# A New Binomial Mapping and Optimization Algorithm for Reduced-Complexity Mesh-based On-Chip Network

Wein-Tsung Shen, Chih-Hao Chao, Yu-Kuang Lien, and An-Yeu (Andy) Wu
Graduate Institute of Electronics Engineering, and Department of Electrical Engineering, National Taiwan University, Taipei 106, Taiwan, R.O.C.
{wtshen,chihhao}@access.ee.ntu.edu.tw, andywu@cc.ee.ntu.edu.tw

***Abstract* – This paper presents an efficient binomial IP mapping and optimization algorithm (BMAP) to reduce the hardware cost of on-chip network (OCN) infrastructure. The complexity of BMAP is $O(N^2 log(N))$. Based on our OCN system synthesis flow, the proposed algorithm provides more economic network component mapping in comparison with traditional OCN mapping algorithm. The experimental result shows total traffic on network is reduced by 37% and average network hop count is reduced by 46%. With further optimization, the hardware efficiency is enhanced therefore the total hardware cost of network infrastructure is reduced to 51%~85%.**

## I. Introduction

With the increasing complexity of System-on-chip (SoC) design, data exchange within chip is becoming more difficult. Traditional interconnection approaches cannot provide sufficient support for future giga-scale SoC design. On-chip network (OCN) [3] is an approach to solve the incoming physical routing, flexibility, scalability, and reliability problems. OCN provides a possible and economical method to integrate complex systems on a single chip with the advanced VLSI technology [1][2]. Many researches indicate that the synthesis of OCN dominates the infrastructure hardware cost and network performance [5]. The target of OCN synthesis is to find the suitable network infrastructure with minimum cost. Among many network topologies, we choose two dimensional mesh based topology as our OCN topology. The mesh-based architecture has good scalability and regularity, and hence it's comprehensively adopted as OCN basic topology in many related works Radu's tile[4], Xpipes[6]. SUNMAP[8] propose an OCN system design flow, NMAP, to map IPs on given topology and generate the synthesized OCN target in SystemC.

The NMAP [6]-[8] model the OCN synthesis problem as a shortest-path optimization problem and runs a $O(N^4 log(N))$ complexity algorithm. We discover that the synthesis flow can be partitioned into a two-stage task. With the proposed greedy binomial mapping and optimization algorithm (BMAP), the complexity of synthesis is reduced to $O(N^2 log(N))$. In our OCN design flow, the synthesis is consisted of a mapping stage and an optimizing stage. For comparison, we take two real SoC applications: a video object plane decoder (VOPD) and an MPEG-4 decoder as our applications. The traffic models of the given applications are extracted and input to our implementation program. The experimental results show that the proposed algorithm reduces 37% total traffic load on the network and 46% network hop count.

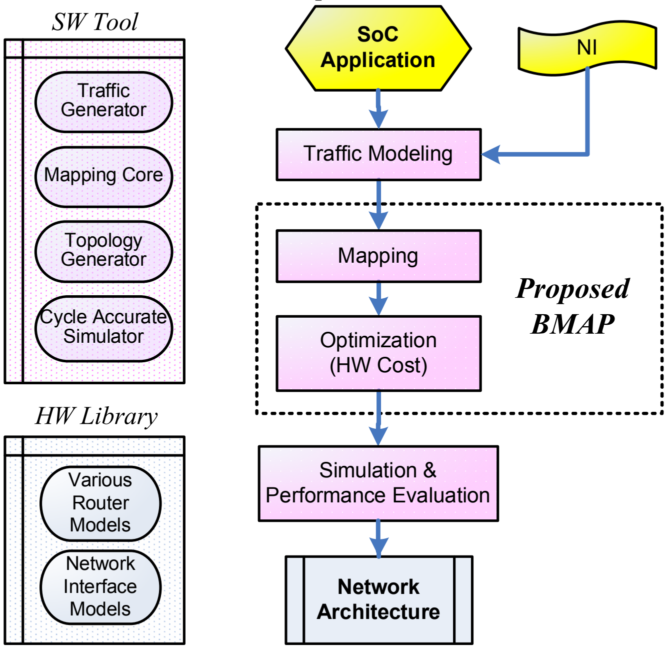


Figure 1. OCN design flow and synthesis (marked by the dotted-line square frame)

