Methods for TSVs Placement in 3D Network-on-Chip

Lev Kurbanov, Nadezhda Matveeva, Elena Suvorova
Saint-Petersburg State University of Aerospace Instrumentation
Saint-Petersburg, Russia
{lev.kurbanov, nadezhda.matveeva}@guap.ru, suvorova@aanet.ru

***Abstract*—In the paper we describe P-medians searching algorithm for Three-dimensional (3D) Network-on-Chip (NoC) design. Modern 3D NoC development is complex and complicated task. Developer should solve different problems: Intellectual Property (IP) blocks placement on the die, organization of vertical links between dies in the 3D stack, energy consumption limitation, system performance improvement. We consider approaches for placement vertical links on the die and suggest a new methods that are based on the P-median problem.**

I. INTRODUCTION

Developing of modern communication system inside chip aid the appearance of Three-dimensional (3D) Network-on-Chip (NoC). It is new trend in field of communication subsystem on an integrated circuit design. 3D NoC is an emerging technology that has the potential to achieve high performance with low power, [1].

The number of IP blocks that make up the chip grows. This is associated with the development of chip manufacturing technologies. The process technology is decreased and there are already devices on the market which are made on 14 nm process technology. For example, Samsung provide Exynos processor from Octa 7 family according to this technology [2]. Intel Core M processors are based on 14 nm process technology. Through three-dimensional tri-gate transistors 2nd generation 14-nanometer technology provides industry-leading performance, energy efficiency, density and value of each transistor and is used to produce a wide range of products: from high-performance computers to systems with low power consumption.

According to the International Technology Roadmap for Semiconductors (ITRS), the number of transistors integrated on a chip continues to grow [3]. Traditional 2D integration can hardly handle such a design complexity in the future. It will result long global wire lengths, causing a high delay, high power consumption and low performance [1]. These parameters are very critical for new advanced systems. Hence, to continue the progress of Moore's law, 3D integration is introduced by stacking multiple dies vertically. The 3D integration becomes a promising alternative because it can potentially offer higher device density, providing shorter wire lengths and faster on-chip communication compared with the 2D integration. A 3D chip contains multiple dies vertically stacked together. There are different approaches for interconnection dies with each other. They are Through-Silicon-Via (TSV), die-to-wafer, flip chip.

We focus on TSV. TSV is a viable solution in building 3D chips by stacking integrated circuit dies (layers) together using vertical interconnects. These interconnects are formed through the silicon die to enable communication among dies.

II. THE COMPLEXITY OF 3D SYSTEM DESIGN

The TSV technology provides new opportunities for increasing the number of vertical interconnection links between dies of chip also it helps to decrease wire length and therefore energy consumption of chip in general. Researchers claim that the key thing for good 3D system development is placement of TSVs. The number and location of TSVs influence quality and robustness of 3D system, [4].

Therefore it is necessary to put interconnection points on chip effectively according to certain rules. Possible solution to this problem can be selected from two options: regular or irregular placement of TSVs, [5]. It is clear that operation characteristics of the whole 3D system depend on data exchange patterns of IP blocks located on different dies. For example, if processors are located on one die and memories are located on another then we should use several TSVs to not have bottleneck in the system.

Also it makes sense to note that TSVs also occupy a certain area on the chip. The TSVs area depends on the process technology of chip. When we use 45 nm process each TSVs occupies 2.47 um × 2.47 um. If TSVs number optimization is not taken into account then total area of all dies of 3D system can get bigger than corresponded area of 2D system.

When moving from 2D technology to 3D technology, some components may require changes in order to use features of 3D technology.

The widely used topology of 2D system is mesh. The structure of switch consists of 5 ports, switch fabric and buffers, (Fig.1). When moving from 2D technology to 3D technology, the number of ports are 7. One port is used for data transmission to upper die. One port is used for interconnection to down die. Increasing of ports leads to complication of switch fabric and arbitration block, [6]. In article [7] the authors provide information about area and energy consumption of switch fabric 5x5. The area is 8523 µm2 and energy consumption is 4.21 mW. When the number of ports equal 7, then the area is 17289 µm2 and energy consumption is 9.41 mW. The process technology is 90 nm, the frequency is 500 MHz. It turns out that the area increases 2 times, and energy consumption - 2.2 times. Therefore if each switch of 3D system has 7 ports then the total system area and energy consumption would be very high. It is additional parameter which affects on the number of TSVs.

The authors of article [8] present methods for allocating and placing a minimal number of vertical links and the corresponding vertical routers to achieve specified performance goals. According to their method a good placement should satisfy the following requirements: 1) each node shall reach at least one TSV within a specified distance; 2) the number of TSV in topology shall be as small as possible; 3) the TSVs shall be distributed uniformly over the topology. Fig. 2(a) shows a worst case placement of TSV in mesh network. In this configuration, the TSV is placed on one side of the chip, causing high delays on the other side of the chip. Fig. 2(b) shows an improved placement. In this configuration, node requires just one hop to access a TSV, which results in a minimal average hop count.

