ \varphi_u \in \{0,1\} \tag{23}$$

As a conclusion, the linearized model of (17)–(23) is solved to assign the tasks to the cores optimally.

4.2. Optimizing the number of routers in NoC design

There are two major sources of power consumption in NoCs, physical links (self and coupling capacitances) and routers; it means that the number of routers plays a significant role in the total power dissipation of the NoC.

Due to this fact, by using an optimization method, the minimum required number of routers can be obtained without a significant degradation in the performance. The input of this step is the optimal layout of NoC obtained in the previous part. In this layout each task is assigned to one core having a router called own router. Besides these own routers, we insert one dummy router in the center of intersections of WSM (shown in Fig. 17). These dummy routers help our algorithm to minimize the number of routers. The router's selection method is based on the gained topology which is found in the mapping step and the connections between the tasks coming from communication trace graph. These dummy routers increase the number of options for Set Covering Problem and as a result they give this algorithm more flexibility and help it to check all the possibilities in the given topology and find the optimum number of routers. The routers are selected according to the layout with optimum mapping and the connections among the tasks. That is, based on the optimum mapping produced in the previous step the algorithm tries to find the optimum number of routers among the dummy and own routers in such a way that all the connections defined in the bandwidth matrix are met. Apparently dummy routers may or may not be selected by the proposed method. The notations of the method are mentioned as follows,

- R is the set of prepositioned routers on the NoC. As mentioned above, for each coreone own router and some extra dummy routers in the spaces between cores are placed.
- C is the set of all connections of NoC. If there are n tasks on the NoC, there will be at most n^2 for directed and $\frac{n(n+1)}{2}$ for bidirectional connections (shown by matrix of bandwidths) note that set C contains the positive bandwidth values.
- T is the set of all tasks which has n entities.

The method follows the steps described below,

Step 1: For each task the set of its related connections from set C is defined by C_i such that $\bigcup_{i\in T}C_i = C$.
Step 2: For each connection of set C, its potential routers from set R is obtained as set R_i such that $\bigcup_{i\in C}R_i \subseteq R$.
Step 3: Using steps 1 and 2, the set of potential routers of each task is collected as RT_i ($\bigcup_{i\in T}RT_i \subseteq R$)
Step 4: For each router of set R, binary variable X is defined. If the value of X is 1, the router should be used on NoC, if it is 0, the router is eliminated from the NoC. Then, the following mathematical model is solved to eliminate the unrequired routers of set R.

$$\min \sum_{j\in R}X_j \tag{24}$$

subject to

$$\forall i \in T:\ \sum_{j\in RT_i}X_j \geqslant 1 \tag{25}$$

$$\forall j \in R:\ X_j \in \{0,1\} \tag{26}$$

The constraint set (25) ensures that at least one router is assigned to each task under minimization of the total number of used routers.

The proposed model finds minimum number of routers for the given layout of NoC based on the initial layout of routers of NoC.

4.3. Optimizing the number of virtual channels

So far an irregular topology with optimum mapping number of routers and optimal mapping has been gained. A simulation is accomplished to figure out the relation of latency over the number of virtual channels in different generation rates in the gained topology which is obtained in the last two steps using the static routing. The result of this simulation, shown in Fig. 13, depicts that by increasing the number of virtual channels, the latency will be increased because the physical line is divided to more sections. On the other hand, the number of received packet per cycle will be improved and in turn the congestion of network would be relieved. As a result, the performance of the network will be improved. It can be concluded that there is a relationship between the number of virtual channel, latency and performance. It is evident that minimizing the number of virtual channels has a direct effect on decreasing the power consumption but the number of received packet per cycle will be damaged. In this section, we find the proper number of the virtual channels in such a way that no degradation happens to the latency. Fig. 13 plots the latency (y-axis) versus the injection rate (x-axis) in terms of variety of number of virtual channels.