## II. OCN Synthesis Review

Figure 1 shows the modified OCN design flow. The SoC application is the target SoC system running specific application. We assume the network interface (NI) used for the given SoC application is chosen by system designer before OCN synthesis. NIs packetize the data transmitted between IPs in SoC. In this paper, our NIs adopts open core protocol (OCP) and use synchronous request-ack full handshake. We assume that the target OCN adopts wormhole-based architecture. Therefore, the NIs should be able to recognize three types of flits which compose a packet: header, body, and tail. For a given SoC application, the traffic model is extracted through modeling the original system components and dataflow by vertices and edges of a directed graph. The traffic matrix of given SoC application is generated from the modeling graph. The proposed BMAP is marked by the dotted-line square frame in Figure 1. BMAP partitions the OCN synthesis into mapping and optimization. The final synthesized network architecture is improved and evaluated through three steps:

- **Mapping**: The mapped network architecture is the first stage output of the OCN design flow. The basic one comes from our binomial mapping algorithm.
- **Optimization**: The optimized network architecture is the second stage output of the OCN design flow. This one is evolved from the mapped with proposed optimization algorithm.
- **Simulation**: The simulated network architecture is the third stage output of the OCN design flow. This one is simulated with our cycle-accurate SystemC simulator for performance evaluation.

Each of the output architecture in the three steps can be translated to a real composition of existing hardware components to form the synthesizable OCN for further verification. In comparison with Xpipes[6] and SUNMAP[8], the proposed BMAP tries to minimize the total traffic on network, hop number, and the OCN hardware cost. The proposed binomial mapping algorithm and optimization algorithm raise the hardware efficiency of OCN. Therefore, the total network traffic loading and transmission latency is reduced. In the following sections we describe the proposed mapping algorithm and optimization approach in detail.

## III. Proposed Binomial Mapping and Optimization Algorithm (BMAP)

Mapping a SoC system to OCN is the first and the most important step in our design flow because it will dominate the overall performance and cost. The computation complexity of binomial mapping algorithm is $O(N^2 log(N))$. Compared with the Xpipes' work [9][10], whose computation complexity exceeds $O(N^4 log(N))$, our work saves $O(N^2)$ time to accomplish the mapping. Instead of modeling the mapping problem as the shortest-path optimization problem, we adopt a greedy style on the extracted traffic model. The proposed style generates a component mapping with better total traffic load, which is advantageous for the optimization of hardware cost. Furthermore, the binomial mapping algorithm can be easily combined with other existing optimization methods, such as shortest path optimization. Figure 2 shows the proposed binomial mapping and optimization algorithm is mainly composed by three major operations: binomial merging iteration, topology mapping and traffic surface creating, and hardware cost optimization.

### A. *Binomial Merging Iteration*

The binomial mapping iteration, shown in Figure 2, contains three steps: calculating IP ranking, merging IP-set, and refreshing IP-set. The iteration runs until only one IP-set is left. The proposed fast mapping method uses binomial merging iteration to decide the location of each IP on mesh topology. To satisfy the requirement surface from the extracted traffic model, this method greedily finds the best merging result through the binomial merge. Binomial merge is based on concept of tree structure representation for IPs. The unmapped IPs are viewed as unconnected tree roots. Each set of IPs is viewed as a merged sub-tree. The binomial merging iteration transforms the IP mapping problem as a simple tree merging problem. The tree merging takes IP ranking as cost function. According to the ranking of each set of IPs, we merge sub-tree and view the new sub-tree as an IP-set. After several iterations, we can get the final tree. Considering the square feature of mesh-based topologies, we choose binomial merge method for efficiency. It takes $log_2(N)$ iterations in $O(N^2)$ time, where N denotes the total IP number. Due to the low computation complexity, binomial mapping performs well for large scale SoC. Furthermore, binomial mapping have the flexibility of tree structure, and hence it is easy to apply other optimization methods, such as localization, shortest path [9], etc. [5][10][11][12]

