JACET Journal of Advances in Computer Engineering and Technology, 2(3) 2016

Application Mapping onto Network-on-Chip using Bypass Channel

Elnaz Alikhah-Asl[1], Midia Reshadi[2]

Received (2016-04-02)
Accepted (2016-08-15)

Abstract — **Due to increasing number of cores, the placement of the cores in NoC platform has become an important issue. If we can map the application cores close to each other to place them with more communication requirements, the performance parameters will improve and the network will be more efficient. In this paper, we propose two low complexity heuristic algorithms for the application mapping onto NoC to improve latency. In addition, one approach has been proposed to extract an Abstract graph from an application core graph, so, using this resent approach, we can map applications in two proposed algorithms. Moreover, we use bypass routers that can route packets in a cycle from the source to destination. Proposed algorithms and previous papers were compared on two real applications VOPD and MPEG-4 and results were reported.**

Index Terms — **Application Mapping, Bypass Router, Latency, Network-on-Chip, Throughput**

1-Computer Engineering Department, Science and Research Branch, Islamic Azad University,Tehran,Iran (e.alikhah@srbiau.ac.ir)
2-Computer Engineering Department, Science and Research Branch, Islamic Azad University,Tehran,Iran

I. INTRODUCTION

Increasing density of transistors, frequency and such issues have encouraged researchers to study the future of semiconductor industry. Therefore, the IC design is moving to the direction that heterogeneous elements can all be integrated on a single chip all of which are referred to as System on a Chip or System-on-Chip (SoC or SOC) [4]. In System-on-Chip, point-to-point interconnections and buses were among important communication methods. Since the buses could provide interconnections with high performance had many fans. But the buses only had the ability to connect up to 3-10 common units and are not scalable. Moreover, they also reduce performance, increase power consumption and noise phenomenon. Therefore, with the development of integrated circuits they lost their performance [3]. Accordingly, structuring principles of the communication architecture could have a great impact on overall system performance and energy

consumption of the system compared to network computing resources. Around 1999, researches were conducted to study the problems and designing communication part of SoC. As a result of the investigations, it was found that the problems associated with SOCs include all physical and architectural levels as well as operating system and the application [3]. Accordingly, a new approach called Network-on-Chip was created that is an appropriate way for buses and old communications. The Network-on-Chip that was taking shape had many features such as scalability and reusability. Network-on-Chip (NoC) architecture has the task to providing a communication infrastructure for resources and

the hardware resources have been developed in this direction and the blocks were formed that their relationship with each other formed a Network-on-Chip. In fact, the concept of Network-on-Chip is adopted from Off-Chip Interconnects in which for each chip a router has been implemented [2].

With the growth of network on chip technology, different aspects of it has developed. Among the others aspects, due to the increase in number of the cores Mapping is one of the important issues that has turned to a critical issue. Mapping issue is NP-hard, i.e. in order to map N functions on a mesh network with n tile, N! Status can exist. Therefore, instead of using N! Status for mapping, other methods such as heuristic, genetic and so on are presented. Mapping is defined as the arrangement of the each function on cores of the Network on Chip. It should be noted that the entire process of the system, depends on arrangement of the function core. They interchange with each other and complete the process of network on chip. Mapping substantially impacts throughput, latency, energy consumption and bandwidth. Therefore, mapping is one of the most important steps in designing network on chip. There are two methods for assigning the functions to the cores: dynamic and static. In dynamic mapping, tasks mapping will be implemented at run time, but in static mapping, task mapping will be implemented before run time. In networks on chip mostly static mapping will be used, because dynamic mapping increases the latency and energy consumption [2].

The rest of this paper is organized as follows. Section 2 presents the previous works. Section 3 gives a detailed description of the Mathematical Formulation for application mapping. Section 4 extract an Abstract Graph from application core graph. In Section 5, we present the First method of application mapping. In Section 6, we present second method for application mapping. In Section 7, proposed algorithm and previous papers were compared and Section 8 concludes this paper.

II . RELATED WORK

In paper [7], a mapping algorithm is provided on a two-dimensional (2D) network on chip. The main goal of this paper is to create a chain of connected cores, in order to use it to provide a new method for mapping. This article also has tried to achieve less bandwidth in compare to comparable articles.