A simulation in the gained topology has been conducted to figure out more precisely the relationship between the generation rate and latency considering different number of virtual channels. The result is shown in Table 1. We compare the latency of the obtained NoC topology in the previous part with different generation rate as shown in the first column and variety of virtual channel as is indicated in the first row of Table 1. These generation rates are so close to the real generation rate where our NoC is working. The amount of total latency in the last row of Table 1 is concave-like. The values of Table 1 satisfy the general concave inequality which is $f(\lambda x_1 + (1-\lambda)x_2) \geqslant \lambda f(x_1) + (1-\lambda)f(x_2)$ and $0 < \lambda < 1$ if the amount of generation rate is fixed as 0.001. If the generation rate is fixed then the number of virtual channel is the x-value and the elements in the row belonging to the fixed generation rate value are the function values. Concavity can be observed only on the boundary of the region when generation rate is fixed as 0.001.

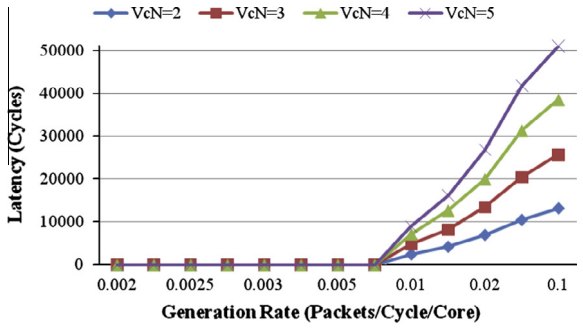

Fig. 13. The effect of generation rate and variety of number of virtual channels versus latency in the gained topology.

According to the above remarks, it can be concluded that concave functions can be used in the regression analysis to obtain a good approximation of the total latency.

To have a function showing the relationship of latency over generation rate and number of virtual channels regression analysis is used which is a well-known statistical method to estimate the relationships of variables. To minimize the approximate error of regression, highest R square is considered which achieved by the following function.

$$Log(latency) = C_0 + C_1\ Log\ (1/GR) Log(VC) + C_2\sqrt{1/GR}\sqrt{VC} + C_3/GR\sqrt{VC} + C_4/GR\ Log(VC) \quad (27)$$

where GR and VC denote the generation rate and virtual channel, respectively. $C0$, $C1$, $C2$, $C3$ and $C4$ are constant coefficients. The value of the R square in the regression statistics is 0.928, meaning that it has excellent quality which makes sure that Eq. (27) can estimate the latency precisely enough if the generation rate of network and number of virtual channel would be chosen in the range shown in Table 1. Latency is the function of generation rate and the number of virtual channels and can be derived as follows:

$$L(GR, VC) = 10^{Log(latency)} \quad (28)$$

where L is the latency and right side of Eq. (28) is the latency gained by Eq. (27). A Greedy method [29–31] is used for assigning the virtual channels to each link. At first, we assign two virtual channels to every links; then, according to Eq. (27), we estimate the latency of each link. Again this stage is done by three virtual channels and the gain of this stage for each link can be calculated as the difference of these two amount of latencies; each link having the maximum gain deserves to get one extra virtual channel. Increasing the number of virtual channel should be continued until the gain gets a negative number. Based to the result of the first step, after finding the optimal layout, there are 15 links on our NoC. The Greedy method says that the next virtual channel is assigned to the link in such a way that the maximum decrease of latency would be achieved. If the number of virtual channels is increased in link j then the gain in latency is:

$$Gain^j = L(GR_j, VC_j) - L(GR_j, VC_j + 1) \quad (29)$$

where $Gain^j$ shows the gain of link j when the number of virtual channel (VC_j) is increased by one. According to the

Table 1
Comparison of latency with different generation rate and the number of virtual channels in the gained topology.

