

To Overcome Network Problem using Network-on-chip

Name:- Parshav Jain
Teerthanker Mahaveer University, Moradabad
aj.parshav2201@gmail.com
Guide Name:-Dr.Arpit Jain

Abstracts--Network-on-Chip (NoC) is a newly introduced paradigm to overcome the communication problems of System-on-Chip architectures. Mapping applications onto mesh-based NoC architecture is an NP-hard problem and several heuristic methods have been presented to solve it so far. Scalability is the main problem of the heuristic methods and it is very difficult to conclude that one heuristic is better than the others. Integer Linear Programming (ILP) based methods determine the optimum mappings. However, they take very long execution times. In this paper, we propose a clustering based relaxation for ILP formulations.

INTRODUCTION

The custom on-chip network, which targets a given application, has proved to be more efficient than the regular structure on-chip network design in [2]. The reason is that the communication requirement for each data flow is available in the design time, so the power consumption and packet latency are predictable once the links of networks are determined. Having this knowledge makes custom on-chip networks more efficient with topology synthesis as shown in [1], [2], [7]-[12]. Among those, [1], [2], [9] use a partition-based algorithm to reduce the time complexity. MPSoC are now integrating more and more processors on a single die with the increase in transistor budgets enabled by Moore's law. As the number of processor cores on a single die increases, the power consumption and wire delay will also have a significant increase, which makes the on-chip communication among cores becomes a critical issue. A scalable, energy efficient on-chip interconnect

network is needed to address these difficulties in order to facilitate the on-chip communication named AT_NOC, to synthesize the topology by creating a series of refinement steps from an initial topology design, in which any pair of communicating nodes are connected by links. In each refinement, we pick the router that has the highest power consumption among the un-refined routers, and refine the sub-topology of this router to reduce the power consumption and keep the packet latency lower. Many previous topology synthesis approaches use a Steiner-tree-based topology construction (such as [2]). The reason we use an A Tree instead of the Steiner-tree is that an A Tree is the shortest path tree that can reduce the packet latency. By using the A Tree-based algorithm, in the sub-topology of the chosen router, the paths from all nodes to the chosen router are the shortest paths. Additional routers might be added into these shortest paths at some Steiner points. The locations of these added routers are chosen to minimize a cost function, which considers both power and packet latency. In the cost function, the power numbers of routers with different port sizes are measured by ORION [5], [6]. In order to predict the packet latency between any two nodes, we build a latency prediction model by noting that the latency between two routers is related to the injection rates of packets at the routers [7]. The latency prediction model is built on a latency-injection rate table, which is constructed by a simple simulation. For different

injection rates, we randomly inject packets into the wire and measure the average packet latency. By looking up this latency injection rate table, we can easily predict the packet latency under different traffic.

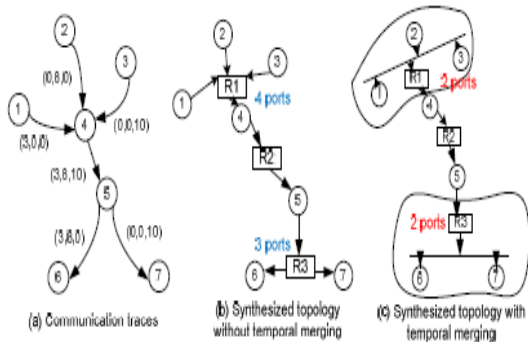


Figure 1. Temporal merging based on communication traces

In this paper, we present a cluster-based ILP formulation for application mapping problem for mesh-based NoC architectures. Our method partitions the task graph, representing the given application, and the mesh into smaller sub-graphs and sub-meshes to decompose the given solution space into smaller polyhedral. It then maps each sub-graph onto the corresponding sub-mesh using our ILP-based method. Finally, it merges each mapping to determine the final solution. We implemented our method using commercial ILP tool [13] and tested its effectiveness on several multimedia benchmarks and randomly generated graphs. Our experiments show that the proposed method is very effective to reduce the execution times of ILP method while determining similar results.