In paper [9], a heuristic algorithm is provided on a mesh two-dimensional network on chip. The article aims to create a list of priorities based on total communication bandwidth and average communication bandwidth. The method presented in this paper was compared with 3 other method where better results were achieved.

In paper [10], an accelerated heuristic algorithm is provided on a two-dimensional network on chip. This algorithm is presented in order to decrease bandwidth, as well as to minimize the latency.

In paper [11], mapping will be implemented through creating a list of priorities. The method presented in this article aims to decrease communication cost.

In paper [12] the aim is to present an algorithm with low complexity and to minimize the number of steps between IP cores, in order to improve energy consumption and other parameters of efficiency. The purpose of this heuristic method is to reduce communication cost and to obtain better results in compare to other articles.

In paper [13], a reliability-aware algorithm is provided on a two-dimensional network on chip. In this paper that is used to minimize communication traffic, function graph is divided into two Sub-Graphs. So, communication traffic between two Sub-Graphs will be minimized and the traffic in each graph will be maximized. In this article, percentage of correctly delivered packets and average packet delivery time are investigated. In terms of routing, in common network on chip, sending packets from source to destination will be done step by step, and this leads to an increased latency. In recent years some articles have been proposed [5, 6], in which bypass channel is added to the router structure. Due to the existence of these bypass channels there is no need that packs stop in each router during their movement and consequently the amount of latency will improve. The necessity of using these kinds of routers has increasingly been felt due to increase in number of cores on chip and scalable topology of network on chip, such as mesh and due to the increase in number of the steps from source to destination.

In article [5], a network on chip known as SMART is provided, where a single cycle data path is created for the whole direct route from source to destination. This method uses a series of SSR control links for sending the packs. SSRs are

part of the control route that carry the steps that the flit should pass. It also should also be noted that due to the existence of bypass channels there is no need to buffer in each router again. This method uses a series of SSR controlling links for sending the packs. SSRs are part of control route that carry the steps that flits should pass. Also, it should be noted that due to the existence of bypass channels, there is no need to buffer once again in each router.

In paper [5], SSR overhead wires are provided. In paper [6] SSR wires are removed and header flits are used instead of control links. Header flit leads to creating pass routes in intermediate routers and after creating this pass route, body flit and tail flit pass from intermediate routers without being stopped. Also, in this article the amount of latency has improved in compare to paper [5].

In order to improve latency parameter, two methods are presented for mapping, also an abstract graph is proposed, that both methods presented for mapping will use it. As we know, mapping is one of the most difficult steps in the process of network on chip. Therefore, applications that should be implemented to improve the process have been described as a graph, such as vopd application graph and etc. In this article, we will turn these application graphs to a simpler graph. This graph will be created according to the initial graph. Mapping first and second methods will be implemented according to the abstract graph which is extracted from application graph. In abstract graph, all communications are shown. Both methods presented in this paper will be conducted in a single phase and there will be no swap phase. Therefore, mapping process will be simpler. This paper, also tries to reduce the number of steps from source to destination. In both methods presented for this paper, mesh topology is used due to its appropriate structure and simple arrangement, though the presented methods are applicable to all topologies.

III . PROBLEM DEFINITION

communication between the cores of the SoC is represented by the core graph:

Definition 1. The core graph for an application is a directed graph, G(C, E) with each vertex $c_i \in C$ representing a core and the directed edge $e_{i,j} \in E$ representing the communication between the cores ci and cj. The weight of edge ei, j, denoted by commi, j, represents the bandwidth requirement of the communication from ci to cj. On the other hand, the given NoC topology can be represented in the form of a topology graph [2].