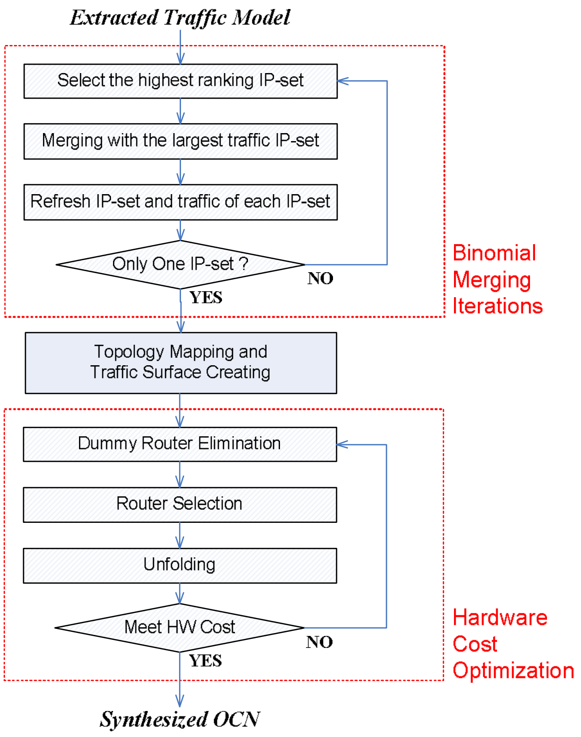

Extracted Traffic Model
Select the highest ranking IP-set
Merging with the largest traffic IP-set
Refresh IP-set and traffic of each IP-set
Only One IP-set ?
NO
YES
Binomial Merging Iterations
Topology Mapping and Traffic Surface Creating
Dummy Router Elimination
Router Selection
Unfolding
Meet HW Cost
NO
YES
Hardware Cost Optimization
Synthesized OCN

Figure 2. Proposed BMAP flow chart

1) *Calculate IP Ranking*: First we define ip-sets as basic element which descripts a set of merged ip. Based upon the extracted traffic load of original SoC system, we can calculate the ranking of each IP. After completing a merging iteration, the ranking of each IP-set is updated. The IP ranking is calculated by summing the traffic from each IP to the other IPs and from the other IPs to each IP. The IP ranking is calculation is expressed by Eq. (1). The *ranking(i)* denotes the ranking of IP-set *i*. The *requirement(i,j)* denotes the bandwidth requirement from IP-set *i* to IP-set *j*.

2) *Merging IP Set*: Merging is the main step of binomial mapping. Based upon IP ranking, the IP sets are merged two-by-two on every iteration. Therefore, it takes total $log_2(N)$

iterations to complete the binomial merging iteration. Figure 3 shows an example with N = 16 and how the IP-sets are merged. By merging two IP-sets, it finds the best contact between boundaries (i.e. minimal traffic load). In binomial mapping, there are only 4 or 16 cases. These cases are caused by rotations of IP-sets and shown in Figure 4.

3) *Refreshing IP Set*: In this step, the new requirements of merged IP-sets is recalculated they are IP-sets as an individual IP. The new ranking of two merged IP-sets is calculated by summation their origin requirements and subtract requirements between them, which can be expressed as follows:

$$ranking(i) = \sum_{j=1}^{N} \left( requirment(i,j) + requirment(j,i) \right) \quad i = 1 \sim N, \tag{1}$$

$$ranking(k) = ranking(i) + ranking(j) - requirment(i,j) - requirement(j,i). \tag{2}$$

### B. Topology Mapping and Traffic Surface Creating

After binomial mapping, our tool produces a traffic surface. The traffic surface shows the traffic load of each router and the centralized traffic after binomial mapping. We use minimal path routing, such as X-Y routing. Then the traffic load of each router on OCN can be accumulated as centralized traffic. Based on this surface, we can easily optimize hardware cost by selecting proper routers from the given library of hardware models. Figure 5 shows an example of the traffic surface. Based upon this surface, we can find that the result of binomial mapping is very suitable to centralize traffic.

### C. Hardware Cost Optimization

Hardware cost of OCN is an important issue. Routers of OCN system dominate the hardware cost, especially for their buffers. According to the traffic surface, we use several approaches: (1) eliminate dummy router, (2) router selection, and (3) unfolding, in BMAP to reduce the hardware cost of router buffers to mostly save the cost.

1) *Dummy Router Elimination*: Some dummy routers are added at the start point of binomial map for $4^n$ routers. After binomial merging iteration, the dummy routers are put to the boundary of mesh. Therefore removing the dummy routers can't affect network performance.