GR	VC = 2	VC = 3	VC = 4	VC = 5	VC = 6
1	14,312	14,153	14,093	14,150	14,045
0.1	12,187	11,976	12,184	12,172	12,234
0.01	275	242	230	229	211
0.001	44	47	49	51	53

```
1.  VC_j = 2
2.  L(GR, VC) =
        10^(C_0 + C_1 .Log(1/GR).Log(VC) + C_2.sqrt(1/GR).sqrt(VC) + C_3 .1/GR.sqrt(VC) + C_4.1/GR.Log(VC))
3.  for j = 1 to n
4.  Gain^j = L(GR_j,VC_j) − L(GR_j,VC_j + 1)
5.  end
6.  while ∃ j | Gain^j > 0
7.      Gain^p = max_j{Gain^j}
8.      for j = 1 to n
9.          if Gain^j = Gain^p then
10.             VC_j = VC_j + 1
11.             Gain^j = L(GR_j,VC_j) − L(GR_j,VC_j + 1)
12.         end
13.     end
14. end
15. Introduce final VC_j
```

Fig. 14. Pseudocode of optimum number of virtual channels algorithm.

greedy principle link p, indicated in Eq. (30), is merit to get one extra virtual channel.

$$Gain^p = \max_j\{Gain^j\} \tag{30}$$

In this study the stopping criterion of the Greedy method is the existence of the negative gain. Other stopping criteria could be the followings: (a) the maximal gain is under a threshold, (b) the maximum number of virtual channels is a priori determined to be achieved.

Fig. 14 gives the pseudo code of optimization number of virtual channels algorithm. We consider that the starting number of virtual channels is two. The gain of each link which is the difference between two latencies is evaluated. According to the greedy method, one virtual channel is assigned to the link with the maximum gain. This procedure will be continued until the existence of the negative gain is achieved.

5. Experimental results

In this section, we analyze the proposed method by applying it on multimedia benchmarks called MPEG4 decoder (MPEG4) and video object plane decoder (VOPD). MPEG4 and VOPD which are categorized into the video processing applications have 12 cores and 13 edges [4]. We compare the results of OPAIC with MOCA [13] and non-optimized NoC topology. The optimization problem is divided into three steps whose results are shown after each step.

5.1. Experimental results for mapping step

In the mapping step of the optimization section, QAP and its linearized version are introduced. QAP is linearized by the previous section. Generally the linearized versions of QAPs are more effective than QAP to be solved, thus, the linearized version of QAP is used to formulate task to core mapping problem. The communication task graph (CTG) of a specific application and weighted super mesh (WSM) topology are inputs and proposed linearized formulation of QAP tries to map the tasks of CTG to the best cores in WSM in such a way that the two tasks with higher bandwidth communication mapped to the closer cores. The number of the cores in WSM is considered to be more than the number of the tasks in CTG, to have more flexibility in the mapping procedure. There is a trade-off in number of these dummy cores; the more number of dummy cores, the more options software has to choose and better result would be gained but run time goes up. As an instance for MPEG4 decoder with 12 tasks we consider $5 * 5$ WSM topology. Therefore, there will be 13 dummy cores. The optimization problem of task to core mapping was coded in Xpress optimizer [32]. Xpress solved the linearized model optimally. Fig. 15 and Table 2 illustrate the communication trace graph and node description of MPEG4 decoder respectively.

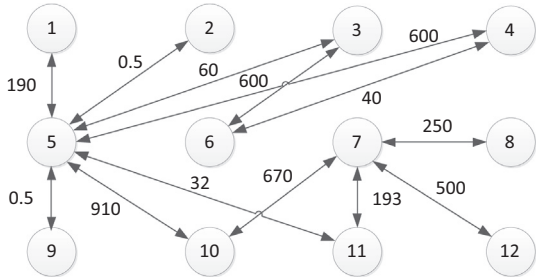

Fig. 15. Communication trace graph for MPEG4 decoder.

Table 2
Node description.

Node	MPEG4 decoder
1	VU
2	AU
3	CPU
4	RAST
5	MEM1
6	MEM2
7	MEM3
8	IDCT
9	DSP
10	UPSP
11	BAB
12	RISC

After applying our method for solving the problem of task to core mapping, the optimum mapping is shown in Fig. 16.

