

6th International Conference On Advances In Computing & Communications, ICACC 2016, 6-8  
September 2016, Cochin, India

## Communication centric floorplanning of NoC based System On Chip

Jyothi Thomas John<sup>a</sup>, Nikhil Dahare<sup>a,\*</sup>, Budamagunta Chaithanya<sup>a</sup>, John Reuben<sup>b</sup>

<sup>a</sup>School of Electronics Engineering, VIT University, Vellore, Tamilnadu 632014, India

<sup>b</sup>Assistant Professor, School of Electronics Engineering, VIT University, Vellore, Tamilnadu 632014, India

---

### Abstract

In the past, shared bus based architecture was used as a communication architecture in SoC. They consume more area, power and do not meet the proper bandwidth requirement. Network on chip (NoC) is evolving as a viable communication architecture because they offer better scalability, modularity, design predictability, lower power consumption and shorter latency compared to bus based systems. Application Specific Network on chip (ASNoC) topologies are found to be superior than regular NoC topologies for designing SoC with known communication demands. In this paper, a communication centric floorplanning algorithm is proposed in which communication architecture is synthesized along with the floor plan. This is achieved by grouping IP cores based on their communication requirements in pre floorplanning stage and placing network components judiciously in the post floorplanning stage. Simulated annealing is used as a search engine to obtain the optimal location of IP cores and network components in the floorplan.

© 2016 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

Peer-review under responsibility of the Organizing Committee of ICACC 2016

**Keywords:** Intellectual Property (IP) core; Floorplanning; Simulated Annealing(SA); Network on chip(NoC); Application specific NoC(ASNoC); System-on-chip(SoC); Network Interface; Switch; Network Component (NC).

---

### 1. Introduction

A System-on-Chip(SoC) is an integrated circuit that integrates many standalone VLSI designs(also called the IP cores) to provide complete functionality for the specified application<sup>1</sup>. Various studies have shown that the average number of IP cores in emerging SoC designs varies between 80 and 100<sup>2,3</sup>. These IP cores are pre-designed and pre-verified and made modular to enable reusability. A typical SoC has a main processor, a secondary processor, memory and other function specific cores and interface cores. The function specific cores are the ones which perform arithmetic functions, transform functions (DWT/IDWT, FFT/IFFT), video/graphics processing, error detection/correction, encryption/decryption, etc. The interface cores are used to interface the SoC with the external world. The communication between these cores can be established by standard bus architectures like AMBA, Wishbone etc. But there are disadvantages such as large area, poor performance, delay in the transmission of data and high power dissipation with

---

\* Nikhil Dahare Tel.: +91-9087858767.

E-mail address: [nikdahare09@gmail.com](mailto:nikdahare09@gmail.com)

bus based architecture<sup>2</sup>. Network on chip (NoC) is a communication subsystem between IP cores in a SoC. It uses packet-switched networks for on chip communication and gives notable improvements over bus based architecture. The NoC architecture consists of on-chip switches, network interface, and links(together called Network components (NCs)) on a predefined topology. Using these network components, we develop a network for proper communication between the IP cores in SoCs. In this paper, we synthesize an Application Specific NoC(ASNoc) topology during system level floorplanning which results in a floorplan with minimum area and an efficient on-chip network. Floorplanning determines the optimal locations of the IP cores so as to optimize chip area, link length and power consumption of NoC. It is a crucial step in physical design because it has a profound influence on power consumption of the entire SoC.

The rest of the paper is organized as follows, Section 2 gives the survey on previous works, Section 4 presents the methodology used for implementing the proposed algorithm, Section 5 shows the experimental results and Section 6 concludes the paper.