2) *Router Selection*: The cost of router buffers dominates the OCN hardware cost. We share single buffer among low bandwidth, simplified as BW, required input channels. Therefore a hardware library, which contains a variety of routers with different buffer banks, is build. Table 1 shows the relation of BW and n-bank router.

3) *Unfolding*: After binomial merging iteration, the router with heavy traffic would be gathered. Some router will have traffic load over router BW. Unfolding technique is used to double the link BW and the router BW by adding additional router for critical node that needs larger bandwidth without increasing latency, which is shown in Figure 6.

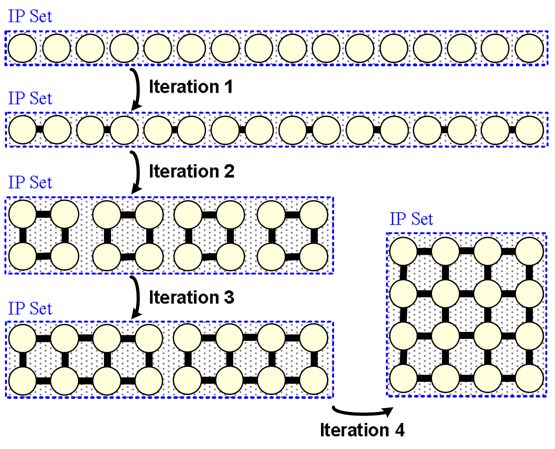
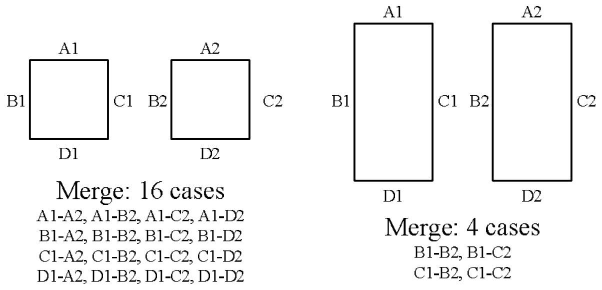


Figure 3. An example of binomial merging iteration (N = 16).


A1
A2
B1
C1
B2
C2
D1
D2
Merge: 16 cases
A1-A2, A1-B2, A1-C2, A1-D2
B1-A2, B1-B2, B1-C2, B1-D2
C1-A2, C1-B2, C1-C2, C1-D2
D1-A2, D1-B2, D1-C2, D1-D2
A1
A2
B1
C1
B2
C2
D1
D2
Merge: 4 cases
B1-B2, B1-C2
C1-B2, C1-C2
Figure 4. Merging cases of two IP-sets.

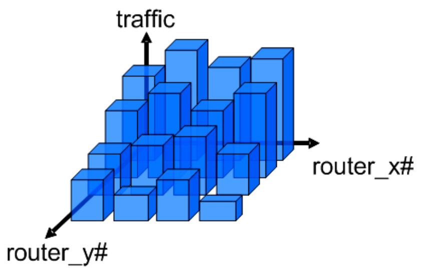
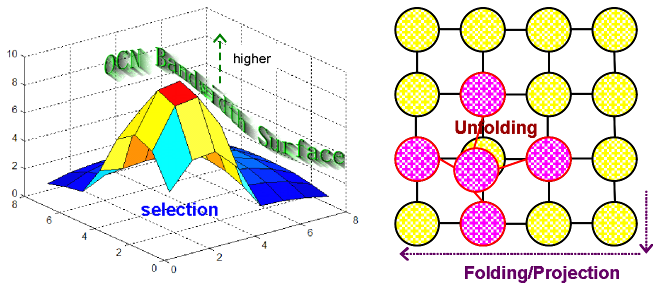


Figure 5. Traffic Surface of OCN system after Binomial Mapping.



(a)Traffic Surface. (b)Unfolding Approach

Figure 6. Optimization Approaches of BMAP.

Using these approaches, we can observe the traffic surface. For traffic under router BW, we adopt n-bank router in

Figure 7. For traffic over router BW, we unfold the critical nod with more routers. Finally, we can get a very low-cost optimized OCN architecture under bandwidth constraints of SoC applications.