5.2. Experimental results for the least number of routers

In part 4.2 an algorithm based on a linear optimization model is introduced to reduce the number of routers on the NoC. Set of dummy and own routers of the NoC in the optimum layout obtained in the last step are determined and considered as inputs of this step. A linear optimization model minimizes the total number of routers on the NoC conditioning that each task is served by one router of its set of assignable routers. We use Xpress optimizer [32] to solve the problem of minimization of router resources. Fig. 17 illustrates the optimum layout which has been achieved in the first step with all potential routers. Finally, after optimizing the number of routers we obtained the layout depicted in Fig. 18. In Figs. 17 and 18 the darkened squares represent routers and white rectangular represent cores. Due to the key contribution of routers in the power consumption, router reduction has a great impact in decreasing the power dissipation as compared to the other sections of the proposed method.

5.3. Experimental results for the optimum number of virtual channels

As mentioned in part 4.3, the output obtained from mapping and minimizing the number of routers steps are used to simulate the total latency of the NoC over different numbers of virtual channels of each link. The output of this simulation is a function of latency in terms of generation rate and the number of virtual channels. Then we apply a greedy algorithm to this function to find the appropriate number of virtual channels of each link.

In the optimum layout found in last two steps (mapping and minimizing the number of router steps), there are 15 links. These links are depicted in Fig. 19. The results of the greedy algorithm as the final assignment of the virtual channels to each link are given in Tables 3 and 4. Table 3 shows the number of virtual channels on the links between the cores and routers and Table 4 presents the number of virtual channels on the links between the routers. In Fig. 19 the big squares represent cores while the small squares represent routers.

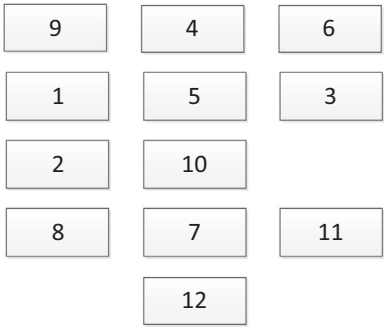

Fig. 16. Layout of MPEG4 decoder after task to core mapping.

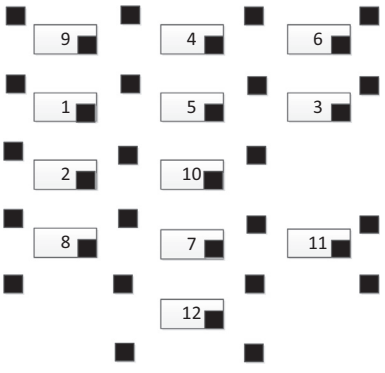

Fig. 17. MPEG4 decoder layout with dummy and own routers.

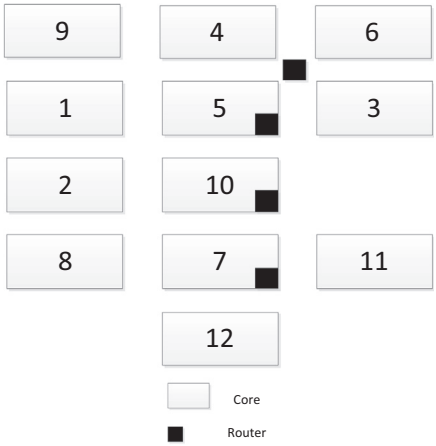

Fig. 18. MPEG4 decoder layout with optimum number of routers.

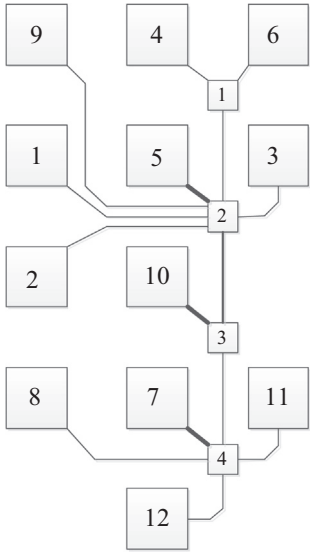

Fig. 19. Layout with different number of virtual channels for MPEG4 decoder.