Network-on-Chip Architecture and Function Layers:-

Network on chip is the term used to describe an architecture that has maintained readily designable solutions in face of communication-centric trends. In

this section, we will briefly review some concepts on the design of an NoC communication system. Moreover, the NoC function can be classified into several layers, which will be introduced sequentially.

2.1. Network-on-Chip Architecture.:

A typical NoC architecture consists of multiple segments of wires and routers as shown in Figure 1. In a tiled, city-block style of NoC layout, the wires and routers are configured much like street grids of a city, while the clients (e.g., logic processor cores) are placed on city blocks separated by wires. A network interface (NI) module transforms data packets generated from the client logic (processor cores) into fixed-length flow-control digits (flits). The flits associated with a data packet consist of a header (or head) flit, a tail flit, and a number of body flits in between. This array of flits will be routed toward the intended destination in a hop-by-hop manner from one router to its neighbouring router. Usually, for each data packet, the corresponding head flit specifies its intended destination. After examining the head flit, the router control logic will determine which output direction to route all the subsequent (body and tail) flits associated with this data packet according to the routing algorithm applied. In a city-block style NoC, each router has five input ports and five output ports corresponding to the north, east, south, and west directions as well as the local processing element (PE). Each port will connect to another port on the neighboring router via a set of physical interconnect wires (channels). The router's function is to route flits entering from each input port to an appropriate output port and then toward the final destinations. To realize this function, a router is equipped with an input buffer for each input port, a 5×5 crossbar switch to redirect traffic to the desired output port and necessary

control logic to ensure correctness of routing results as shown in Figure 2.

2.2. Network-on-Chip Function Layers.

The No. C function can be classified into several layers: application, transport, network, data link, and physical layers. An NoC router should contain both software and hardware implementations to support functions of the layers.

2.2.1. Application Layer- At the application layer, target applications will be broken down into a set of computation and communication tasks such that the performance factors like energy and speed can be optimized. Placement of cores on anNoC has to be optimized to reduce the amount of total communication or energy but at the same time recognizing the limitations of any one particular link. The task mapping and communication scheduling problem is an instance of a constrained quadratic assignment problem which was known to be NP-hard [12]. Given a target application described as a set of concurrent tasks with anNoC architecture, the fundamental questions to answer are (1) how to topologically place the selected set of cores onto the processing elements of the network and (2) how to take into consideration the complex effects of network condition, which may change dynamically during task execution, such that the metrics of interest are optimized [13]. To get the best trade off between power and performance, application mapping and scheduling should be considered with several kinds of architecture parameters

2.2.2. Transport Layer. To prevent buffer overflow and to avoid traffic congestion, some management schemes should be applied to guide the transport of packets in an No. C. The transport layer addresses the congestion and flow control issues [14]. Key performance metrics of an No. C

include low packet delivery latency and high throughput rate, and these metrics are critically impacted by network congestions caused by resource contentions. Accordingly, contention resolution is a key to avoid network congestions [14]. One of the most crucial issues for the contention resolution is, under a premise of a deadlock- and live lock-free routing algorithm, to enhance the utilization efficiency of available network resources in order to come up with a better communication performance.

2.2.3. Network Layer- Network topology or interconnect architecture is an important issue in this layer, which determines how the resources of network are connected, thus, refers to the static arrangement of channels and nodes in an interconnection network. Irregular forms of topologies can be derived by mixing different forms of communication architectures in a hierarchical, hybrid, or asymmetric way by clustering partition, which may offer more connectivity and customizability at the cost of complexity and area. In addition, optimization of a topology, which affects the connectivity of the routers and the distance of any one core to the other, is difficult. Furthermore, the trade off between generality and customization that, respectively, facilitate scalability and performance is important. As future designs be come more complex, the non-recurring costs of architecting and manufacturing a chip will become more and more expensive. A homogenous NoC is one where the cores and routers are all the same, while a heterogeneous NoC selects individual cores from an IP library and may have its communication architecture customized to suit the needs of an application. Since NoC designs must be flexible enough to cover a certain range of applications, most of the state-of-the-art NoC designs use a mesh or torus topology because of its performance benefits and

high degree of scalability for two-dimensional systems, yet it may not achieve the best performance for a single application [15, 16]. Conventional design of a router consists of circuit switched fabrics and an arbitration controller. In each arbitration decision, more than one path can be constructed by the crossroad switch as long as no contention exists between these paths. For most existing switch designs, virtual-channel flow-control-based router design, which provides better flexibility and channel utilization with smaller buffer size, is a well-known technique from the domain of multiprocessor networks [17–24].