## IV. Experiment Results

We compare BMAP with NMAP. In algorithm domain, NMAP maps the cores onto a mesh topology under bandwidth constraints, and minimizes the average communication delay by iterations. It costs more than *$O(N^4log(N))$* (i.e. *N* is IP number) computation time to do its shortest path optimization. On the other hand, BMAP uses binomial merge method to get a fast and semi-greedy optimized result, and prepares for the next optimization phase for hardware cost-down. It costs *$O(N^2log(N))$* complexity and takes only a few seconds for the VOPD and MPEG-4 cases. The comparison in algorithm domain is shown in Table 2. In order to compare performance in a reliable way, we choose 2 different applications: VOPD and MPEG4 (shown in Figure 8). These two applications are also used in [9].

Figure 9 is the simulation results of binomial mapping of BMAP comparing with NMAP, PMAP, GMAP and PBB in [9]. Because NMAP has the best performance in [9], NMAP is adopted as a reference point to judge our approach. In application 1, the bandwidth is improved to 0.98 but the hop number is increased to 1.02. It means that the binomial merging algorithm of BMAP efficiently get a better semi-greedy bandwidth-optimized result and can substitute the initial phase of NMAP if necessary. Moreover, the hop number can be reduced in optimization phase.

Base on the simulation results of application 1, we can say that the binomial algorithm of BMAP is the best solution in the initial state for regular 2-D mesh topology (i.e. IP number = $4^N$, where N is integer). It seems that the results of application 2 (12 IPs) are not similar to application 1 (16 IPs). It's because that we add 4 dummy IPs to keep the regularity of 2-D mesh topology and cause lower average router traffic (0.8) but increase the Hop number (1.71). The performance degradation of Hop will be solved in the optimization phase.

Figure 10 shows that the HW cost of application 1 and 2 reduced 50% after optimization. Therefore TABLE 3 shows that BMAP, compared with NMAP, reduces to 89% bandwidth and reduces to 68% HW cost. The traffic ratio and HW cost ratio can be defined as follows.

$$Traffic\ ratio = \frac{Traffic\ load\ of\ BMAP}{Traffic\ load\ of\ NMAP}, \qquad (3)$$

$$HW\ ratio = \frac{HW\ cost\ of\ BMAP}{HW\ cost\ of\ NMAP}. \qquad (4)$$

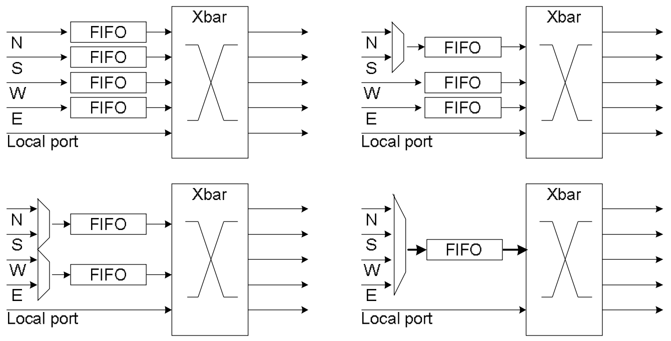


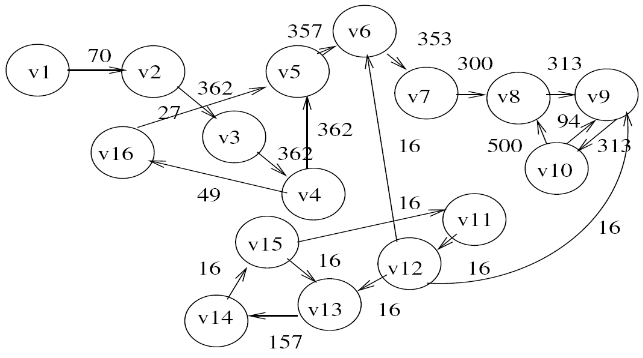
Figure 7. Block diagram of {4, 3, 2, 1}-bank routers.

