# Power Optimization for Mesh Network-on-Chip Architecture: Multilevel Network Partitioning Approach

**[1]A. Lit, S. K. Sahari, R. Spawi, S. Suhaili. D.N.S.D.A. Salleh, [2]A. F. Z. Abidin and , [2]A. F. Hassan

**Abstract**

This paper presents a power optimization for mesh Network-on-Chip (NoC) architecture by using Multilevel Network Partitioning approach. Power consumption is reduced by re-dividing the large networks into few smaller partitions. This approach assigns excessively communicated Intellectual Property (IP) cores into the same portion that result the minimal average inter-core distance. The efficiency of this methodology is verified through a System-on-Chip (SoC) application known as Video Object Plan Decoder (VOPD). Experimental results show a promising improvement of 16.59% in the power consumption.

**Keywords:** Network-on-Chip, Power Optimization, Multilevel, Network Partitioning.

## I. INTRODUCTION

The fast growing complexity in System-on-Chip (SoC) designs inspires both academic and industrial researchers to investigate a better solution to solve an on-chip communication problem among IP cores. To date, with SoC designs that have hundreds of IP Cores, implementing on-chip communication using shared buses is an impractical solution. To address this problem, Networks-on-Chip (NoC) is proposed as an emerging paradigm to provide a better solution to an efficient on-chip communication infrastructure among the IP cores.

Optimizing the power consumption of NoC-based designs has become more crucial with the usage of high speed, complex ICs in mobile and portable devices [1]. Managing the power does not only target on the power reduction, but also ensures that all CUs provide an efficient amount of power to keep the application stable and reliable. Meanwhile, power constraints are among the major bottlenecks that limit functionality and performance of complex NoC-based designs [2]. Therefore, several techniques have been proposed to address the high-power dissipation problem from both circuit and system perspectives [3,4].

The selection of a network topology and the mapping of CUs applications greatly impact an NoC power consumption [2]. Hence, graph partitioning techniques have been proposed to enhance these NoC metrics including power dissipation [3]. In [4], a topology generation methodology to minimize the area consumption cost of NoCs was proposed. Multilevel partitioning may further improve NoC performance to fit the design necessities, while minimizing the cost.

## 2. Related Works

Application mapping in NoC is very challenging because of the increasing number of IP Cores with rich amount of data exchange [3]. Direct mapping of applications on large SoCs require the optimization of NoC's cost and performance. Moreover, SoCs are a combination of multiple networks and each core only connects with a small subset of cores. Network partitioning de-composes a large system to many smaller subsystems to optimize the metrics of interest. The standard of NoC topologies namely ring, octagon, mesh, and torus were compared and evaluated with respect to throughput, average latency, energy consumption and area in [6], [7]. Various mapping algorithms, namely PBB [8], GMAP [8], BMAP [9], and NMAP [10] were proposed to mapping out any SoC application onto those standard totpologies. In [11], a tool called SUNMAP was introduced to automatically choose the best standard topology for any given application.

**Corresponding author

[1] A. Lit is with the Universiti Malaysia Sarawak (UNIMAS), 94300 Kota Samarahan, Malaysia (Phone +6082-58-3319; Fax:+6082-58-3410; E-mail: lasrani@feng.unimas.my)

# 3. Graph Theory Concept

Graph theory concept is used to represent the topological structure of any interconnection network. The traffic criteria and application description are assumed to be statistically analyzed and identified. Therefore, any SoC application could be described by a traffic distribution graph (TDG) [10], [11]. A TDG describes the IP cores in the application as well as the communication traffic among them. A TDG incorporates of directed graph G(C,W), where each vertex $c_i \in C$ represents a computational core in the SoC application, and the directed edge $w_{ij} \in W$ is the communication between computational cores $c_i$ and $c_j$ (in terms of the number of packets per time step) as shown in Figure 1.