Definition 2. The NoC topology graph is a directed graph P (U, F) with each vertex $u_i \in U$ representing a node in the topology and the directed edge $f_{i,j} \in F$ representing a direct communication between the vertices ui and uj. The weight of the edge fi, j, denoted as bwi,j, represents the bandwidth available across the edge fi, j. A mapping of the core graph G(C, E) onto the topology graph P(U, F) is defined by the function map: C→U, such that, $\forall c_i \in C, \exists u_j \in U$ and map(ci) = uj. The function associates core ci to router uj. Naturally, mapping is defined only when |C| ≤ |U|. The quality of such a mapping is defined in terms of the total communication cost of the application under this mapping [2]. The communication between each pair of cores can be treated as flow of a single commodity dk, k = 1, 2,..., |E|. The value of commodity dk, corresponding to the communication between cores ci and cj is equal to commi, j, and the bandwidth requirement. If ci is mapped to the router map (ci) and cj is mapped to map (cj), the set of all commodities D = {dk} is defined as follows

$$D = \{d^k \mid \text{value}(d^k) = \text{comm}_{i,j}, \text{for } k = 1,2,..|E| \text{and} e_{i,j} \in E\} \quad (1)$$

Also, Source (dk) = map (ci) and sink (dk) =map (cj), The link between two individual routers ui and uj of the topology has a maximum bandwidth of bwi,j. The total commodity flowing through such a link should not exceed this bandwidth. The quantity x_(i,j)^k indicating the value of commodity dk flowing through the link

$$x_{i,j}^k = \begin{cases} value(d^k), \ if \ link(u_i, u_j) \in Path(source(d^k), \ sink(d^k) \\ 0, \ otherwise \end{cases} \quad (2)$$

(ui, uj) is given by,

Where Path (a, b) indicates the deterministic routing path between the mesh nodes a and b in the topology. Satisfaction of bandwidth limitations of individual links must be ensured. That is, all mapping solutions should satisfy the following relation.

$$\sum_{k=1}^{|E|} x_{i,j}^k \leq bw_{i,j}, \text{ for all i,j} \in \{1,2 \ldots, |U|\} \quad (3)$$

If all bandwidth constraints are satisfied, the communication cost of a mapping solution is given by,

$$T = \sum_{k=1}^{|E|} value(d^k) hopcount(source(d^k), sink(d^k)) \quad (4)$$

Here, hopcount(a, b) is the number of hops between the topology nodes a and b. For a deterministic shortest path routing, hopcount corresponds to the minimum number of hops between the constituent nodes. Since communication cost is very much dependent on the mapping solution, the overall mapping problem is to optimize the communication cost, ensuring that the bandwidth constraints of all individual links are satisfied. The communication cost affects the performance of the overall system and its energy consumption, as both of these factors are directly proportional to the Total hopcount [14]

IV. ABSTACT GRAPH

In this paper we want to propose low complexity methods to map applications in NoC and achieve better results than other algorithms.

Abstract graph is a graph that is extracted from the application core graph, with it we can map cores simpler than other mapping methods. The Abstract graph is applicable to any kind of application graphs.

To extract the Abstract graph from core graph, the application with the most node degree, considered as the Root. As follow, we explain the stages of root selection:

1.Root: the application from core graph that has the most node degree.

2.If in an application core graph, there are more than one application that have equal maximum node degrees, then, the application will be chosen that has the most communication requirement.

3.If there are more than one application in the above second condition, then a root will be chosen randomly.

4.The above steps are continued constantly for all applications up to leaves.

For example, in VOPD graph as seen in figure 1 each of the applications 12 and 9, has node degree of 4 and the other ones have node degree of 1 or 2 or 3. We must choose an application between 12 and 9, according to condition 2, the application of 9 will be chosen as the root. After selecting the root, we must complete the abstract graph due to relations between applications until to leaves. The neighbors will be chosen based on their communication requirements and set as root's child from left to right. For instance, Neighbors of the root in VOPD graph are 8, 10 and 12 that have communication requirements 1113, 907, 64 respective. They will be arranged according to their priority from left to right as shown in figure 2.