TABLE 1. HW COST AND BANDWIDTH OF N-BANK ROUTERS

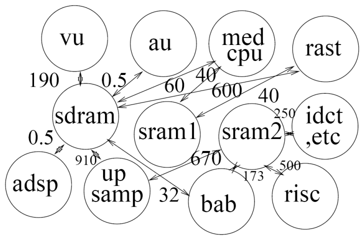
| n | Area | | Max. BW (MB/s) | |
|---|---|---|---|---|
| | um$^2$ | normalized | @ 200MHz | @ 66.7 MHz |
| 4 | 100000 | 100 % | 3200 | 1068 |
| 3 | 83400 | 83 % | 2400 | 801 |
| 2 | 66800 | 67 % | 1600 | 534 |
| 1 | 50200 | 50 % | 800 | 267 |
| 0 | 0 | 0 % | 0 | 0 |

TABLE 2. CHARACTERISTIC COMPARISON BETWEEN NMAP AND THE PROPOSED MAPPING AND OPTIMIZATION ALGORITHM.

| | NMAP[8] | BMAP |
|---|---|---|
| Algorithm | Initialization & Iteration | Greedy Binomial Merge |
| Optimization | Shortest Path | Low Cost |
| Complexity | O(N$^4$logN) | O(N$^2$logN) |

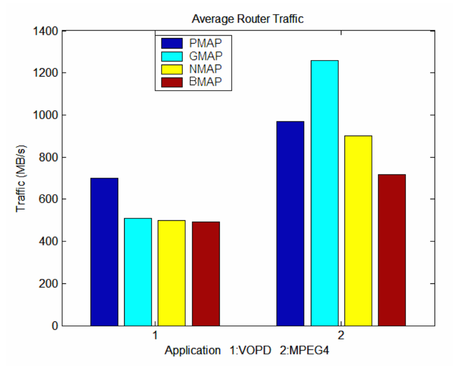


(a). Core Graph of VOPD.[9]



(b). Core Graph of MPEG-4.[9]

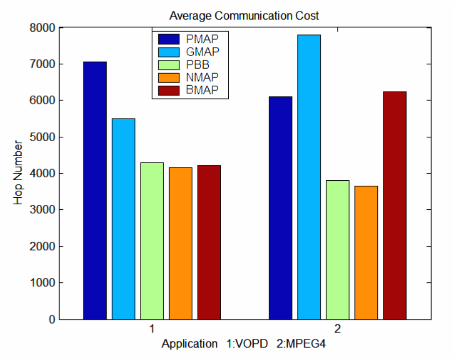
Figure 8. Core graphic extracted for traffic load modeling.



(a)Total traffic load chart.

| Average Router (Traffic) | | |
|---|---|---|
| | VOPD | MPEG4 |
| NAMP | 1 | 1 |
| BMAP | 0.98 | 0.80 |

(b) Total traffic load table.



(c)Hop count chart.

| Communication Cost (Hop) | | |
|---|---|---|
| | VOPD | MPEG4 |
| NAMP | 1 | 1 |
| BMAP | 1.02 | 1.71 |

(d) Hop count table.

Figure 9. Simulation results of binomial mapping of BMAP.

Traffic table (application 1)

| | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| 1 | 356 | 362 | 610 | 368 |
| 2 | 517 | 531 | 576 | 194 |
| 3 | 362 | 373 | 165 | 104 |
| 4 | 35 | 50 | 31 | 112 |

$\Rightarrow$

n-bank selection

| | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 |
| 2 | 1 | 1 | 1 | 1 |
| 3 | 1 | 1 | 1 | 1 |
| 4 | 1 | 1 | 1 | 1 |

(a) VOPD.

Traffic table (application 2)

| | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| 1 | 140 | 100 | 0 | 0 |
| 2 | 700.5 | 60.5 | 0 | 0 |
| 3 | 1872.3 | 1035.3 | 474.8 | 221.5 |
| 4 | 1580.5 | 1593.5 | 423.5 | 173.5 |

$\Rightarrow$

n- bank selection

| | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| 1 | 1 | 1 | 0 | 0 |
| 2 | 1 | 1 | 0 | 0 |
| 3 | 4 | 4 | 1 | 1 |
| 4 | 4 | 4 | 1 | 1 |

(b) MPEG-4.

Figure 10. Optimization Step of VOPD and MPEG-4.