## 2. Literature Survey

### 2.1. Network on chip(NoC)

To meet the increasing levels of integration, we find that engineers are pushed to design a communication subsystem with increased performance and bandwidth requirements. The fundamental issue in designing a communication subsystem is that the communication and computation for any task must be balanced to meet the performance requirement of the system. Bus based communication architecture results in large area, poor performance, delay in the transmission of data and high power dissipation. Moreover, as the number of cores increase, the efficiency of bus based approach decreases. Hence, a solution for the communication bottleneck is to use a switching network called Network-on-Chip (NoC). It provides high performance and IP cores can be connected to the network in a plug-and-play manner. Thus the design paradigm has shifted from bus based to packetized form of on chip communication. In large-scale SoCs, the power dissipated on the communication infrastructure should be minimized for feasible, reliable and cost-efficient implementations. NoC approach offers greater scalability and lower power consumption. NoCs can be designed as regular topologies(mesh,torus,star,ring,butterfly) or ASNoC topologies. For ASNoC topology design, the design challenges are different from regular topology in terms of core locations, different communication bandwidth requirements and irregular core sizes<sup>4,17,19</sup>. ASNoC are demonstrated to be better than regular NoC architecture in terms of area and performance<sup>6</sup>. There are three important building blocks of NoC. The first component is Link which provides a physical path between the IP cores to implement the communication. Intensive parallel communication is achieved because links can simultaneously operate with multiple packets. The second component is switch which is composed of input ports(from the network interface), output ports(to other switches), a switch matrix that connects the input ports to output ports. The terms router and switch are synonymous in this paper. The third component is network interface(NI) which is used to connect the IP cores to network and translate the core information to a form which the network can propagate. (Fig.1)

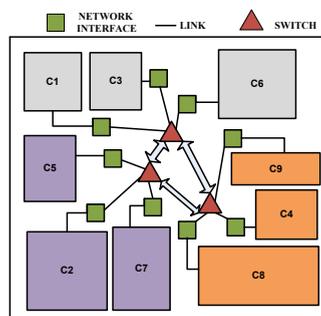


Fig. 1. Application specific Network on chip

## 2.2. Application specific Network on chip (ASNoC)

ASNoC is found to have great significance in designing SoCs. ASNoC can be designed to meet the communication requirements of any design<sup>5</sup>. In application-specific NoC topology, the interconnection between the switches and cores are optimized to match the application traffic patterns. If an application does not require full connectivity between the cores, then the topology is optimized to provide only the required connectivity between the IP cores. The factors like area, power and performance are estimated after the system level floorplanning is done. Therefore, it is necessary to include the information of floorplan for the ASNoC topology. Since the output of system level floorplanning greatly effects the characteristics of the on chip communication architecture, we can design an efficient ASNoC by integrating the design of ASNoC during the floorplanning phase. A few techniques has been recommended for generation ASNoC topologies. The concept of clustering of IP cores based on communication bandwidth is carried out in pre-floorplanning stage as mentioned in<sup>12</sup>. This helps to reduce power dissipation in on chip network. In<sup>17</sup>, the authors proposed a two step algorithm in which mini cut partitioner is used for grouping the IP cores that highly communicate i.e more bandwidth. In<sup>18,19</sup>, the authors use three steps for synthesizing the ASNoC topologies. They assume the floorplan result as the input and find the optimal topology for the existing floorplan. This method does not accommodate change in the location of the cores and hence limits the power reduction that can be achieved in the network. Moreover, the drawbacks of the above method is that the authors can't determine the optimal number of switches for clusters and also can't generate topology for different number of partitions.<sup>7</sup> explains a two phase technique for topology generation, where floorplan is generated in the first phase and placement of switches is done in the second phase. The authors represent the floorplan using CBL (Coin Based Logic)<sup>20</sup>, with some dummy blocks. In<sup>7</sup>, authors show another way to insert the switch for each cluster and use min-cost max flow algorithm for finding shortest path between the switches. The NoC topology in<sup>13</sup> forms the floorplan with clusters in the first phase and then uses ILP (Integer Linear Programming) to place the NCs. ILP with even grid structure is used for the placement of network components. But the insertion of components can overlap with IP cores and depending on the grid size, the run time varies. The path allocation depends on the location of NCs, which limits the quality of the solution. In<sup>11</sup>, the optimal locations for routers and cores are found out using PSO (Particle Swarm Optimization). Hence, the previous research works have not considered the area of NCs during topology generation and also couldn't achieve proper reduction in area and power dissipation. In this work, we try to overcome these drawbacks.