Table 3
Result of proper number of virtual channels for each core for MPEG4 decoder.

Core	Router			
	1	2	3	4
1		2		
2		2		
3		2		
4	3			
5		8		
6	3			
7				8
8				2
9		2		
10			7	
11				2
12				2

Table 4
Result of proper number of virtual channels for each router for MPEG4 decoder.

Router	Router			
	1	2	3	4
1		3		
2	3		4	
3		4		3
4			3	

Table 5
Results of area.

Benchmark	Area (μm^2)		Ratio
	Non Optimized NoC and MOCA [13]	OPAIC	OPAIC vs Non Optimized NoC and MOCA
MPEG-4	48,834	32,379	0.66
VOPD	48,834	31,550	0.64

5.4. Experimental results of implementation

In our implementation, topology characterized in 65 nm technology, V_{dd} is equal to 1 V and according to the critical path of the system the clock frequency is 500 MHz. Length of the metal wires is 2 mm. The capacitance of the wire links and coupling capacitance are selected as 0.2 pF/mm and 0.6 pF/mm respectively [1]. Modelsim is used to evaluate the number of the transitions of wires under the uniform distribution for sending the packets. Power of physical link and coupling power consumption is considered as a total power of links in NoC.

As mentioned, researchers in [13] proposed a mesh based method subject to the latency by the name of MOCA. In [13] researchers compared their method with two previous state of the art works which are called NMAP [10] and MILP [18]. NMAP does not consider latency constraint and as a result NMAP fails for most of the benchmarks [13]. They compare MILP and MOCA in two cases: with and without latency constraints. The results of [13] show that MOCA can improve power consumption in comparison with MILP in both cases. In this section we compare the proposed method with a non-optimized layout of Network on Chip with the same number of cores and also with MOCA in two cases, with latency and without latency constraints, as best approach comparatively to the others.

Table 5 shows the comparisons of area consumption between OPAIC and the existing technique.

MOCA does not optimize the number of routers; thus it is not able to decrease the area consumption compared with non-optimized NoC architecture. In the other word, non-optimized NoC and MOCA algorithm have same number of routers due to the fact that this algorithm does not have any router reduction method. As mentioned in Table 5, the area consumption of OPAIC, on an average, is 34.5% less than non-optimized NoC architecture and MOCA approach.

Tables 6 and 7 compare the results obtained for OPAIC, Non Optimized NoC architecture and MOCA. In Table 7, the second and third columns present the ratio of the energy consumption between different methods, the fourth and

Table 6
Comparison of OPAIC.

Layout	Energy consumption (μJ)		Latency (nS)		No. of router	
	MPEG-4	VOPD	MPEG-4	VOPD	MPEG-4	VOPD
Non Optimized NoC	924.76	376.03	9192	456	12	12
MOCA with latency [13]	921.76	341.08	918	273	12	12
MOCA without latency [13]	921.38	340.34	935	290	12	12
OPAIC	172.64	81.11	1298	303	4	4

Table 7
Improvement in OPAIC versus other approaches.

Layout	Energy consumption (ratio)		Latency (ratio)		No. of router (ratio)	
	MPEG-4	VOPD	MPEG-4	VOPD	MPEG-4	VOPD
OPAIC Vs. Non Optimized NoC	0.186	0.215	0.14	0.66	0.33	0.33
OPAIC Vs. MOCA with Latency [13]	0.187	0.237	1.41	1.04	0.33	0.33
OPAIC Vs. MOCA without Latency [13]	0.187	0.238	1.38	1.1	0.33	0.33

fifth columns give the proportion of the latency and the two last columns denotes the ratio of number of routers of the OPAIC over the non-optimized NoC architecture and MOCA with and without latency constraints, respectively.