2.2.4. Data Link and Physical Layers.

The main purpose of data-link layer protocols is to increase the reliability of the link up to a minimum required level, under the assumption that the physical layer by itself is not sufficiently reliable [14]. The emphasis on physical layer is focused on signal drivers and receivers, as well as design technologies for resorting and pipelining signals on wiring. In addition, as technology advanced to ultradeep submicron (DSM), smaller voltage swings and shrinking feature size translate to decreased noise margin, which cause the on-chip interconnects less immune to noise and increase the chances of nondeterminism in the transmission of data over wires (transient fault) [2, 25–28]. Electrical noise due to crosstalk, electromagnetic interference (EMI), and radiation-induced charge injection will likely produce timing error and data errors and make reliable onchip interconnect hard to achieve.

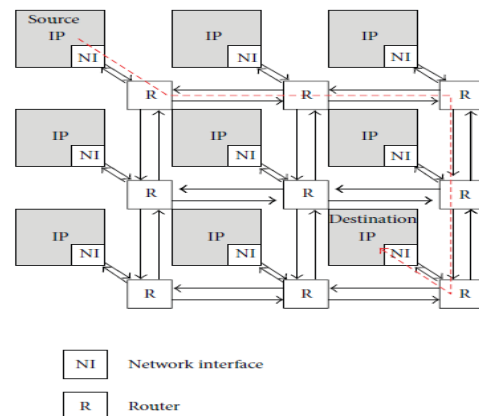


Figure 1: Typical NoC architecture in a mesh topology.

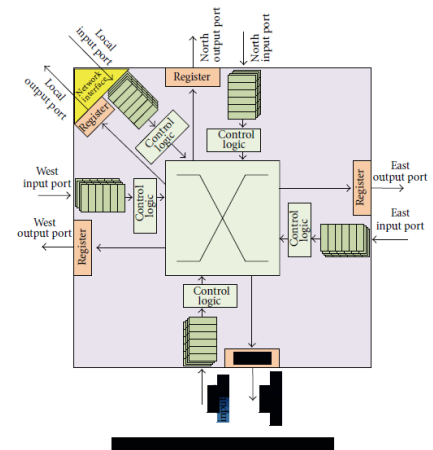


Figure 2: Typical NoC router architecture.

CLUSTER BASED METHOD:-

Our ILP-based mapping tool obtains optimum results for the stated problem. However, when the number of tasks in the application graph increases the computation time also increases dramatically, which makes this method inapplicable in practice. To remedy from this timing problem, we propose a cluster-based method to reduce the computation times. In this method, we decompose the mapping problem into several sub-problems and solve each sub-problem separately. In other words, we first decompose the mesh into sub-meshes and partition the application graph into smaller

sized clusters. We then apply our ILP-based mapping method for each cluster-sub-mesh pairs. The proposed clusterbasedmapping method follows the steps given below.

4.1. Mesh partitioning

Mesh dimensions are the designer guided inputs to our system. The system designer inputs X_{dim} and Y_{dim} values such that $j \in V, j \in T, j = X_{dim} \times Y_{dim}$. Our mesh partitioning method is a recursive procedure. It cuts the given mesh M into two sub-meshes M_1 and M_2 and it continuously cuts the sub-meshes until each partition has at most t tiles. The stopping criteria t is a predefined value. From our experiments, we observed that our ILP-based system can map around 12 tasks in a tolerable time. Therefore, we used $t = 12$ in our experiments. The effects of the granularity of the partitions are twofold: If the number of nodes in a partition is big, then the ILP solution times increase dramatically. On the other hand, if we partition the graph into very small portions, then the final solution becomes far from the optimal one since the number of mapping options for each node decreases.