## 3. Floorplanning Algorithm

### 3.1. Problem formulation

In this problem, a set of  $N$  IP cores in an application is considered which is represented in the form of a Core Communication Graph (CCG),  $G(V,E)$ .  $G(V,E)$  is a directed graph where  $V$  represents the set of  $N$  IP cores and  $E$  represents the communication requirements of the cores (an edge from core  $a$  to core  $b$  in CCG represents a physical link). The shape of the core (Width and height) is given as  $(w_i, h_i)$  for  $i=1,2,\dots,N$ . Weight of edge in the CCG represents the bandwidth in Mbps/Kbps. A Switch Communication Graph (SCG) is also represented as  $G(V,E)$  where  $V$  represents the switch for each cluster and  $E$  represents the communication between the switches. A Network Communication Graph (NCG) shows the connection of switch to cores via Network Interface (NI). It transmits the packets from source node to destination node.

### 3.2. Communication centric floorplanning algorithm

During the floorplanning phase, cores are placed in optimal location using the landmark Wong-Liu algorithm<sup>15</sup>. In this classical work, the floorplans are represented by normalized polished expressions (NPE). Let's take an example, say a candidate floorplan represented by NPE,  $1\ 2\ V\ 3\ H\ 4\ 5\ H\ V$ , where  $V$  and  $H$  are the operators and  $1, 2, 3, 4, 5$  are the operands representing the IP cores.  $V$  represents the vertical placement of the cores and  $H$  represents the horizontal placement of the cores. There are three kinds of moves suggested by Wong-Liu algorithm to perturb the NPE of candidate floorplan solution during Simulated Annealing.

1. Swap: Swap two adjacent blocks.  $(1,2 \rightarrow 2,1)$

2. Complement: Complement a chain of operators. (VHV  $\rightarrow$  HVH)
3. Swap Op: Swap a block with its adjacent operator. (2H  $\rightarrow$  H2)

Simulated Annealing is used as a search engine to find the optimized floorplan. It is a probabilistic algorithm to find global minima of an optimization problem whose search space is very large. In pre-floorplaning stage, we considered communication bandwidth as an important factor to design the ASNoC. In the core communication graph (CCG), the pair of IP cores which have large bandwidth are grouped into a cluster. Such a clustering will reduce power dissipation in NoC because the data has to pass through less switches. Clustering will also help to reduce latency of NoC. Once, the set of clusters are obtained, we insert a switch(S) in each cluster as shown in Fig. 4. For each cluster, NPE is generated and Simulated Annealing is used to optimize the area (Level-1). All the clusters are collected together in the floorplan. Assuming each clusters as a black box, NPE is generated and SA (Level-2) is used to optimize the area of the overall floorplan. After obtaining minimum area, we provide spaces between the IP cores in each cluster. Network interface is attached to IP cores in such a way that the manhattan distance between the switch and NI is minimum. Therefore, the two levels of Simulated Annealing results in the reduction of overall area of the floorplan which in turn reduces the total link length required for connecting switches as well as network interface and switch. The adjacent clusters in the floorplan are scattered depending on the dead space availability to further reduce area of the entire floorplan. Fig.4 shows the design flow and Algorithm 4.1 shows the pseudo code of proposed algorithm.

## 4. Methodology

### 4.1. Clustering and Floorplanning

The topology generation issue is considered to be NP hard<sup>16</sup>. The clustering of core communication graph based on communication bandwidth is implemented to reduce the power dissipation. The set of partitions are obtained by using a multilevel algorithm for partitioning graphs<sup>10</sup>. The CCG and the partitioning of a benchmark with 8 modules as mentioned in Table I (PIP) is shown in Fig .2(a) and Fig. 2(b).

Simulated Annealing (SA) is very successful for solving many combinatorial optimization problems in physical design automation. The philosophy behind this technique is to perform a targeted random search instead of exhaustively searching the whole solution space. This is done by perturbing the initial solution to get a neighboring solution. The neighboring solution is accepted if it improves the solution in the required context. Unlike a greedy search, a worst neighboring solution is not always rejected in this technique. Instead, a worst neighboring solution is accepted with some probability in the hope that this decision would lead to an optimum solution in the future. As shown in the algorithm, we use Wong-Liu algorithm for optimal placement where floorplans are represented by normalized polished expressions(NPE). Pseudo code of SA algorithm for Floorplanning is given in Algorithm 4.2.