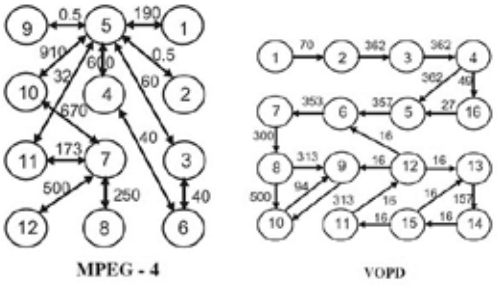

Figure. 1. Application core Graph

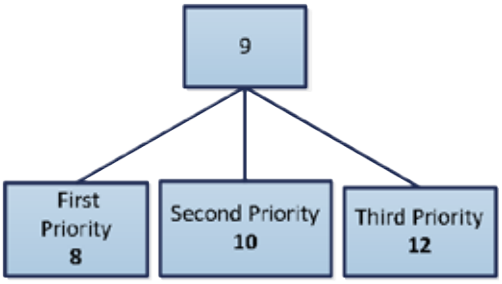

Figure.2. Priority of setting child in Abstract graph

1.Input: Core Graph G=(C,E), Topology Graph P(U,F) 2.Output: Abstract Graph;
3. Find the core with maximum node degree in G(C,E);
4.The core with maximum node degree is Root;
5.If the number of cores with maximum node degree>1
6.Find the core with maximum communication requirements;
7.the core with maximum{communication requirement and node degree}=root;8. Else
9.if The number of cores with maximum communication requirement and node degree>1;
10.Select a random core from maximum{node degree and communication requirement}as root;
11.Traverse all neighbors of root on G(C,E) and set them from the highest communication requirement to lowest requirement from left to right;
12.Traverse all of neighbors of child nodes until all of G(C,E) have been set in abstract graph according to 4;
13.If communication requirement of child nodes in a level are the same
14.select randomly a node to set it in left side of its parent;

Figure.3. The Pseudo code of the Abstract Graph

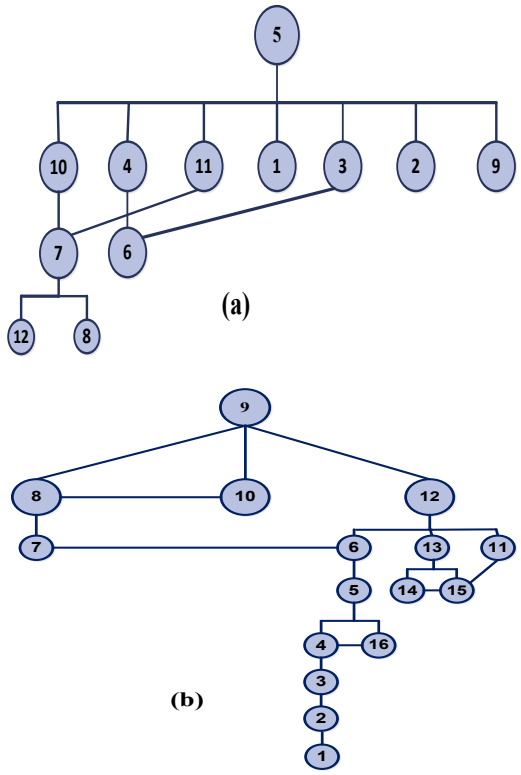

Figure.4. (a) The abstract graph of MPEG-4 (b) The abstract graph of VOPD

The following example explains how to extract the abstract graph from the application core graph of MPEG-4:

In MPEG-4 core graph, as seen in fig 4(a), the application of 5 has the most node degree, so this application is the root in the abstract graph. Neighbors of root are set as the child of this abstract graph in order to their communication requirements from left to right.

V. FIRST MAPPING METHOD

In this case, first of all, the application that is chosen as the root in the abstract graph, Should be mapped. There is no limitation for choosing the location, we can map the root in the first free place on the mesh topology. Remaining steps for mapping child are as follow:

1.The child of root, in order to their priority, from left to right in abstract graph, are mapped in the x-axis and in direction of the root, closer free tiles have priority. As seen in fig.6(a) after mapping root 5, according to priority, the first application 10 is mapped in the closest place to root, then applications of 4 and 11 are mapped. All of them in x-axis and in the direction of root.

2. If in root's child, there are unmapped applications, they will be mapped in y-axis and in the direction of root. In the abstract graph, there are 4 child of root that are unmapped, so according to priority, the applications of 1, 3 and 2,respectively, will be mapped . If there is still unmapped application, it should be mapped in the nearest place to root like application 9 in MPEG-4.