The proposed algorithm intends to optimum power consumption with trivial effect on performance of system in such a way that the energy consumed in NoC would be dwindled. The results show that even though performance of OPAIC compared with previous methods has a trivial increase, due to optimal mapping and number of router reduction, the energy of system is improved drastically. Previous methods consume on average over 78.5% more energy and require 3 times as many routers as OPAIC topology has.

The connections, locations and number of routers in both the non-optimized NoC topology and MOCA impose the data to pass through more routers rather than proposed method, therefore, energy dissipation is increased.

6. Conclusion

This paper addressed the mapping and obtaining the optimum number of routers and virtual channels for application specific NoC architectures. A new method was proposed, which maps the tasks onto the cores and generates an optimum layout such that the energy dissipation is minimized subject to the performance constraints.

The most important effective factors on energy and performance of NoC are used comprehensively; these factors contains bandwidth, link length, latency, number of the routers and number of the virtual channels which are considered as different constraints in OPAIC. We solved the presented problem by dividing it into three parts. Firstly, this paper proposed an optimal algorithm for tasks to core mapping in application specific NoCs. Secondly, we offered a new approach to find the least number of routers in layout gained in first step. Thirdly, a new algorithm to obtain the optimum number of virtual channels for every router with considering the performance of NoC has been presented. In comparison with the previous work, OPAIC is able to reduce on average 78.5% of the energy consumption as well as 34.5% of area of implementation.

Acknowledgements

The authors are indebted to the editors and the referees of the journal for their helpful and constructive comments that improved the quality of this paper.