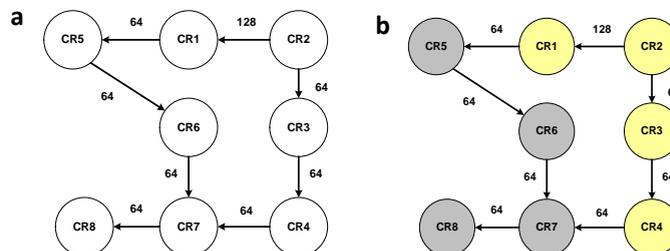


Fig. 2. PIP Benchmark a)CCG; b)Partition of PIP's CCG

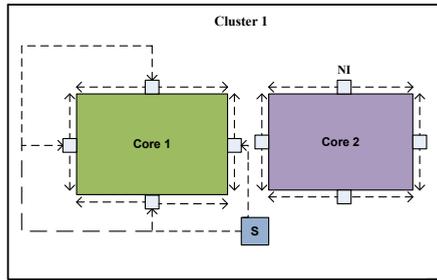


Fig. 3. Placement of NI

**Algorithm 4.1:** PROPOSED ALGORITHM( )

```

N ← Total number of IP cores in CCG (G (V,E));
Use HMETIS22 to obtain the set of clusters(n) based on communication bandwidth
Place the IP cores and switch in the floorplan of each cluster
for i= 1 to n do
Si ← Initial floorplan of clusteri with switch(s) inserted;
SA algorithm (Level-1);
Stotal = combine all the n clusters to form the entire floorplan;
SA algorithm(Stotal) (Level-2);
Placement of NI;
if (No dead space)
    then {Provide spacing (0.2 mm) between IP core in x (& y) direction;

    else {Keep the cores in the same position;
Attach 4 Hypothetical NI to all the 4 sides of the IP cores;
(dotted lines in Fig 3 shows possible locations of NI in each side)
For each cluster;
For each IP core, calculate the Manhattan distance between switch and all possible positions
of NI;
Place the NI at the position at which the Manhattan distance is minimum
Check for dead space across the adjacent cluster
if (dead space)
    then {Shift the IP cores across the cluster boundary so as to reduce the dead space;

    else {Keep the clusters in the same location;
Measure the link length and area of the entire floorplan after the placement of NCs;
    
```

4.2. Placement of Network components

The actual size of the Network components are found to be very small in  $\mu m^2$ <sup>21</sup>. To highlight in the floorplan representation, we assume the size for switch and NI as  $0.3*0.3 \text{ mm}^2$  and  $0.1*0.2 \text{ mm}^2$  respectively. In CCG, the pair of IP cores which has large bandwidth has to be grouped into a cluster. If not possible, it has to be placed in the adjacent cluster. Once, the set of partitions are obtained, we insert a switch in each cluster. For each cluster, NPE is generated and Simulated Annealing is called to obtain a floorplan with minimum area (Level-1). After applying SA on each cluster, we bring all the clusters together in the floorplan. Assuming each cluster as a black box as in Fig .5 (a), we generate NPE and SA is called once again to obtain a floorplan with minimum area (Level-2). Fig .5 (b) shows

the floorplan after Simulated Annealing with switch in PIP benchmark. The modules with similar color belong to a cluster as shown in the example.

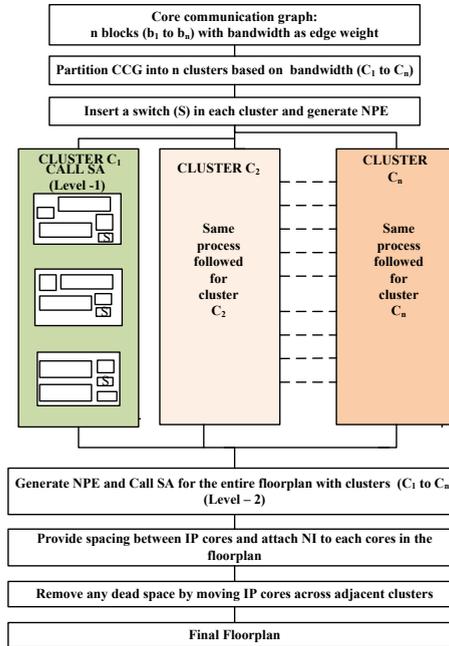


Fig. 4. Proposed algorithm for communication centric floorplanning

**Algorithm 4.2:** SA ALGORITHM( )

```

S ← Initial floorplan; Sbest ← S;
Get an initial temperature T > 0
while (T > 0)
do
for i= 1 to k do
Snew = generateNeighbor(S);
(Perturb the floorplan to get a neighboring Snew from S)
dA = Area(Snew) – Area(S);
if (dA ≤ 0)(Good solution)
then {Sbest = Snew;
else (Worse solution)
then {Sbest = S;
T = r * T; reduce temperature
output (Sbest)