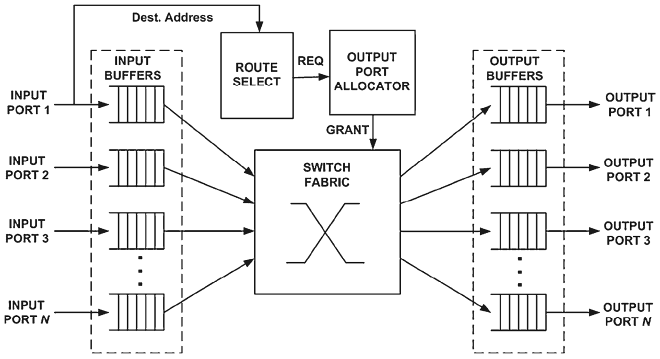

Fig. 1. Architeture of switch

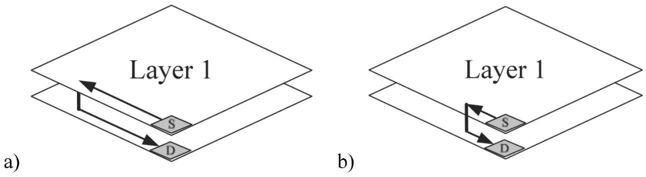

Fig. 2. Comparison TSV placement

The issue of finding an optimal placement of pillars in a network topology can be expressed as an optimization problem. Authors present an Integer Linear Programming (ILP) method that considers the above requirements to find an optimal placement. As shown in the some chars of article, the time required to find an optimal distribution grows fast with increasing topology size. In order to obtain placements for sizes larger than 17x17, researchers explore a Divide and Conquer strategy whereas placements of smaller topologies are used to create placements for large topologies. The main idea is division a large network into several smaller networks and using ILP method for them. The important disadvantage of this approach is non-minimal placement for network after concatenation small network in one big network, [8].

In the book [6] authors describe task of generation of the 3D stack and determination of the communication (routing paths) among dies. Fig. 3 (b,c) shows two alternative partition to die assignments regarding the application's hypergraph depicted in Fig. 3 (a). The vertical lines in this figure correspond to TSVs. As we can conclude, the assignment depicted in Fig. 3 (b) needs 80 TSVs (45 TSVs between Die1 and Die2 and 35 TSVs between Die2 and Die3). On the other hand, the solution depicted in Fig. 3 (c) requires only 55 TSVs (20 TSVs between Die1 and Die2 and 35 TSVs between Die2 and Die3), without affecting either the functionality of target application, or its performance.

In the section 3 we consider our method for placement TSVs on chip which based on P-medians search algorithm.

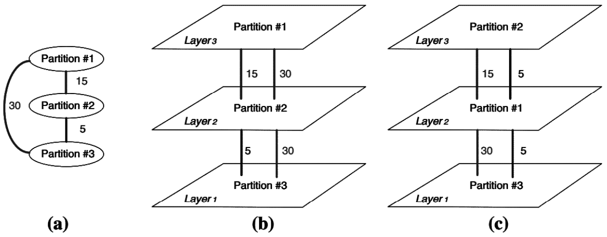

Fig. 3. Two alternative solutions derived (b and c) from partitioning to die assignment for the same application (depicted in (a))

III. Method 1: Placement TSV nodes on the dies with the same topologies

For each specific 3D system designer may independently calculate the required amount of TSV for communication on chip. Designer can use various approaches proposed by scientists from different countries. For example, if you know the area of the 2D chip, it is possible to calculate the maximum number of TSV, which can be used to build 3D (1), where total 3D area will not larger than that 2D area to a certain value (equation 1). Where N_{TSVmax} - the maximum number of TSV, S_{2D} - chip area according to the technology 2D, S_{3D} - estimated area of 3D chip, S_{TSV} - TSV area in according to the process technology. The choice of the number of TSVs is out of scope of this paper.

$$N_{TSVmax} = \lfloor (S_{3D} - S_{2D})/S_{TSV} \rfloor \quad (1)$$

The area of the 2D chip in the same design technology will be less than the total area of the 3D chip, because the organization of vertical interconnections between dies requires more area under the TSV. But the wire length between the components of 2D system is longer than in 3D system (Fig.4).

As mentioned above, in addition to determining the number of interconnections between the dies of 3D system, there is an actual problem of their allocations within the chip. In this paper, we propose the method for searching TSVs placement inside the chip.

It is assumed that the designer defines the number of interconnections which must be placed in a single die 3D chip. The proposed approach is based on the algorithm P-medians search in the graph, [9], [10]. This algorithm has been modified for use in the design of 3D systems.

The basic idea of the algorithm P-medians search in the graph is reduced to the definition of the graph vertices that are on the minimum distance from the rest of the vertices. TSV location searching is necessary to find such placement, where the length of communication lines between the chip blocks and TSV will be as small as possible. In addition to the bond lengths in the design of 3D systems, there must take into account the data transmission characteristics between the blocks, disposed on different dies. Therefore it is necessary to consider the loading of the connections between the dies. Their load must be distributed as evenly as possible. Additionally, you must take into account the factor that the TSV location must be sufficiently distant from each other, so as not to interfere with the transmission of signals.