1.Input: Abstract Graph (AG), Topology graph P=(U,F);
2.The number of tile in a row=N;
3.The number of tile in a column=M;
4.The number of neighbors of root=NOR;
5.Output: Mapping application of AG on the P=(U,F);
6. Map root node of AG on a free tile of P(U,F);
7. Map root's child nodes:
8.a←(N-1)-NOR
.9If a≥0
Map all of root's neighbors on X row that root is mapped from left to right;
Map the neighbors with more communication requirement closer to root;
10.Else if a<0 map NOR as many as (n-1);
11.a=$|a|$
12.b←(M-1)-a
13.If b≥0
Map child nodes that aren't map yet in Y column of root;
Map neighbors with more communication requirement closer to root;
14.Else if b<0 map NOR as many as (M-1)
Map remain NOR in closest tile to root;
15.If two parent have a common child
16.If two parent or more are in same row or column
Map child in their row or column and closer to parent that needs more communication requirement ;
17.Else
Map child randomly close to a parent;
19.If parent with common child are in different row or column
Map child in a tile that X row of a parent cross Y column of other parent;
20.If there isn't any free tile in the place of X row that cross Y column
Map the child closer to parent that needs more communication requirement;
20.Else map child randomly in a free tile close to parents;

Figure.5. Pseucode of the first algorithm mapping

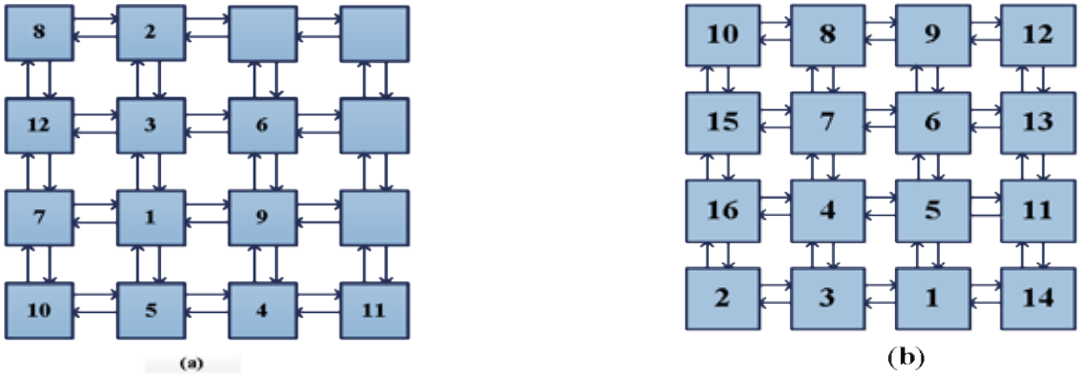

Figure.6. (a) First Mapping of MPEG-4 core graph on NoC architecture core graph (b) First Mapping of VOPD core graph on NoC architecture core graph