```

After obtaining minimum area, we check for dead space after each IP core in x and y directions in the floorplan. In x direction, if there is dead space between the IP cores, then the cores remains in the same location. If not, we provide a space, say 0.2 mm between IP cores. The same process is repeated along y direction. We could see that the area is increased by a very small percentage. Network interface is attached to IP cores in such a way that the Manhattan distance between the switch and NI is minimum ie NI must be closer to the switch. As shown in Fig .3, position

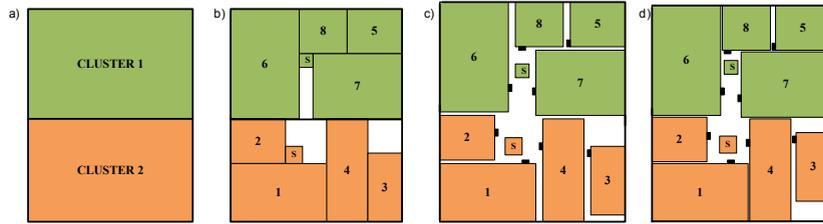


Fig. 5. Floorplan of PIP benchmark a)Floorplan with black box; b) Floorplan with switches; c)Floorplan with switch and NI; d)Further reduction in area

of NI can be any side of the IP core w.r.t the position of switch i.e either left, right, bottom or top. In each cluster, 4 hypothetical NIs are attached on all sides of the core and the Manhattan distance between switch and all possible locations of NI is calculated. The NI is placed at the location at which the Manhattan distance is minimum. Fig .5(c) shows the floorplan after Simulated Annealing with Network components in PIP benchmark. The area of the floorplan can be further reduced by moving the IP cores outside the boundary the cluster by checking the availability of dead space in adjacent clusters. If there is no dead space available, then the clusters will be placed in the same location itself. Fig .5(d) shows the floorplan after dead space reduction.

### 4.3. Communication cost

It plays a major role in the overall power consumption of NoC topologies. There is a trade off between area and communication cost. Hop count is the number of switches a data packet hops to reach the destination core from the source core. As the number of hops are more, the communication cost increases. Our work tries to bring the IP cores close so that the hop count gets reduced for those pairs that communicate more thereby reducing the communication cost. We use the formula for calculating the communication cost as used in<sup>11,14</sup>,  

$$\text{Communication cost} = \sum (\text{No.of hops}) * (\text{Bandwidth})$$
 where the summation is over all pairs of cores in the core communication graph.

Table 1. Power model of switch<sup>21</sup>

Ports (in X out)	2	3	4	5	6	7	8
Bit energy (pJ/bit)	0.22	0.33	0.44	0.55	0.66	0.78	0.90

### 4.4. Power consumption of the NoC

The power consumption for any NoC is given as  $P_{switch} + P_{link}$ , where  $P_{switch}$  is the power consumed by switches and  $P_{link}$  is the power consumed by link.  $P_{switch}$  and  $P_{link}$  are calculated as below,  

$$P_{link} = \sum (\text{Bit energy} * \text{link length between the pair of IP cores} * W), \text{ and}$$

$$P_{switch} = \sum (W * \text{Bit energy}(\text{ corresponding port of switch(s) between the pair of IP cores}))$$
 where W is the bandwidth requirement between the pair of IP cores<sup>7</sup>. According to the formula, the power of switch and link depends on the communication bandwidth and link length between the pair of IP cores. If bandwidth and link length are high, the power dissipation in ASNoC topology will be high.  $P_{switch}$  also depends on the number of switches (i.e the number of clusters) used in the topology. The power dissipation in the link is slightly more than that of the switch. In our method, power dissipated by link is reduced by bringing the pair of IP cores close to each other (with high communication bandwidth) so that the length can be reduced. The link length consists of two parts, link between the switches and the link from core(NI) to switch. Orion 2.0<sup>21</sup> is a power simulator used to find the leakage and dynamic power, and also area models for various architectural components such as switch and network interface

Table 2. Power model of link<sup>21</sup>

Wire length(mm)	1	4	8	12	16
Bit energy (pJ/bit)	0.6	2.4	4.8	7.2	9.6

of NoCs, to enable the rapid performance-power trade-offs at the system and architecture levels. Bit energy for link and switch as given by a power simulator tool called Orion for 180 nm is given in Table 1 and Table 2.