The mesh is the most common topology of chip. The network topology is represented as a graph. We assume that nodes of the graph are switches and edges of the graph are links between switches. IP block of system are connected to switch.

TSV placement is performed taking into account the topology of the network switches interactions within a single die.

The input data are:

- Submission of the topology switches interaction of the one die 3D assembly in the form of a graph interconnections;
- The number of vertical links (TSV) between adjacent dies, referred to as P;
- The minimum distance, that must be between adjacent points location TSV, referred to as H;
- The deviation of the load TSV, referred to as d.

The main objective of the algorithm to find such nodes in the graph that will match the given constraints of the input data. From all possible solutions will be found a solution that will be characterized by the smallest distance between found nodes and other nodes of the graph. The found nodes will be the TSV location.

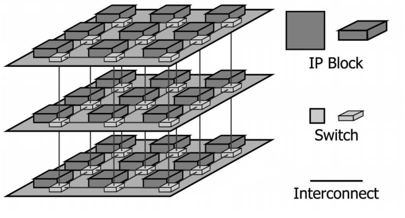

Fig. 4. Structure of 3D system

The algorithm to find the location of nodes consists of several steps:

Algorithm 1 Searching best solution algorithm for one die

```
Step 1. Find the shortest distance between all nodes of the graph using
    the wave algorithm
Step 2. Construct a matrix in which the row corresponds to a node of
    the graph, and the column provides information about the node and its
    distance from the node corresponding to the considered row. The values of
    the columns are sorted in ascending order of distance. (The format is shown
    in the table)
Step 3. Find all the possible combinations of P nodes that are from each
    other by a distance greater than or equal to a given input parameter H.
Step 4. Choose from the list of solutions, that found in Step 3, the solution,
    which will be satisfy evenly connection of nodes to the P-medians with the
    smallest distance to them.
```

A. Construction of a distance matrix between nodes of the graph

When constructing a distance matrix between nodes of the graph used wave algorithm for solving the same problem of searching the shortest route between the nodes of the graph, you can use the Ford-Fulkerson algorithm or other similar algorithms used to find the shortest paths in graphs.

Initially, between all pairs of nodes are determined the shortest distances (Fig. 5). Then, from the obtained data is generated the matrix according to the following structure (Fig. 6). The first column contains a list of all nodes in the remaining columns contain information which indicates how far from the node of the first column to the other nodes of the graph in ascending order. It should be noted that the distance between the nodes is considered to be in the number of ribs, which are between a given pair of nodes.

Node id Node Id	1	2	3
1	0	1	2
2	1	0	1
3	2	1	0

⇦ The shortest distance between all nodes in graph

Fig. 5. The shortest distance matrix

Nodes id	Distance (node id)	
1	1(2)	2(3)
2	1(1)	1(3)
3	1(2)	2(1)

⇦ The sorted distance matrix in ascending

Fig. 6. The shortest distance matrix in ascending

B. The calculation of the distance between the medians of the graph

The vertical interconnections between dies of 3D assembly should not be placed closely to each other. This is necessary in view of the large heat release vertical links. Therefore it is necessary to place them at some distance from each other. In this approach, it is assumed that this distance is specified by the designer in the input data (parameter H).

It should be noted that the distance between the TSV locations is calculated according to the rule for calculating the distance Chebyshev. Chebyshev distance (L) is the maximum of modulus of the difference components of two vectors, the formula 2.

$$L(\vec{x},\vec{y}) = \max_{i=1,\dots,n}|x_i - y_i| \qquad (2)$$

Since we are working with the Mesh topology is convenient to represent nodes in the network as elements of a two-dimensional array. Each node has an index indicating the row [x] and column [y], in which the node is located. As the index value of x and y coordinates can be correlated with the switch position next to which will be located vertical line connections between the dies.

The condition for verifying the constraint on the distance between the medians is displayed in the following rule:

$$(|x_k - x_s| \ge H)\text{or}(|y_k - y_s| \ge H), \qquad (3)$$

Where x_k, x_s - index indicating a row of k-th and s-th node respectively; y_k, y_s - index indicating a column of k-th and s-th node respectively. If the condition is satisfied, then the k-th and the s-th nodes can be medians.

Thus, for each pair of median nodes is checked the restriction on the distance between them. For example, for the node with coordinates (3;3) and the limitation of distance H=2 to any other median node, you can select all the nodes that satisfy this constraint. Fig. 7 selected nodes that satisfy the constraint H=2 for a node with coordinates (3;3).

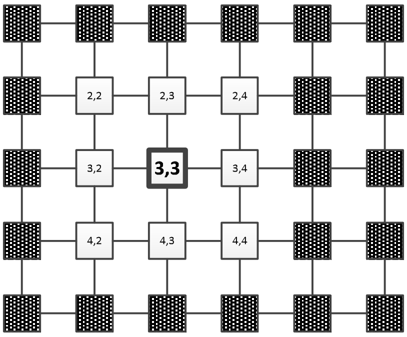

Fig. 7. Nodes that satisfy the constraint Dp=2 for a node with coordinates (3;3) are colored by black.