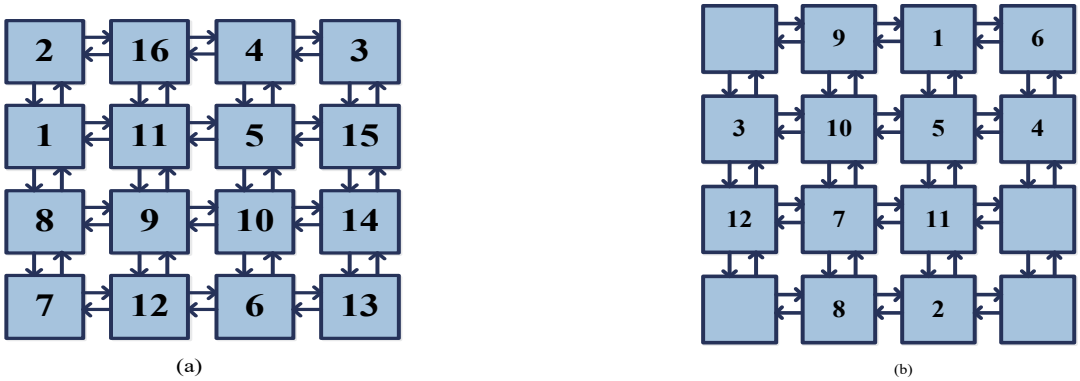

Figure. 7. (a) Second Mapping of VOPD core graph on NoC architecture core graph (b) Second Mapping of MPEG-4 core graph on NoC architecture core graph

```
1.Input: Abstract Graph(AG), Topology Graph P=(U,F);
2.Output: Mapping application of AG on the P=(U,F);
3.Mapping phase:
4.Map root node of AG in free tile with maximum neighbors in P(U,F);
5.to map child nodes follow below rules;
6. First priority to map is closest tile on west tile of parent;
7.Second priority to map is closest tile on east tile of parent;
8.Third priority to map is closest tile on south tile of parent;
9.Fourth priority to map is closest tile on north tile of parent;
10.Map child node of root according to 1;
11.If there isn't free tile close tile to parent
12.Close free tiles on row have more priority than column tiles;
13.Else map child in a close tile to parent;
14.If two parents have a common child
15. If two parents or more are in same row or column
16.Map child in their row or column and closer to the
parent that needs more communicant requirement;
17.Else
18.Map child randomly close to a parent;
19.If parent with common child are in different row or column
20.Map child in a tile that X row of a parent cross Y column of other parent;
21.If there isn't any free tile in the place of X row that crosses Y column
22.Map child closer to the parent that needs more communication requirement;
23.Else map child randomly in a free tile close to parents;
```

Figure. 8. Psuedocode of the second algorithm mapping

VI. SECOND MAPPING METHOD

Root is mapped in a free place on NoC. The remaining steps of this algorithm for mapping child are as the follows:

1. First priority is the free closest place in the west side of the root.

2. Second priority is the free closest place in the east side of the root.

3. Third priority is the free closest place in the south side of the root.

4. Fourth priority is the free closest place in the north side of the root.

Fig 7, is VOPD and MPEG-4 that mapped with the second method on NoC .

Fig 8 is Psuedocode of the second algorithm mapping.

VII. EVALUATION

In this section, real performance of the prIn this investigation, we compared our VOPD graph with the results of 3 previous papers [7, 12, 11], in other hand, we compared our MPEG-4 mapping graph with the results of 3 previous papers of [15,12,13].

The results of two mapping methods proposed in this paper, were simulated with asynchronous bypass channel [6] and without it. For simulation, we used Noxim simulator.

Fig. 9 shows the results of mapping for latency. As seen in the figure, the results of latency with and without asynchronous bypass channel are shown. The networks without asynchronous bypass channel, are saturated in injection rate of 0.05 that is better than other mapping algorithms. also using asynchronous bypass channel, saturation for first and second methods are 0.09 and 0.08 respectively. Bypass channel can route Packets without latching in intermediate routers. So latency will decrease significantly. Figure 10 is Results of Latency for MPEG-4 in mesh 4*4.

As seen in Figure 9 and 10, the results of latency without asynchronous bypass channel are developed in compared with previous methods, because we proposed efficient low complexity heuristic algorithms (an Abstract Graph and two mapping algorithms) that can map cores close to each other in compare with previous algorithms.

Against it's advantages, bypass channels have different logic routers that make NoC platforms bypass without latching in intermediate reuters. So it will increase area overhead in compare with baseline routes in NoC.

VIII. CONCLUSION

In this paper, we have presented an Abstract graph and 2 fast mapping algorithms to improve Latency. This paper is proposed to extract an efficient graph from application core graph. Two presented mapping algorithms have used the abstract graph for mapping instead of common application core graphs that is used in most research. Also, the both mapping algorithms did not use swap phase and applications are mapped without iterative improvement. Our approach can be extended to map cores onto various NoC topologies. However in this paper, we have focused on 2D mesh network with XY routing. In Abstract Graph, the application that has the most node degree is chosen as root. Then for completing the Abstract graph from root's child to leaves, the communication requirements of the applications are set from highest value to lowest value in each level of the Abstract Graph. In the first mapping algorithm, we mapped applications using Abstract Graph. The root can be mapped in the first free place of mesh topology and there is no limitation for mapping the first application. The other applications in the abstract graph, are mapped first, in X dimension of root and then, in Y dimension of root. Using this method, we can map the applications with the most communication requirements connected with each other in closest place. In second mapping algorithm, we proposed 4 movement priorities to map applications on to mesh topology. Using the Abstract graph and 2 mapping algorithms proposed in this paper, we could achieve better results of Latency in comparison with the ones of 4 previous papers.