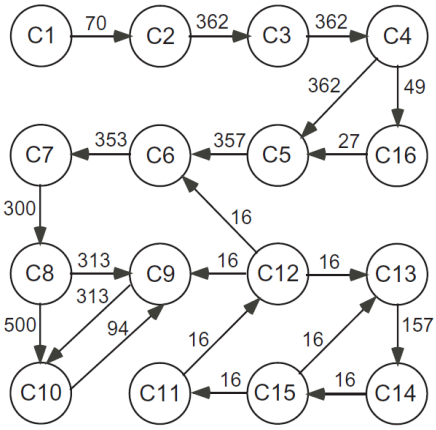



Figure 1: Traffic Distribution Graph for VOPD [9, 10]

In NoC term, traffic distribution matrix ($\lambda$) is another form of TDG [2]. For n computational cores in a particular SoC application, the dimension of the matrix is n $\times$n. The weight of the edge $w_{ij}$ is represented by $\lambda_{ij}$, which is the number of packets transmitted by computational core $c_i$ to computational core $c_j$ per time step. For instance, Equation (1) shows the representation of VOPD application in the traffic distribution matrix form [2].

$$\lambda = \begin{bmatrix}
0 & 70 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 362 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 362 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 362 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 49 \\
0 & 0 & 0 & 0 & 0 & 357 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 353 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 300 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 313 & 500 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 313 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 94 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 16 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 16 & 0 & 0 & 16 & 0 & 0 & 0 & 16 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 157 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 16 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 16 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 16 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix} \quad (1)$$

# 4. Problem Formulation

Fidducia-Mattheyses (FM) algorithm is selected due to its outstanding performance over other network partitioning methods [17]. To formulate the partitioning problem mathematically, n application cores are divided into m partitions $P_1$, ..., $P_m$. The number of IP cores within any partition $P_k$ is $n_k$. Meanwhile, the total traffic, $\lambda_{total}$, is in packets/time step. The communication among all application IP cores is defined as Equation (2) [3]:

$$\lambda_{total} = \sum_{i=1}^{n} \lambda_{ij} \quad (2)$$

For any partition $P_k$, the overall traffic within the partition $\lambda_k$, in packets/time step, is given by Equation (3) [3]:

$$\lambda_k = \sum_{i=1}^{n} \lambda_{ij} \; \forall i, j \quad \text{where } c_i, c_j \in P_k \quad (3)$$

The overall traffic $\lambda_{total}$ is based on the application itself. The intra-partition traffic $\lambda_k$ and the inter-partition traffic $\lambda_k$ are used by the FM partitioning scheme. As previously mentioned, the network partitioning is aimed at minimizing the total traffic among all partitions. Hence, the overall inter partition traffic $\lambda_{inter}$, in packets/time step, is given by Equation (4) [3]:

$$\lambda_{itr} = \sum_{k=1}^{m} \overline{\lambda}_k \quad (4)$$

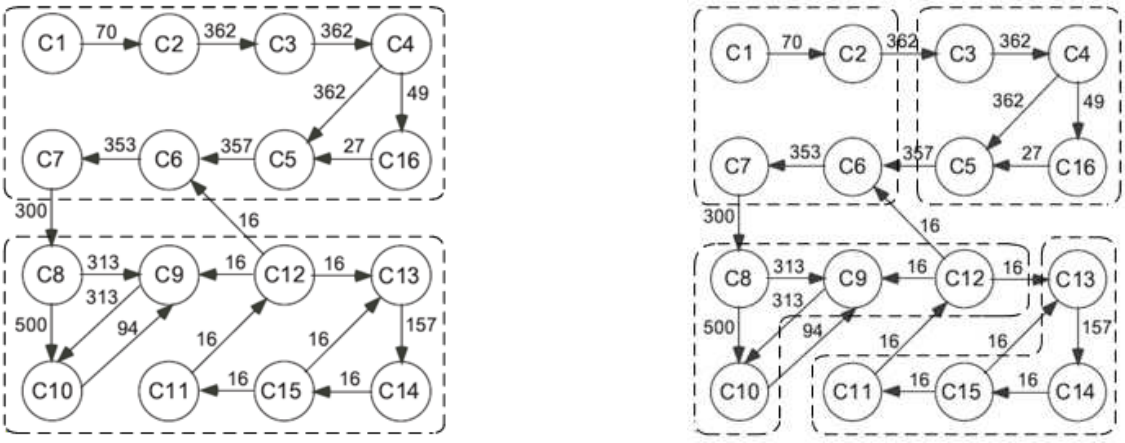
Every partition generates a set of cut edges, $E_c$ specified as the subset of E. The weight of each subset, $|S_i|$ is formed to be the number of vertices mapped to that subset by P. Let TDG as an input, the graph partitioning problem attempts to discover a p-way partitioning in which each subset consists approximately the equivalent number of vertices. Thus, $\left(|S_i| \le |V|/p\right)$ the number of cut edges, $|E_c|$, is minimized [3]. The cut edges constitute the inter processor communication needed by the distribution. As a result, the graph partitioning problem aims at a distribution that balances the computation handled by each processor while minimizing the entire inter processor communication [17].