TABLE 3. TRAFFIC RATIO AND HOP RATIO BETWEEN AMAP AND NMAP.

| NMAP vs. BMAP | | |
|---|---|---|
| Application | Traffic Ratio | HW Cost Ratio |
| VOPD | 98% | 51% |
| MPEG-4 | 80% | 85% |
| Average | 89% | 68% |

## V. CONCLUSIONS

We propose a binomial mapping and optimization algorithm, BMAP, based on our modified OCN synthesis flow in this paper. By using the proposed BMAP, we can save 37% total traffic load and 46% average HW costs. The binomial mapping costs $O(N^2log(N))$ computation complexity and is a fast and efficient algorithm compared to NMAP [9]. The binomial merging iteration results of BMAP is better than NMAP in traffic load. After mapping, the proposed BMAP adopts several approaches to optimize the hardware cost. From simulation results of real SoC applications, BMAP saves 50% hardware cost of the synthesized OCN by the router selection approach. We also use unfolding approach to increase bandwidth of OCN critical nodes to meet the throughput requirement Therefore the total hardware cost of network infrastructure is reduced to 51%~85%. The OCN architecture is verified through hardware-software co-simulation on the established infrastructure with CoWare ConvergenSC ESL design tool.

## REFERENCES

[1] J. A. Davis, R. Venkatesan, A. Kaloyeros, M. Beylansky, S. J. Souri, K. Banerjee, K. C. Saraswat, A. Rahman, R. Reif, J. D. Meindl, "**Interconnect Limits on Gigascale Integration (GSI) in the 21st Century**," *Proceeding of the IEEE*, vol. 89, no. 3, pp. 305-324, March. 2001.

[2] D. Sylvester, K. Keutzer, "**A Global Wiring Paradigm for Deep Submicron Design**," *IEEE Transaction on Computer-Aided Design of Integrated Circuits and Systems (CAD/ICAS)*, vol. 19, no. 2, pp. 242-252, Feb. 2000.

[3] L. Benini, G. De Micheli, "**Networks on Chips: A New SoC Paradigm**," *IEEE Computer*, vol. 35, no. 1, pp. 70-78, Jan. 2002.

[4] Jingcao Hu; Radu Marculescu, "**Energy-aware mapping for tile-based NoC architectures under performance constraints,**" *Asia and South Pacific Proceedings of the ASP-DAC on Design Automation Conference*, 21-24, pp. 233 – 239, Jan. 2003.

[5] Jingcao Hu, "**Design Methodologies for Application Specific Network-on-Chip**," *PhD Thesis, Carnegie Mellon University*, May. 2005.

[6] D. Bertozzi, L. Benini, "**Xpipes: A Network-on-Chip Architecture for Gigascale Systems-on-Chip**," *IEEE Circuits and Systems Magazine*, vol. 4, no. 2, pp. 18-31, 2004.

[7] M. Dall'Osso, G. Biccari, L. Giovannini, D. Bertozzi, L. Benini, "**Xpipes: A Latency Insensitive Parameterized Network-on-Chip Architecture for Multiprocessor SoCs**," *Proceedings of 21st International Conference on Computer Design*, pp. 536-539, Oct. 2003.

[8] S. Murali, G. De Micheli. "**SUNMAP: A Tool for Automatic Topology Selection and Generation for NoCs**," *Proceedings of 41st Design Automation Conference (DAC)*, pp. 914-919, 2004.

[9] S. Murali and G. De Micheli, "**Bandwidth Constrained Mapping of Cores onto NoC Architectures**," *Proceedings of Design, Automation and Test in Europe Conference and Exhibition (DATE)*, vol. 2, pp. 896-901, Feb. 2004.

[10] D. Bertozzi, A. Jalabert, S. Murali, R. Tamhankar, S. Stergiou, L. Benini, G. De Micheli, "**NoC Synthesis Flow for Customized Domain Specific Multiprocessor Systems-on-Chip,**" *IEEE Transactions On Parallel and Distributed Systems*, vol. 16, no. 2, pp. 113-129, Feb. 2005.

[11] Jingcao Hu, R. Marculescu, "**Energy- and Performance-Aware Mapping for Regular NoC Architectures,**" *IEEE Transactions on Computer-Aided Design of Integrated circuits and systems*, vol. 24, no. 4, pp. 551-562, April. 2005.

[12] Chae-Eun Rhee, Han-You Jeong, Soonhoi Ha, "**Many-to-Many Core-Switch Mapping in 2-D Mesh NoC Architectures**," *Proceedings of IEEE International Conference on Computer Design*, pp. 438-443, Oct. 2004.