The following figures (Fig. 8) show examples not satisfying the constraint H=2 in the case of the three medians.

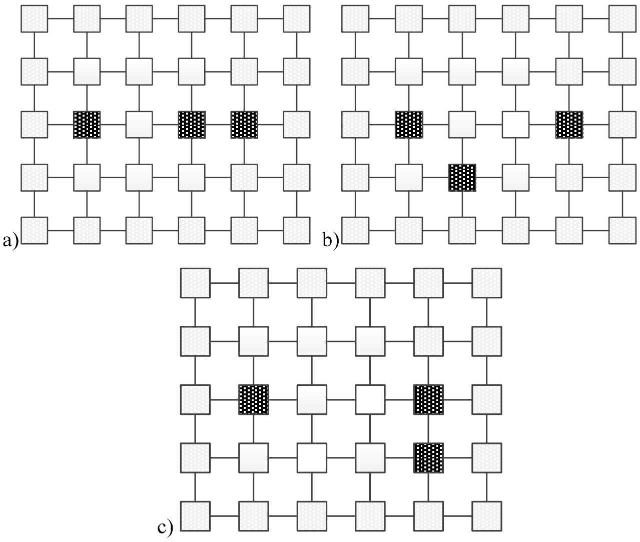

Fig. 8. Examples not satisfying the condition H=2 in the case of the three medians. a) fails the condition by columns; b) fails the condition by columns and by rows; c) fails the condition by the rows.

C. Evenly distribution of serviced nodes between the medians

Evenly distribution of serviced nodes between the medians is necessary for evenly loading of the vertical links. The parameter d specifies the maximum allowable deviation from the average number of serviced nodes. This limitation can be expressed by the following inequality:

$$\frac{NN-P}{P} - d \le NAN \le \frac{NN-P}{P} + d, \qquad (4)$$

Where NN - number of nodes in die; P – the number of medians (nodes with TSV), NAN – the number attachable nodes each median.

For example, the network size 6x5 contains 30 nodes (Fig 9). You must place 3 TSV, the parameter d=1. The number of attachable nodes to each median must be in the range 8≤NAN≤10.

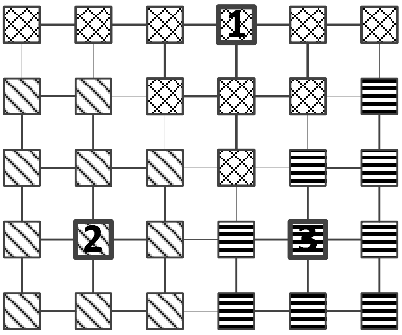

Fig. 9. Solution example for P=3, d=1. NAN_1=9; NAN_2=10; NAN_3=8

D. An example of work of the algorithm

Consider the example of the algorithm on a graph, which is derived from the 3x3 Mesh topology.

Consider the following example (Fig. 10).

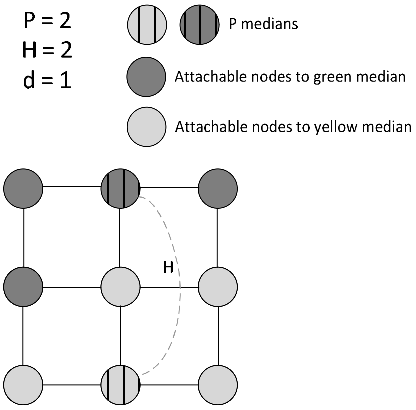
P = 2
H = 2
d = 1
P medians
Attachable nodes to green median
Attachable nodes to yellow median
H

Fig. 10. The example of a graph with found P-medians

For this graph you need to find the 2 nodes that can be placed between each other no closer than at a Chebyshev distance of 2 hops.

Total in graph is 9 nodes, 2 of them will be selected as the P-median, the remaining 7 must be attached to each median, so that the distance between median and the attachable node is minimal possible. Unmedian nodes must as evenly as possible attached to the median nodes and their attachment does not go beyond the limits indicated in the input parameter d. It turns out that attachment in our case is 3.5 nodes to each median. It is impossible, since the nodes are indivisible. Hence three nodes will be attached to a first median, and four nodes will be attached to the second median. The shortest distance between medians and attached nodes is 2. To find solutions with a smaller shortest distance is impossible for this input data.

As a result of the algorithm will be found the graph nodes that have the smallest maximum distance to other nodes and at the same time themselves be positioned with respect to each other by a distance not less than the designated input value; with the remaining nodes of the graph which are will evenly attach to medians.

IV. METHOD 2: PLACEMENT TSV NODES ON THE DIES WITH DIFFERENT TOPOLOGIES

In the 3rd section of the paper we reviewed case, when all dies in the 3D stack had equivalent topologies and network configures. Therefore, the placement task boiled down to p-median problem in 2D for one die, and getting solutions fit for all remaining dies.