# 5. Multilevel Network Partitioning Approach

Graph partitioning algorithms divide a large graph into a smaller networks [3]. The well-known graph partitioning algorithms is the Fiduccia-Mattheyses (FM) which is the derivative from the Kerninghan-Lin (KL) algorithm [12]. The FM algorithm [5] is used for the graph partitioning technique. The selection of FM is two-fold. First, it reduces the impact of partitioning on the average delay by minimizing the inter-partition traffic (cut-edge). Second, it prevents inducing communication bottleneck on the NoC based system. Hence, buffering requirement on inter-partition routers (located at the boundary of partitions) can be reduced.

Multilevel Network Partitioning re-divide the prior partitioning into several level so that heavily communicated nodes is clustered into the same portion. Hence, the partitioning before core placement allows better partitioning as the critical paths are in the same partition and easier for portion placement in network topology especially for complex SoC. With critical paths in the same partition, communication latency and energy can be reduced [3].

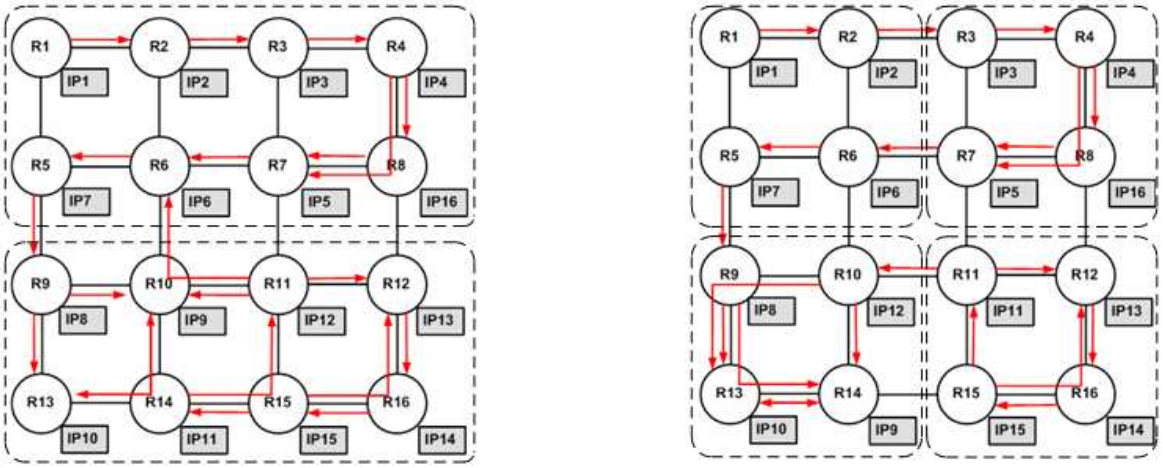
Figure 2 and Figure 3 show the partitioning at the traffic distribution and mesh NoC respectively. Since the numbers of IP cores are relatively small, the level of partitioning is carried out up to two level. Figure 2 shows how the partitioning is carried out at the TDG abstraction. As can be seen, Figure 2(a) is the first level and re-partition to second level as depicted in Figure 2(b).




(a)1st Level (b) 2nd Level

Figure 2: Traffic Distribution Graph Partitioning

Figure 3 shows the network partitioning at the NoC abstraction. After the partitioning at the TDG, the IP cores in mapped into the 4 x 4 mesh model as can be seen below.