Table 3. Dimensions of IP cores (*mm\*mm*)<sup>11</sup>

Applications	IP cores	Core dimensions ( <i>mm*mm</i> )
PIP	8	2.5*1.5,1*1,1*2,1*2.5,1.5*1,1*2.5,2.5*1.5,1*1
MWD	12	2.5*1,2.5*2.5,3*2.5,2.5*1.5,1*2.5,1.5*1.5,2*1.5,2.5*2.5,2*1.2*1.5,1.5*2, 1.5*2.5
263 ENC MP3	12	2*1,3*1,3*2,1.5*2,1.5*3,1.5*3,2.5*2,2*2.5*2, 2*1.5,1.5*1.5,2*1
MP3 ENC MP3	13	1.5*2.5,1.5*2,1.5*2.5,1*2.5,1.5*2.5,1.5*2,1*3,2*2.5*2.5,2*1,1*2, 2*2, 2*1.5
263 DEC MP3	14	1.5*1.5, 1*1.5, 1.5*1, 2*2.5, 3*1.5, 2*1, 1.5*2,1*1.5,1.5*2, 2.5*1, 2.5*2, 2.5*3, 1.5*3, 2.5*1.5
VOPD	16	1*2.5,3*1,3*3,2*2.5,2*1,2.5*1,1.5*1,2*2.5,2*1.5,2*1.5, 2.5*3, 2*1.5, 1*1 2*1.5, 1*1, 2*1
AUTO INDUSTRY	24	2*2,1.5*1.5,1.5*2,1.5*2,1.5*1.5,1.5*2,1.5*1,1.5*1,1.5*2,2*1.5,2*1.5,1.5*2, 2.5*1.5,1.5*2, 1.5*1.5, 1.5*2,2*1.5,1.5*2,2.5*1.5,2*1.5,2*1.5,1*1.5,1.5*2,1.5*1.5
TELECOM	30	1.5*1,1.5*1.5,2*2,1*1.5,1.5*2,1.5*1.5,2.5*1.5,2*1.5,2*1.5,2.5*1.5,2*1.5,1.5*2, 2*1.5, 2*1.5,1.5*1.5,1*1.5,1.5*1,2*1,2*1,1.5*1.5,2*1.5,1*1.5,2*1.5, 1.5*1,2*2,2*1.5,1*2,1.5*1, 1.5*1.5,1.5*1.5

## 5. Experimental Results

The proposed algorithm was implemented in a high level language. SoC benchmarks available in the paper<sup>9</sup> were used to evaluate the efficiency of the proposed algorithm. The work in<sup>8</sup> suggests that the core dimensions are taken in the range of  $1*1 \text{ mm}^2$  to  $3*3 \text{ mm}^2$ . The aspect ratio varies from 0.5 to 2<sup>11</sup>. Table 3 shows the dimensions of the benchmarks<sup>14</sup>. Prior to floor planning, the ‘Clustering’ of the core communication is implemented by hMETIS software. hMETIS is a software used for partitioning the graph<sup>10</sup>. We provide the vertex and edge weight as the input to the software and the output gives a file that shows which cores belong to which cluster. This helps in reducing the number of switches and also for low power dissipation. Power consumption for each benchmark is as shown in Table 5 for 180nm technology. We ignore the power consumption in NI and IP cores.