However, in practice, usually 3D system is heterogeneous. It means that dies have different topologies. Therefore, solution for one die may not be suitable for another die. Further we consider the task of finding the nodes which are the most suitable for TSVs connections inside 3D system where the network topologies in each die are different, the number of nodes are same on all dies, and TSVs must be allocated in the same place in die of 3D stack.

Importantly to notice, the physical nodes location in the die is static, i.e. all nodes in the die have their precisely defined position and are arranged above one another relative to the entire stack. In this paper, we consider the case, where nodes are placed on a die in the form of a matrix. The Fig.11 shows an example of how an arbitrary topology will look at a matrix arrangement of nodes in the die.

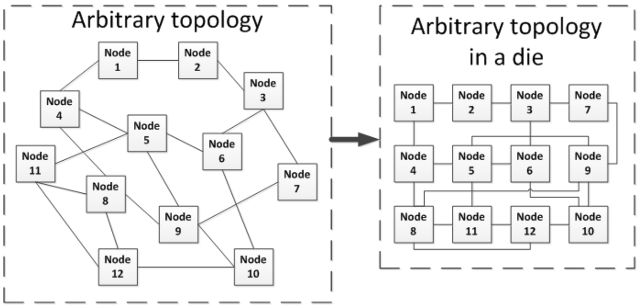

Fig. 11. Example arbitrary topology in a die

Each node in a die has its coordinates. The Fig.12 shows an example of nodes with three coordinates (x, y, z), where x, y - a node position in a die, z - die number in which the node is.

Recall that TSVs must be at a certain distance from each other in the die, as there is the problem of heat dissipation. Standing too close TSV to each other can lead to overheating and failure of the entire system. We propose to solve this problem by placing the TSVs on the die in the matrix form. In other words, it suffices to know the minimum allowable Chebyshev distance H (2) between TSVs and compare their position in the die according to the rule (3).

Nodes number must be the same in all dies.

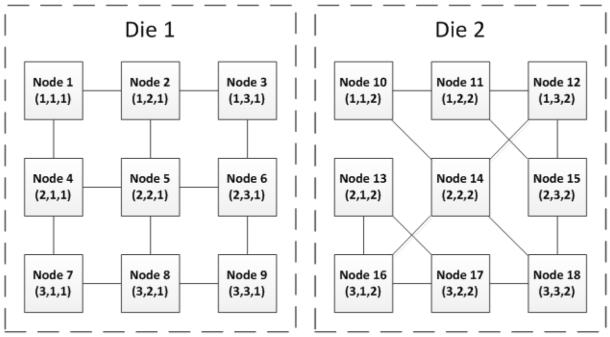

Fig 12. Example of specifying the coordinates of the nodes in the die

For comfortable reading next algorithms we offer use Table I with notations and their descriptions.

TABLE I. USED NOTIFICATION AND DESCRIPTION

Notification	**Description**
$\mathbb{M} = \{1..N\}$	Set of medians' numbers, where N - quantity of medians
$\mathbb{D} = \{1..K\}$	Set of dies' numbers, where K - quantity of dies
$G = \langle \mathbb{V}, \mathbb{E} \rangle$	Chip's graph
$\mathbb{V} = \{v_q : 1 \leq q \leq CNC\}$	Set of the nodes in chip, where CNC - quantity of nodes in chip
$\mathbb{E} = \{e(v_s, v_f) : v_s, v_f \in \mathbb{V} \wedge s \neq f\}$	Set of the interconnections in chip, where $e(v_s, v_f)$ - link between node v_s and node v_f
$coord =< cx, cy, cz >$ $cx, cy, cz : \mathbb{M} \rightarrow \mathbb{Z}_0^+$	Coordinate's structure, where "cx,cy" - functions, which return node's "v" coordinates on the die, "cz" returns die's number, where node "v" is situated
$\mathbb{NS}_j = \{v_q : 1 \leq q \leq CND_j \wedge cz(v_q) = j \wedge v_q \in \mathbb{V}\}$	Set of nodes on the j-th die, where CND_j - count of nodes on j-th die
$VRP =< \mathbb{M}, \mathbb{DVRP}, dp, lp >$	Structure of solution for full system
$\mathbb{VRP} = \{VRP\}$	Set of solutions for full system
$\mathbb{DVRP} = \{VR_i : 1 \leq i \leq N\}$	Set of vertical regions for one partition
$dp \in \mathbb{N}$	Sum of diameters of all vertical regions
$lp \in \mathbb{N}$	Load difference of all vertical regions
$VR_i =< \mathbb{VRV}_i, \mathbb{VRE}_i >$	Vertical region for i-th median
$\mathbb{VRV}_i = \cup_{j=1}^{K} \mathbb{FNS}_{i,j}$	Set of nodes in the i-th vertical region
$\mathbb{FNS}_{i,j} = \{v_q : 1 \leq q \wedge v_q \in \mathbb{NS}_j\}$, $\forall i \neq q \; \mathbb{FNS}_{i,j} \cap \mathbb{FNS}_{q,j} = \emptyset$, $\cup_{i=1}^{N} \mathbb{FNS}_{i,j} = \mathbb{NS}_j$	Set of nodes, which are attached to the i-th median on the j-th die
$\mathbb{VRE}_i = \{e(v, w) : v, w \in \mathbb{VRV}_i \wedge e(v, w) \in \mathbb{E}\}$	Set of links in the i-th vertical region
$\mathbb{BSS}_j = \{S_j\}$	Set of best solutions for j-th die
$S_j =< \mathbb{M}, coord, \{\mathbb{FNS}_{i,j} : 1 \leq i \leq N\} >$	Solution's structure on j-th die
Formulas	
$\mathbb{V} = \cup_{j=1}^{K} \mathbb{NS}_j$ $\mathbb{NS}_j = \cup_{i=1}^{N} \mathbb{FNS}_{i,j}$	