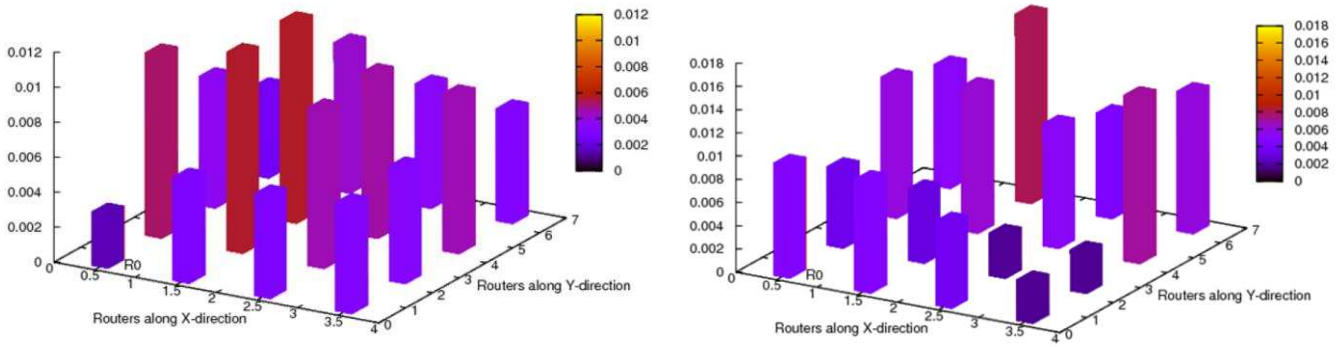

(a)1st Level (b) 2nd Level

Figure 3 : 4 x 4 Mesh NoC Network Partitioning

# 6. Simulation Result

This technique is applied to a VOPD application with 16 number IP Cores. The selection of this SoC application is because it delivers a rich amount of data. The level of partitioning is chosen to 2nd-level. NoC simulator software, NoC Interconnect Routing and Applications' Modelling (NIRGAM) is employed to carry out the simulation [19].

Figure 4 show the simulation results of power consumption between 1st level and 2nd level of partitioning. The improvement of power consumption in NoC based system is indicated by less power consumption and balanced distribution. As shown, the 2nd level of partitioning give a better result since it provides less power distribution among the IP Cores.




(a) 1st Level (b) 2nd Level

Figure 4: Power Consumption Simulations

Table 1 shows the power consumption estimation comparison between 1st level and 2nd level of Network Partitioning of VOPD. The result shows the power improvement with 16.59% from the 1st level to the 2nd level. Hence, from this experimental result can be concluded than the second-level is outperformed the first-level of network partitioning

Table 1: Power Consumption of the First-Level and Second-Level of Network Partitioning

| Tile ID | Power Estimation (mW) | |
|---|---|---|
| | First-Level | Second-Level |
| 0 | 9.659 | 3.042 |
| 1 | 9.564 | 5.914 |
| 2 | 7.284 | 5.821 |
| 3 | 3.508 | 6.198 |
| 4 | 6.787 | 10.426 |
| 5 | 6.198 | 11.323 |
| 6 | 3.662 | 9.021 |
| 7 | 3.508 | 6.571 |
| 8 | 11.896 | 7.356 |
| 9 | 12.433 | 11.415 |
| 10 | 10.486 | 9.324 |
| 11 | 14.246 | 8.978 |
| 12 | 10.426 | 5.066 |
| 13 | 16.065 | 8.438 |
| 14 | 8.835 | 6.952 |
| 15 | 12.028 | 6.408 |
| Total Power | 146.594 | 122.259 |

# 7. Conclusion

NoC is an emerging paradigm for the solution of NoC communication problem in the complex SoC applications. In this paper, the power consumption of NoC-based system is reduceed by using the Multilevel network partitioning approach. This technique is an efficient approach to reduce the NoC power consumption especially for the target application that have an immense number of IP Cores.

# Acknowledgment

The authors would like to express their sincere gratitude to Research and Innovation Management Centre (RIMC), Universiti Malaysia Sarawak (UNIMAS) for the financial support.

# References

[1] Benini, L. (2006),Application Specific NoC Design,Proceeding of *the IEEE Design, Automation, and Test inEurope Conference (DATE'06)*, vol. 1, Munich,Germany, pp. 1–5.

[2] Elmiligi H., Morgan A., El-Kharashi W. and Gebali F. (2007) A Topology-Based Design Methodology for Network-on-Chip Applications,Proceedings of the secon*d International Design and Test Workshop (IDT'07)*, Cairo, Egypt, pp. 61–65.

[3] Morgan A.,Elmiligi H., El-KharashiW. and GebaliF. (2008), Application Specific Network-on-Chip Topology Customization using Network Partitioning,Proceedings of the First International Forum on Next- *Generation Multicore/Manycore Technologies (IFMT'08)*, Cairo, Egypt, pp.13:1–13:6.