The two levels of simulated annealing is applied during the placement of IP cores and during the placement of Network Components. Table 4 shows the reduction of area by an average of 13.46% across 8 benchmarks. The two levels of simulated annealing reduce the overall area of the floorplan. We have the comparison result of area and communication cost given in Table 4 (comparing the results for  $w=0$  where  $w$  is the weight factor used for optimizing communication cost and area in Table 3 of<sup>14</sup>). The proper clustering of IP cores based on communication bandwidth reduce the hop count and in turn reduce the communication cost by an average of 27.07% as shown in Table 4). The graphical comparison between the area and power can be depicted from Fig 6. The link length between the pair of IP cores is also reduced by the placement of IP cores in the same cluster or in the adjacent cluster. The number of

Table 4. Comparison of Area and communication cost

Applications	IP cores	Partitions	Area <sup>14</sup> mm <sup>2</sup>	Area mm <sup>2</sup>	CC <sup>14</sup>	CC
PIP	8	2	23.5	22.00	708.45	704
MWD	12	3	56.04	50.07	1383.04	1376
263 DEC MP3	14	3	59.54	54.02	33.92	23.52
VOPD	16	4	72.86	68.23	5481.31	4262
263 ENC MP3	12	3	55.66	50.99	286.69	230.249
MP3 ENC MP3	13	3	5*8.61	52.58	27.14	16.416
AUTO INDUSTRY	24	5	96.6	92.12	234.49	135
TELECOM	30	5	141.5	128.51	162.35	88

switches depends on the number of clusters used. As the link length and number of switches have reduced, so the power dissipation of the ASNoC topology has reduced by 0.45 % for 4 benchmarks (<sup>7</sup> has used different dimensions for IP cores). The final floor plan obtained using our proposed algorithm for TELECOM and AUTO INDUSTRY benchmarks (which were the biggest benchmarks available) is shown in Fig. 7

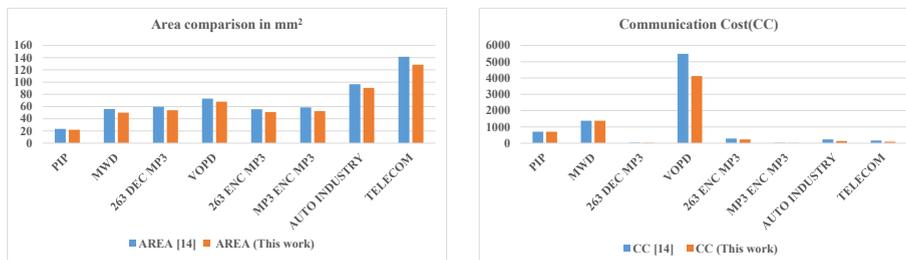


Fig. 6. Comparison of Area and Communication Cost

Table 5. Power consumption comparison

Application	IP cores	Partitions	Power <sup>7</sup> mW	Power mW
MWD	12	3	3.08	3.87
263 DEC MP3	14	3	3.92	4.1
263 ENC MP3	12	3	19.2	17.95
MP3 ENC MP3	13	3	4.4	4.53

## 6. Conclusion

In this paper, we have presented an effective algorithm which integrates floor planning and communication architecture design for a system-on-chip. In the first step, the floorplan is partitioned into ‘clusters’ by grouping the IP cores that have large communication bandwidth. Optimal location for IP cores is found by Simulated Annealing. The second step in the proposed algorithm finds the proper choice of location to place the Network Components within the floorplan. Simulated Annealing plays a major role in reducing the area of the entire floorplan. The proper placement of the network components in the floorplan reduced the link length between the pair of IP cores which has large

communication bandwidth. Experimental results on various benchmarks shows improved solution for area, power and communication cost using our algorithm.

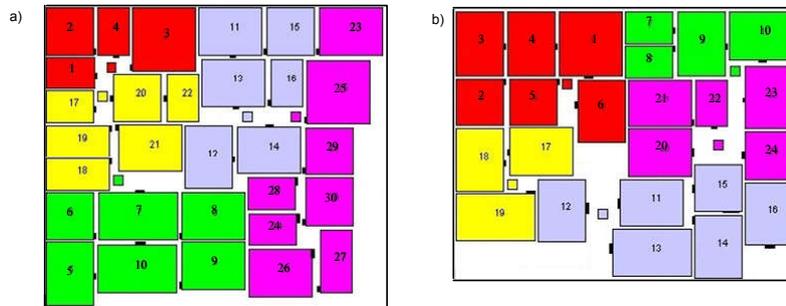


Fig. 7. Generated topology result a) TELECOM Benchmark; b) AUTO INDUSTRY benchmarks