To solve TSV placement problem for dies with different topology we propose following algorithm:

Algorithm 2 Searching best solution algorithm for full system

```
function SEARCHBESTSOLUTIONFORFULLSYSTEM
    for all $die \in \mathbb{D}$ do
        $\mathbb{BSS} \leftarrow BestSolutionsForDie(die)$
        for all $S_{die} \in \mathbb{BSS}$ do                    ▷ The mapping stage
            $VRP \leftarrow BuildVerticalRegions(S_{die}, die)$
            $\mathbb{VRP} \leftarrow \mathbb{VRP} \cup \{VRP\}$
        end for
    end for
    $\mathbb{VRP} \leftarrow WithMinDiamSum(\mathbb{VRP})$
    $\mathbb{VRP} \leftarrow WithMinLoadDiff(\mathbb{VRP})$
    return $\mathbb{VRP}$
end function
```

Algorithm for a function “BestSolutionForDie(die)” in line 3 has the same logic with Algorithm 1.

In this algorithm the mapping is to get an idea as the best solution of one die will have a view on the other dies.

We introduce the term vertical region. Under the vertical region we mean subgraph of the entire network, which comprises on each die one node with TSVs and nodes attached to it. The Fig.13 shows an example of a vertical die. Bold lines highlight the first vertical region and dashed - the second vertical region.

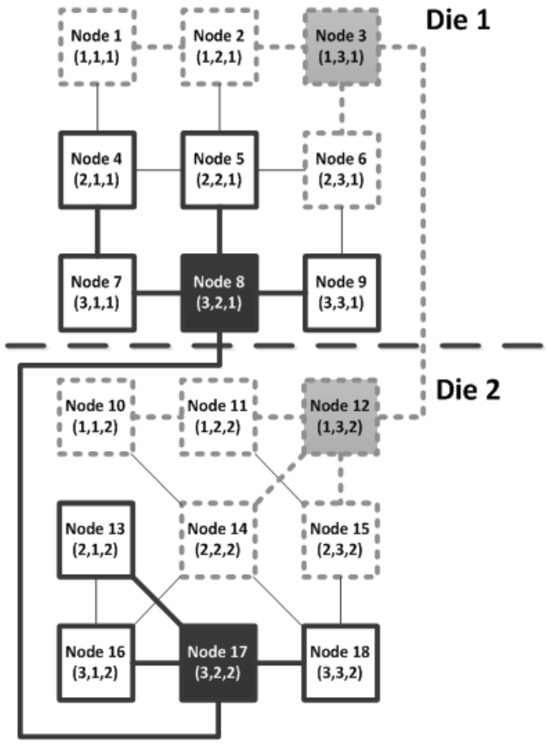

Fig. 13. Example of vertical regions

We offer following algorithm to build the vertical regions using a solution for current die:

Algorithm 3 Algorithm for building the vertical regions using a solution for current die

```
1:  function BuildVerticalRegions($S_{die}, die$)
2:      for all $otherDie \in \mathbb{D} \setminus \{die\}$ do
3:          $S_{otherDie} \leftarrow BestPartitionWithFixedMedians(S_{otherDie}, otherDie)$
4:      end for
5:      for all $m \in \mathbb{M}$ do
6:          $VRV_m \leftarrow \cup_{j=1}^{K} FNS_{m,j}$
7:          $VRE_m \leftarrow \{e(v,w) : v, w \in \mathbb{VRV}_m \wedge e(v,w) \in \mathbb{E}\}$
8:          $VR_m \leftarrow < \mathbb{VRV}_m, \mathbb{VRE}_m >$
9:      end for
10:     $\mathbb{DVRP} \leftarrow \{VR_i : 1 \leq i \leq N\}$
11:     $VRP \leftarrow < \mathbb{DVRP}, dp, lp >$
12:     return $VRP$
13: end function
```

The function “BestPartitionWithFixedMedians” in line 3 has the same logic with Algorithm 1, but here we use the fixed medians.

Basic criteria in selection of the best solutions for entire system:

- Minimum sum of diameters of vertical regions. D_{min} [hop].
- Minimum of load difference of TSVs, that forms the vertical regions. Δ_{min} [Number nodes].