$$X_{dim1} = \frac{1}{4} dX_{dim} = 2e; \delta 12P$$

$$X_{dim2} = \frac{1}{4} X_{dim} - X_{dim1} \delta 13P$$

$$Y_{dim1} = \frac{1}{4} Y_{dim} = 2e; \delta 14P$$

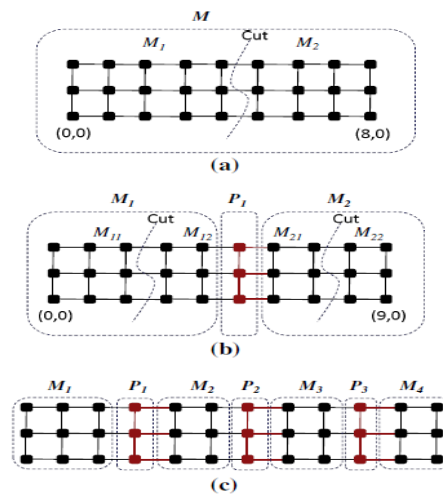


Fig. 2. Mesh partitioning example. (a) Initial mesh and the cut, (b) The sub-meshes after the first cut, and (c) result. In the figures, M_i represents a sub-mesh and P_j represents dummy tiles connecting two sub-meshes.

4.2. Graph clustering

Our graph clustering follows the similar recursive steps with the mesh partitioning. We recursively divide the application graph into clusters until each cluster contains the nodes less than the tiles of corresponding sub-mesh. We modified Kernighan–Lin algorithm [15] for our partitioning method. Before elaborating the details of our clustering method, we give some definitions as follows.

Definition 3. A cut $CGA;GB$ of graph $G(V,E)$ is a partition of V into GA and $GB = V - GA$. The cut degree $d_{GA;GB}$ is the total number of edges crossing the cut $CGA;GB$. The cut degree $d_{GA;GB}$ of the cut $CGA;GB$ can be determined by using the equation $d_{GA;GB} = \sum_{vi \in GA, vj \in GB} X_{8ei;j} \delta 16P$ where $X_{8ei;j}$ is a binary variable and it becomes 1 when $vi \in GA \wedge vj \in GB$. Otherwise, it is 0.

Definition 4. A node v_i in cluster GA is called a free node if v_i does not have any neighbor node in a dummy node set D and FA is the set of the free nodes in GA. Formally, $FA = \{v_i | \exists e_{i,j}, (v_j \in D)\}$.

Definition 5. A node v_i in cluster GA is a bound node if there is atleast one node v_j connected to v_i , such that v_j is a dummy node in a dummy node set D and BA is the set of the bound nodes in GA. Formally, $BA = \{v_i | \exists e_{i,j}, (v_j \in D)\}$. We define the free and bound nodes in a cluster to decide which nodes are candidates to be swapped between two clusters. By only moving the free nodes between clusters, we force the neighbour nodes of the CTG to be either in the same cluster or in the neighbour clusters. The rationale behind this idea is to increase the probability that they may be mapped in one hop distance. Initially, all nodes are free nodes since there is no dummy node set before the partitioning starts. For each cluster, the sum of free and bound nodes gives us the total nodes in a cluster. That is, $GA = FA \cup BA$. In Fig. 3a, all the nodes are free nodes, whereas v_9 , v_{17} , v_8 , and v_{16} are bound nodes in Fig. 3b.

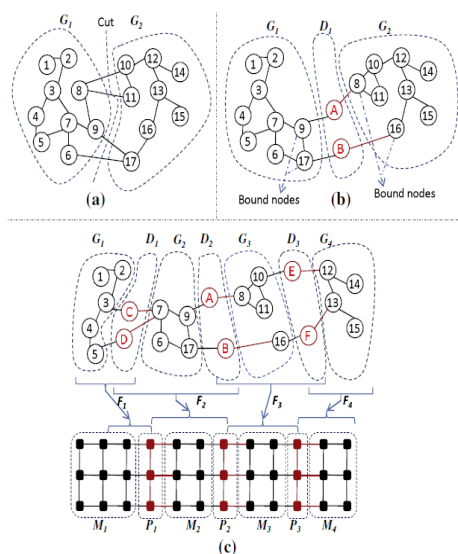


Fig. 3. Graph clustering.