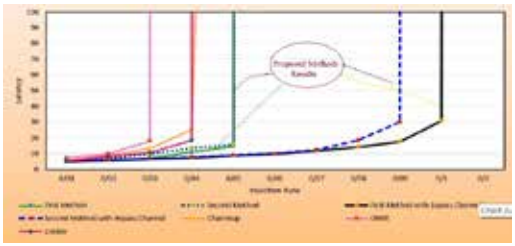

Figure 9. Results of Latency for VOPD in mesh 4*4

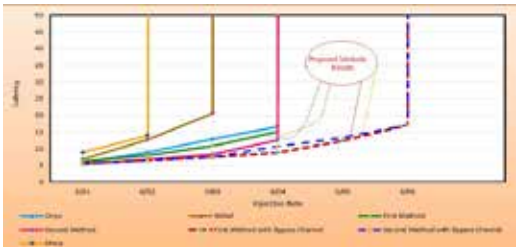

Figure 10. Results of Latency for MPEG-4 in mesh 4*4

REFERENCE

[1] Santanu kundu, Santanu chattoadhyay, "Network-on-Chip:The Next Generation of System- on- Chip Integration", London, CRC Press, 2015.

[2] pradip Kumar Sahu, Santanu Chattopadhyay, "A survey on application mapping strategies for Network-on-Chip design, Elsevier, Journal of Systems Architecture, pp.60-76, 2013

[3] Axel Jantsch, Hannu Tenhunen, "Networks on Chip", United State of America, Kluwer Academic Publishers, 2003.

[4] Fernando Moraes, Ney Calazans, Aline Mello, Leandro Möller , Luciano Ost, "HERMES: an infrastructure for low area overhead packet switching network on chip", Integration, the VLSI Journal, 2004, pp.69–93.

[5] Krishna T, Chia-Hsin Owen Chen, Woo Cheol Kwon, Li-Shiuan Peh, "Breaking the on-chip latency barrier using SMART", IEEE 19th International Symposium on High performance Computer architecture, 2013.

[6] Amir Fadakar noghondar, Midia Reshadi, "A low-cost and latency bypass channel-based on chip network", Springer, Journal of Super Computing, 2015.

[7] M.tavanpour, Ahmad Khademzadeh, Majid Janidarmian, "Chain-Mapping for mesh based Network-on-Chip architecture", IEICE Electronics Express,pp.1535-1541,2009.

[8] Y.chen, Lunguo Xie, Jinwen Li, "An energy-aware heuristic constructive mapping algorithm for Network-on-Chip", IEEE 8th International Conference on ASIC, pp.101-104, 2009.

[9] S.Tosun, "New heuristic algorithm for energy aware application mapping and routing on mesh-based NoCs", Elsevier, Journal of system Architecture, pp.69-78, 2011.

[10] S.murali, G.demicheli, "Bandwidth Constrained mapping of cores onto NoC architectures, Proc. of design, Automation and test in Europe conference and exhibition, pp. 896-901, 2004.

[11] S.saeidi, Ahmad Khademzadeh, Fatemeh Vardi "Crinkle: a heuristic mapping algorithm for network on chip", IEICE Electronics Express, pp.1737-1744, 2009.

[12] M.janidarmian, Ahmad Khademzadeh, Misagh Tavanpour, "Onyx: a new heuristic bandwidth-constrained mapping of cores onto network on chip", IEICE Electronics Express, pp.1-7, 2009.

[13] A.Patooghy, Hamed Tabkhi, Seyed Ghassem Miremadi, "RMAP: a Reliability-Aware Application Mapping for Network-on- Chips", 2010 third International conference on dependability, pp. 112-117, 2010.

[14] Partha Pratim Pande, Grecu.C, Jones.M, Ivanov.A, Saleh.R., "Performance Evaluation and Design Trade-Offs for Networkon- Chip Interconnect Architectures", IEEE Transactions on Computers, pp.1225-1040, 2005.

[15] Xiaohang Wang, Mei Yang, Yingtao Jiang, Peng Liu, "Power-Aware Mapping for Network-on-Chip architectures under bandwidth and latency constraint", International conference on embedded and multimedia computing, pp.1-6, 2009.