A. *Minimum sum of diameters of vertical regions*

This criteria allows to find for entire system the solutions, in which there is minimum diameter of each vertical region, i.e. the most remote node of the one die is on the minimum distance from the most remote node of other die. These solutions increase speed of data transfer between remote nodes in the several dies, lower power consumption, i. e. in this case involved less communication lines for data transfer.

The following algorithm describes how you can filter your solutions by minimum sum of diameters:

Algorithm 4 Algorithm to filter solution set by the dp criterion

```
function WithMinDiamSum($VRP$)
    for all $VRP \in \mathbb{VRP}$ do
        for all $VR \in \mathbb{DVRP}$ do
            $dv_i \leftarrow \max_{v,w \in \mathbb{VRV}_i}(\min P(v,w))$  ▷ $P(v,w)$ - path between node v and node w
        end for
        $dp \leftarrow \sum_{i=1}^{|\mathbb{DVRP}|} dv_i$
    end for
    $Res \leftarrow \left\{VRP' : dp' = \min_{VRP \in \mathbb{VRP}} dp\right\}$
    return $Res$
end function
```

B. *Minimum of a load difference*

The load of TSV, which forms the vertical region is cardinality of vertical region. This criterion allows you to balance the load among all TSVs. It may happen that one TSV will transmit data for example 80% of all system nodes, and the second TSV will transmit 20% of the remaining nodes. Thus, the second TSV is idle and lose payload, while the first TSV is overloaded. To avoid such situations, it is necessary to balance the number of nodes in vertical regions. We offer following algorithm to your solutions by minimum of load difference:

Algorithm 5 Algorithm to filter solution set by lp criterion

```
function WithMinMinLoadDiff($\mathbb{VRP}$)
    for all $VRP \in \mathbb{VRP}$ do
        $lp \leftarrow \max_{i \neq j}(| n_i - n_j |)$        ▷ Where $n_i =| \mathbb{VRV}_i |$, $n_j =| \mathbb{VRV}_j |$
    end for
    $Res \leftarrow \left\{VRP' : lp' = \min_{VRP \in \mathbb{VRP}} lp\right\}$
    return $Res$
end function
```

Fig.14 and Fig.15 show examples of allowed solutions for topologies “Mesh” and “Torus” and for topologies “Torus” and “Butterfly” respectively. Each die has 12 nodes, the H is 2, the number of TSV is 2.

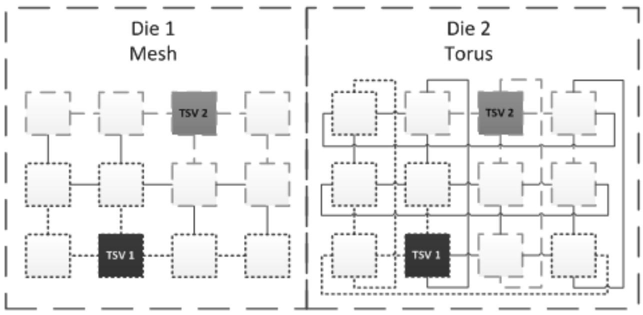

Fig. 14. Example of solution for 2 dies with 2 TSV. “Mesh” and “Torus” topologies on the dies

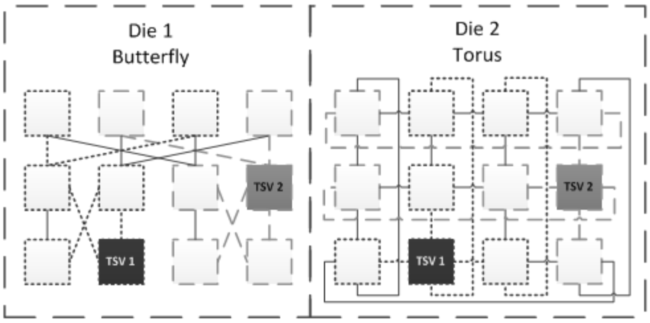

Fig. 15. Example of solution for 2 dies with 2 TSV. “Butterfly” and “Torus” topologies on the dies

These both solutions have similar characteristics: D_{min}=10 и Δ_{min}=0. Diameter of each vertical region is equal 5, and in summary we have got D_{min}=10. Each vertical region consists of 10 nodes, without considering nodes with TSV. As a result minimum of TSVs load difference is equal 0.

3-D network-on-chip are not constrained by two dies. In present, amount of dies may achieve 16 [11, 12]. In our case of solving this problem you need to find mappings for all dies pairs. Formula (5) calculates the necessary count of mappings for searching best solutions.

$$M = 2 \cdot C_n^2 = 2 \cdot \frac{n!}{(n-2)2!} = n(n-1) \qquad (5)$$

where n – count of dies.

Therefore, if we want to place TSV for 8 dies, we need do 56 mappings.

Fig.16 shows allowed solution for 3 dies with “Mesh”, “Torus” and “Butterfly” topologies.

Solution on Fig.16 has following characteristics: D_{min}=12 и Δ_{min}=0. Diameter of each vertical region is equal 6, and in summary we have got D_{min}=12. Each vertical region consists of 15 nodes, without considering nodes with TSVs. As a result minimum of TSVs load difference is equal 0.