5. Experimental results

We evaluated our cluster based method by comparing it with ILP solutions on different real benchmarks and randomly generated graphs. Before presenting the results, we demonstrate the working procedure of our cluster based method. For this demonstration, we used ILP and cluster-based methods to map multimedia benchmark 263-enc mp3-dec onto 4×3 mesh. We present both these mapping in Fig. 4. In this figure, (a) presents CTG of multimedia benchmark 263-enc mp3-dec with weights given in kbits/s and (b) shows ILP-based mapping onto 4×3 mesh, which took 1292 s resulting in 230,407 kbits/s total communication cost. Fig. 4c and d shows the mesh partitioning and graph clustering steps, respectively. Finally, Fig. 4e and f present the mapping and merging results, respectively. Our mappings took 0.9 s for the first cluster and 0.3 s for the second, resulting in a total communication cost of 230,426 kbits/s. As the numbers illustrate, we achieved tremendous time savings while we sacrificed 0.01% on the optimum result.

We tested our cluster based method on six multimedia benchmarks and four custom generated graphs. We give the results in Table 2. We list the name and the number of nodes of these graphs in the first two columns of Table 2. Columns three and four give the total communication costs (CommCost) of ILP-based and cluster based methods, respectively. Note that we limit the running time of ILP tool to 8 h and accept its best solution returned within this time limit. In our six experiments, ILP-based method could not obtain the solutions in these time limits. We show these solutions with the t.o. (timeout) in column six of Table 2. In column five of Table 2, we give the difference between our cluster based method and ILP-based method in percentages.

Application	Number of tasks	Total communication (Mbits/s)		Difference from ILP (%)	CPU time (s)	
		ILP	Cluster		ILP	Cluster
VODP [5]	16	4119	4205	2.01	T.o.	114.6
MWD [18]	12	1184	1184	0.0	380	0.3
MPEG4 [18]	12	3567	3567	0.0	7750	2.6
263 Dec. [16]	14	19.823	20.098	1.38	T.o.	34.8
263 Enc. [16]	12	230.407	230.426	0.001	5825	1.2
Mp3 Enc. [16]	13	17.021	17.021	0.0	8476	17.4
Graph 1	20	1856	1867	0.59	T.o.	198.8
Graph 2	25	2874	2874	0.0	T.o.	286
Graph 3	30	4572	4613	0.89	T.o.	425
Graph 4	35	5128	5096	-0.62	T.o.	562

TABLE 2: Experimental results on multimedia benchmarks and custom generated graphs

CONCLUSION

- We proposed a new cluster based technique for application mapping onto NoC structures.
- Our cluster-based mapping method can also be used to map the tasks in such a way that the communication is distributed over the chip evenly.
- Low cost to make chips for communication.

REFERENCES

- [1] S.R. Vangal, et al., An 80-tile sub-100-WteraFLOPS Processor in 65-nm CMOS, IEEE J. Solid-State Circuits 43 (1) (2008) 29–41.
- [2] A. Agarwal et al., Tile processor: embedded multicore for networking and multimedia, in: Proceedings of the IEEE Hot Chips 19, Stanford, CA, USA, August 2007.
- [3] W.J. Dally and B. Towles, Route packets, not wires: on-chip interconnection networks, in: Proceedings of the 38th Design Automation Conf. (DAC 2001), June 2001, pp. 683–689.
- [4] L. Benini, G. De Micheli, Networks on chips: a new SoC

paradigm, IEEE Computer (2002) 70–78.

[5] B. Grot, et al., Express cube topologies for on-chip interconnects, HPCA 2009 (2009) 163–174.