[4] Elmiligi H., Morgan A., El-Kharashi M. W.and GebaliF. (2008), Power Aware Topology Optimization for Network-on-Chips, Proceedings of the IEEE International *Symposium on Circuits and Systems (ISCAS'08)*, Seattle, WA, USA, May 18-21, 2008, pp. 360–363.

[5] Fiduccia, C. and Mattheyses, R. (1982), A Linear-Time Heuristic for Improving Network Partitions , Proceedings of the 19th Annual IEEE/ACM Design Automation Con*ference (DAC'82)*, Las Vegas, NV, USA, 1982, pp. 175–181.

[6] Benini, L. and Concer, N. (2006), Simulation and Analysis of Network-on-Chip Architectures : Ring, spidergon and 2DMesh,Proceeding of the IEEE *Design, Automation and Test in Europe Conference (DATE'06*), vol. 2, Munich, Germany, Mar. 6-10, 2006, pp. 154–159.

[7] Pande, P., Grecu, C., Jones, M., Ivanov, A., and Saleh,R. (2005), Performance Evaluation and Design Trade-offs for Network-on-Chip Interconnect Architecture, IEEE Transactions on Computers, vol. 54, no. 8, pp. 1025– 1040.

[8] Hu, J. and Marculescu, R. (2003), Energy-Aware Mapping for Tile-based NoC Architecture under Performance Constraints, Proceeding of the IEEE Asia and South Pacific Design Automation Conference (ASP-*DAC'03)*, Kitakyushu, Japan, pp. 233–239.

[9] Shen, W., Chao, C., Lien, Y. and Wu, A. (2007), A New Binomial Mapping and Optimization Algorithm for Reduced-Complexity Mesh-Based on Chip Network, Proceeding of the IEEE International Symposium on Network-on-*Chip (NOCS'07)*, Princeton, NJ, USA, pp. 317–322.

[10] Murali, S. and Micheli,G.D. (2004),Bandwidth-constrained Mapping of Cores onto NoC Architectures, Proceeding of the IEEE Design, Automation *and Test in Europe Conference and Exhibition (DATE'04)*, vol. 2, Paris, France, pp. 896–901.

[11] MuraliS. and MicheliG.D. (2004), SUNMAP: A Tool for Automatic Topology Selection and Generation for NoCs, Proceedings of 41st IEEE/ACM *Design Automation Conference (DAC'04)*, San Diego, CA, USA, pp. 914–919.

[12] Kerninghan, B. and Lin, S. (1970), An efficient Heuristic Procedure for Partitioning Graphs, Bell System Technical Journal, vol. 49, no. 2, pp. 291-307.

[13] Chan, P., Schlag, M. and Zien, J. (1994), Spectral k-way Ratio-Cut Partitioning and Clustering, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 13, no. 9, pp. 1088–1095.

[14] Ahonen, T., Tortosa,D. S., Bin, H. and Nurmi, J. (2005), Topology Optimization for Application-Specific Networks-on-Chip, Proceedings of the 2005 ACM International Workshop on System Level Interconnect Prediction *(SLIP'04)*, Paris, France, pp. 53–60.

[15] Coenen, M., Murali, S., Radulescu, A., Goosens, K. and Micheli, G. D. (2006), A Buffer-Sizing Algorithm for Networks-on-chip using TDMA and Credit Based end-to-end Flow Control, Proceeding of the Third IEEE/ACM/IFIP International Conference on Hardware/Software Codesign and *System Synthesis (CODES+ISSS'06)*, Seoul, Korea, pp. 130–135.
[16] Dumitriu, V. and Khan, G. (2009), Throughput-oriented NoC Topology Generation and Analysis for High Performance SoCs, IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 17, no. 10, pp. 1433–1446.
[17] Chamberlain, B. (1998), Graph Partitioning Algorithms for Distributing Workloads of Parallel Computations, Technical Report UW-CSE-98-10-03, University of Washington, WA, USA,.
[18] Karypis, G.,Schloegel, K., and Kumar, V. (2003), PARMETIS: Parallel Graph Partitioning and Sparse Matrix Ordering Library, Version 3.1,Tech. Rep. University of Minnesota, Minneapolis, USA.
[19] Lavina, J., Gaur, M.S., Al-Hashimi B. M., Laxmi, V. and Navaneeth, R. (2007), NIRGAM : A Simulator for NoC Interconnect Routing and Applications' Modeling, University of Southampton, UK.