V. Conclusion

In this paper was considered the problem of placement TSVs inside 3D system. We proposed two methods for placement TSV based on the search algorithm P-median in the graph. First method can be used for searching placement TSVs inside 3D system which consists of dies with the same topology. Second method can be used for searching placement TSVs inside 3D system which consists of dies with different topologies. Using these methods it is possible to find a placement TSVs where the minimum wire length between IP blocks in 3D system are achieved. These methods helps to get TSVs placement where the minimum sum of the diameters and the minimum of load difference of the nodes with TSV for system are achieved. By the way, most remote node of the one die is on the minimum distance from the most remote node of other die. These solutions increase speed of data transfer between remote nodes in the several dies, use less communication lines for data transfer.

Acknowledgment

The research leading to these results has received financial support from the Ministry of Education and Science of the Russian Federation according to the base part of the state funding assignment in 2016, project # 1810.

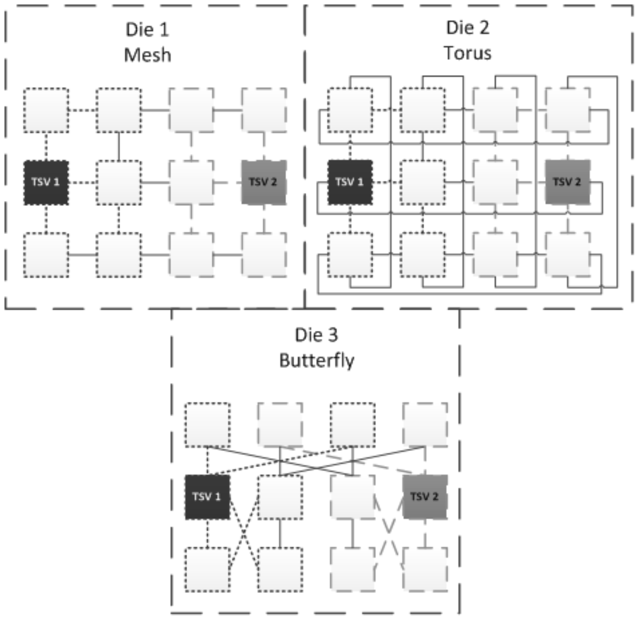

Fig. 16. Example of solution for 3 dies with 2 TSV. “Mesh”, “Torus” and “Butterfly” topologies on the die

References

[1] Sourav Das, Janardhan Rao Doppa, Dae Hyun Kim, Partha Pratim Pande and Krishnendu Chakrabarty, “Optimizing 3D NoC design for energy efficiency: A machine learning approach”, *in Proceedings of theInternational Conference on Computer-Aided Design (ICCAD)*, 2015, pp. 705 – 712.

[2] Samsung Announces Mass Production of Industry’s First 14nm FinFET Mobile Application Processor, Web: http://news.samsung.com/global/samsung-announces-mass-production-of-industrys-first-14nm-finfet-mobile-application-processor.

[3] The international technology roadmap for semiconductors (ITRS).

[4] A. Arakelyan, “3D IC design problems and challenges”, *Modern directions of theoretical and applied researches ‘2014,* 2014, pp.1-10.

[5] Dae Hyun Kim, Krit Athikulwongse and Sung Kyu Lim, “A Study of Through-Silicon-Via Impact on the 3D Stacked IC Layout”, *in Proceedings of the International Conference on Computer-Aided Design - Digest of Technical Papers*, 2009, pp. 674 – 680.

[6] K. Tatas, K.Siozios, D. Soudris and A. Jantsch, *Designing 2D and 3D Network-on-Chip Architectures*, Springer, 2014.

[7] L. Carloni, P. Pande, Y. Xie, "Networks-on-chip in emerging interconnect paradigms: advantages and challenges", *in Proceedings of the International Symposium on Networks-on-Chip (NoCS)*, 2009, pp. 93–102.

[8] Thomas Canhao Xu, "Optimal placement of vertical connections in 3D Network-on-Chip", *Journal of Systems Architecture*, May 2013, pp. 441–454.

[9] N. Christofides, *Graph theory: An algorithmic approach*, Academic press inc., 1978.

[10] *Suvorova, N. Matveeva, L. Kurbanov, "Applying the P-Medians in the Design of Modern Systems-on-Chip", in Proceedings of the Proceedings of the AINL-ISMW FRUCT 2015, 9-14 November 2015, pp. 102-109.*

[11] José L. Ayalaa, Arvind Sridharb, David Cuestaa, "Thermal modeling and analysis of 3D multi-processor chips", *Integration, the VLSI Journal,* Volume 43, Issue 4, September 2010, pp. 327–341.

[12] Thomas Canhao Xua, Alexander Wei Yina, Pasi Liljeberga, Hannu Tenhunena, "A study of 3D Network-on-Chip design for data parallel H.264 coding", *Microprocessors and Microsystems*, Volume 35, Issue 7, October 2011, Pages 603–612.