## References

1. R. Rajsuman, *System-on-a-Chip: Design and Test*, Artech House, Inc. Norwood, MA, USA, 2000.
2. J.Isoaho, *Interconnect-Centric Design for Advanced SoC and NoC*, 2004.
3. S.W. Jeong, *IP strategies for SoC design*, Synopsys Journal, 2004.
4. Krishnan Srinivasan , Karam S. Chatha , and Goran Konjevod , *Linear-Programming-Based Techniques for Synthesis of Network-on-Chip Architectures*, IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS, VOL. 14, NO. 4,2006.
5. A. Jalabert , S. Murali , L. Benini and G. De Micheli, *A tool for instantiating application specific networks on chip*, *Design, Automation and Test,XpipesCompiler*, Design, Automation and Test in Europe Conference and Exhibition, 2004.
6. L. Benini , *Application specific NoC design*, Proceeding of the conference on Design, Automation and Test in Europe,2006.
7. Bei Yu , Sheqin Dong , Song Chen and S. Goto, *Floorplanning and topology generation for application-specific Network-on-Chip*, *Design Automation*, Design Automation Conference (ASP-DAC),Asia and South Pacific,2010.
8. E. Salminen , A. Kulmala and T.D. Hmlinen , *Survey of network-on-chip*, Technical report, www.ocpip.org/socketwhitepapers,2008.
9. P.K. Sahu and S. Chattopdhyay, *A survey on application mapping strategies for Network-on-Chip Design*, Journal of Systems Architecture: the EUROMICRO Journal,Elsevier North-Holland, Inc. New York, NY, USA, 2013.
10. Bruce Hendrickson and Robert Leland , *A multilevel algorithm for partitioning graphs*, ACM/IEEE conference on Supercomputing ,ACM New York, NY, USA,1995
11. J. Soumya , Srijan Tiwary and Santanu Chattopadhyay, *Area-performance trade-off in floorplan generation of Application-Specific Network-on-Chip with soft cores*, Journal of Systems Architecture: the EUROMICRO Journal,Elsevier North-Holland, Inc. New York, NY, USA, Volume 61 Issue 1, January 2015.
12. BuWei Zhong , Bo Hung ,Takeshi YOSHIMURA and Sathoshi GOTO, *Floorplanning and topology synthesis for ASNoC*, IEICE Transactions on Fundamentals of Electronics Communications and Computer Sciences, 2013.
13. Wei Zhong, Bei Yu, Song Chen, T. Yoshimura, Sheqin Dong and S. Goto, *Application-specific Network-on-Chip synthesis: cluster generation and network component insertion*, international Symp. Quality Electronic Design (ISQED), 2011.
14. J. Soumya and Santanu Chattopadhyay, *Application-Specific Network-on-Chip synthesis with flexible router Placement*, Journal of Systems Architecture: the EUROMICRO Journal,Elsevier North-Holland, Inc. New York, NY, USA, 2013.
15. D. F. Wong and C. L. Liu , *A new algorithm for floorplan design*, ACM/IEEE Design Automation Conference ,IEEE Press Piscataway, NJ, USA ,1986
16. A. Pinto, L.P. Carloni and A.L. Sangiovanni-Vincentelli ,*Efficient synthesis of networks on chip*, 21st international conference Computer Design, 2003.
17. S.Murali , P.Meloni , F.Angiolini , D.Atienza , S.Carta , L.Benini , G.De Micheli and L. Raffo ,*Designing application-specific networks on chip with floorplan information*, IEEE/ACM International Conference on Computer-Aided Design, 2006.
18. S.Murali , C.Seiculescu , L.Benini and G.De Micheli ,*Synthesis on network on chips for 3D system on chips*, IEEE/ACM Asia and South Pacific Design Automation conference, 2009.
19. C.Seiculescu , S.Murali , L.Benini and G.De Micheli ,*Sunfloor 3D: A tool for network on chip topology synthesis for 3D systems on chips*, IEEE Design Automation and Test in Europe Conference, 2009.
20. X.Hong , G.Huang , Y.Cai , J.Gu , S.Dong and C.Cheng,*Corner block list: An effective and efficient topological representation of non-slicing floorplan*, IEEE/ACM International Conference on Computer-Aided Design, 2000.
21. H. Wang, X. Zhu, L. Peh, S. Malik, *Orion: A Power-Performance Simulator for Interconnection Networks*, IEEE/acm international Symp. on Microarchitecture, 2002.
22. hmetis: <http://glaros.dtc.umn.edu/gkhome/metis/hmetis/overview>