References

[1] International Technology Roadmap for Semiconductors (ITRS), 2010.
[2] L. Benini, G. De Micheli, Networks on chips: A New SoCParadigm, Computer 35 (1) (2002) 70–78.
[3] G. De Micheli, C. Seiculescu, S. Murali, L. Benini, F. Angiolini, A. Pullini, Networks on Chips: from research to products, in: Proceedings of DAC, 2010, pp. 300–305.
[4] Jalabert, S. Murali, L. Benini, G. De Micheli, ×pipesCompiler: a tool for instantiating application specific Networks on Chip, in: Proceedings of DATE, 2004, pp. 884–889.
[5] K. Srinivasan, K.S. Chatha, G. Konjevod, Linear Programming Based Techniques for Synthesis of Network-on-Chip Architectures, IEEE Transaction On Very Large Scale Integration (VLSI), System 14 (4) (2006) 407–420.
[6] X. Chen, L.-S. Peh, Leakage power modeling and optimization in interconnection networks, in: Proceedings of ISLPED, 2003, pp. 90–95.
[7] M. Taassori, S. Hessabi, Low power encoding in NOCs based on coupling transition avoidance, in: Proceedings of DSD Conferences, 2009, pp. 247–254.
[8] M. Palesi, G. Ascia, F. Fazzino, V. Catania, Data Encoding Schemes in Network on Chip, IEEE Transaction on Computer Aided Design of Integrated Circuit and System 30 (5) (2011) 774–786.
[9] J. Hu, R. Marculescu, Energy and performance aware mapping for regular NoC architectures, IEEE Transaction On Computer-Aided Design of Integrated Circuits and Systems 24 (4) (2005) 551–562.
[10] S. Murali, G. De Micheli, Bandwidth-constrained mapping of cores onto NoC architectures, in: Proceedings of DATE, 2004, pp. 896–901.
[11] S. Murali, M. Coenen, A. Radulescu, K. Goossens, G. De Micheli, A methodology for mapping multiple use-cases onto Networks on Chips, in: Proceedings of DATE, 2006, pp. 118–123.
[12] J. Hu, Radu Marculescu. Exploiting the routing flexibility for energy/performance aware mapping of regular NoC architectures, in proceedings of DATE, 2003, pp. 688–693.
[13] K. Srinivasan, K.S. Chatha, A technique for low energy mapping and routing in Network-on-Chip architectures, Presented at the ISLPED, San Diego, CA, 2005, pp. 387–392.
[14] P.K. Sahu, T. Shah, K. Manna, S. Chattopadhyay, Application Mapping onto Mesh-Based Network-on-Chip Using Discrete Particle Swarm Optimization, IEEE Transaction on Very Large Scale Integration (VLSI), System 22 (2) (2014) 300–312.
[15] L. Benini, Application specific Network-on-Chip design, in: Proceedings of DATE 2006, pp. 491–495.
[16] S. Murali, L. Benini, G. De Micheli, Mapping and physical planning of Networks-on-Chip architectures with quality-of-service guarantees, in: Proceedings of ASPDAC, 2005, pp. 27–32.
[17] K. Srinivasan, K.S. Chatha, G. Konjevod, An automated technique for topology and route generation of application specific on-chip interconnection networks, in: Proceedings of ICCAD, 2005, pp. 231–237.
[18] K. Srinivasan, K.S. Chatha, G. Konjevod, Linear programming based techniques for synthesis of Network-on-Chip architectures, in: Proceedings of ICCD, San Jose, USA, 2004, pp. 422–429.
[19] K.S. Chatha, K. Srinivasan, G. Konjevod, Automated techniques for synthesis of application-specific Network-on-Chip architectures, IEEE Transaction on Computer-Aided Design of Integrated Circuits and Systems 27 (8) (2008) 1425–1438.
[20] S. Murali, G. De Micheli, SUNMAP: a tool for automatic topology selection and generation for NoCs, in: Proceedings of DAC 2004, pp. 914–919.
[21] O. He, S. Dong, W. Jang, J. Bian, D. Pan, UNISM: Unified Scheduling and Mapping for General Network on Chip, IEEE Transaction on Very Large Scale Integration(VLSI), System 20 (8) (2012) 1496–1509.
[22] U.Y. Ogras, P. Bogdan, R. Marculescu, An Analytical Approach for Network on Chip Performance Analysis, IEEE Transaction on Computer-Aided Design of Integrated Circuits and Systems 29 (12) (2010) 2001–2013.
[23] A.E. Kiasari, Z. Lu, A. Jantsch, An Analytical Latency Model for Network on Chip, IEEE Transaction on Very Large Scale Integration (VLSI), System 21 (1) (2013) 113–123.
[24] G. Paul, An efficient implementation of the robust tabu search heuristic for sparse quadratic assignment problems, European Journal of Operational Research (2011) 215–218.
[25] G. Paul, Comparative performance of tabu search and simulated annealing heuristics for the quadratic assignment problem, Operations Research Letters (2010) 577–581.
[26] M. Czapinski, An effective parallel multistart tabu search for quadratic assignment problem on CUDA platform, Journal of Parallel and Distributed Computing (2013) 1461–1468.
[27] W. Chaovalitwongse, P.M. Pardalos, O.A. Prokoyev, A New Linearization Technique for Multi-Quadratic 0–1 Programming Problems, Operations Research Letters 32 (2004) 517–522.
[28] X. He, A. Chen, W. Chaovalitwongse, H. Liu, An Improved Linearization Technique for a Class of Quadratic 0–1 Programming Problems, Optimization Letters 6 (2012) 31–41.
[29] O.F. Alcin, A. Sengur, S. Ghofrani, M.C. Ince, GA-SELM: Greedy algorithms for sparse extreme learning machine, Measurement (2014), http://dx.doi.org/10.1016/j.measurment.04.012.
[30] A.C.K. Vendramin, A. Munaretto, M.R. Delgado, A.C. Viana, GrAnt: Inferring best forwarders from complex networks' dynamics through a greedy ant colony optimization, Computer Networks 56 (3) (2012) 997–1015.
[31] E.G. Talbi, Metaheuristics: From Design to Implementation, Wiley, 2009.
[32] Xpress, <http://www.fico.com/en/Products/DMTools/Pages/FICO-XpressOptimization-Suite.aspx>.