Many-to-Many Core-Switch Mapping in 2-D Mesh NoC Architectures

Chae-Eun Rhee
SoC R&D Center
Samsung Electronics Co., Ltd
Suwon, Korea
ce.rhee@samsung.com

Han-You Jeong
EECS/EE
Seoul National University
Seoul, Korea
hyjeong@ccs.snu.ac.kr

Soonhoi Ha
EECS/CS
Seoul National University
Seoul, Korea
sha@iris.snu.ac.kr

Abstract

In this paper, we investigate the core-switch mapping(CSM) problem that optimally maps cores onto an NoC architecture such that either the energy consumption or the congestion is minimized. We propose a many-to-many core-switch mapping(mCSM) that allows a switch(core) to have multiple connections to its adjacent cores(switches). We also present decomposition methods that can obtain the suboptimal solutions with enhanced computational efficiency. Our work is the first to provide an exact mixed-integer linear programming(MILP) formulation for the complete CSM problems, including the optimal choice of core placements, switches for each core, and network interfaces for communication flows. Experiments with four random benchmarks show that 4:4 mCSM achieves 81.2 % of energy savings and 2.5 % of bandwidth savings compared with one-to-one mapping. They also show that, for one-to-one mapping, our optimal solutions obtained by the full MILP save 34.8 % of energy consumption and 34.4 % of bandwidth requirement compared with those from the existing algorithms.

1. Introduction

Continuing advance of the semiconductor technology will enable us to integrate billions of transistors on a single chip by the end of the decade. With this trend, the future *System-on-Chips(SoCs)* will face new challenges in overall areas of VLSI technologies, such as design, synthesis, verification, test, etc. To meet these challenges under the strong time-to-market pressure, it is essential to increase the reusability of cores(or components) and system architectures, and the ability to interconnect the existing cores in a plug-and-play fashion[1]. Simultaneously, the SoC design methodology must address 1) the system-level synthesis through HW/SW partitioning and mapping of the application tasks onto the pre-designed cores, and 2) the communication-architecture synthesis through mapping of the cores onto the communication architecture[2]. Of the two problems, the latter becomes more important due to the following reasons: 1) As the integration level of an SoC system scales up, the volume of data and control traffic among the cores grows increasingly and becomes the major performance bottleneck; 2) Unlike the power consumed within a core, the power dissipated in communication architecture does not benefit from down-scaling of feature size[3]. Therefore, the focus of this paper is on the communication-architecture synthesis.

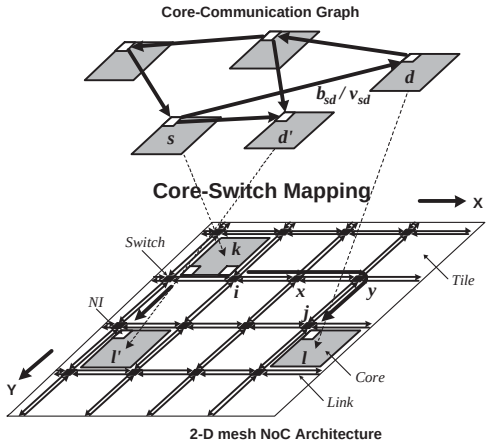

Figure 1. Mapping of cores onto an NoC architecture

The communication-architecture synthesis consists of defining a communication topology and mapping cores onto the communication architecture. Despite its simplicity and widespread deployment, the current bus-based architectures will not meet the requirements of the future SoCs because of their seriously limited scalability and energy inefficiency. An alternative and promising approach is to use the *Network-on-Chip(NoC)* architecture which well exploits the advantages of the interconnection networks, such as the structured network wiring, modularity, and scalability.

Given the communication requirements of cores, in this

paper, we address the problem of optimally mapping the cores onto an NoC architecture with one of two possible objective functions: 1) to minimize the *average hop distance* and 2) to minimize the *maximum link bandwidth* of the NoC architecture; Both of the metrics are the key parameters of the performance and cost, e.g. the energy consumption and the communication delay. We refer to this problem as the *core-switch mapping(CSM)* problem(See Fig. 1).

We propose a *many-to-many core-switch mapping(mCSM)* that allows a switch(core) to have multiple connections to its adjacent cores(switches), each of which is directed to a different one. The mCSM encompasses one-to-one, one-to-many, and many-to-one mapping, depending on the number of such connections, while all of the previous works consider only one-to-one mapping[5]-[8] to our best knowledge. To quantify how much the mCSM can improve the performance compared with one-to-one mapping, we formulate the complete CSM problem as a *mixed-integer linear programming(MILP)* problem that can be solved by an LP optimization packages[14]. Unfortunately, the full MILP problem is usually computationally prohibitive, especially for large-scale CSM problems. To mitigate this difficulty, we propose a decomposition method that divides the CSM problem into two subproblems: the core-tile mapping(CTM) and the tile-switch mapping(TSM) problem. In particular, the previous CSM algorithms can be used to obtain the CTM solution in the proposed decomposition technique. Experiments with four random benchmarks show that 4:4 mCSM achieves approximately 81.2 % of energy savings and 2.5 % of bandwidth savings compared with 1:1 CSM, both on average. The optimal solutions obtained by the full MILP formulation also reduce 34.8 % of energy consumption and 34.4 % of bandwidth requirement, when compared to those from the existing algorithms.

This paper is organized as follows. We review the related works in Section 2. Section 3 describes the communication architecture and the energy model assumed in this paper. Then, we present the full MILP formulation of the mCSM in Section 4. We also provide our decomposition method in Section 5. Experimental results are discussed in Section 6. Finally, a conclusion is given in Section 7.

2. Related Work

The CSM problem has recently received a considerable attention in the literature. In [5], Hu *et al.* proposed a heuristic(greedy) branch and bound algorithm to solve the CSM problem in 2-D mesh topologies; They aimed to minimize the energy consumption under bandwidth constraints. They also extended their algorithm to the non-bifurcated deadlock-free routing strategies in [6]. In [8], Lei *et al.* proposed a two-step genetic algorithm that finds a mapping of cores onto NoC architecture such that the overall execution time is minimized. To obtain the mapping solution that minimizes the average communication delay, Murali *et al.* proposed the NMAP algorithm that partly utilizes linear formulation for modeling the bifurcated routing[7].

However, all of the previous works in [5]-[8] consider only one-to-one mapping, and do not guarantee the optimality of their solutions. Our work is the first to address many-to-many core-switch mapping for the NoC architectures, and provide a complete formulation of the CSM problems optimizing either the energy consumption or the congestion. While the above heuristics are fast and scalable, the advantage of our approach over those methods is the ability to guarantee that the obtained solution is the *global optimum value*. Thus, the optimality of any heuristic solutions can be evaluated by comparing them with our optimum solution. We also provide an efficient decomposition heuristic that is suitable for large-scale CSM problems.

3. Platform Overview

In this section, we briefly describe the NoC architecture and the energy/delay model used throughout this paper.

3.1. Core-Communication Graph

A Core-Communication Graph(CCG) is a *directed* graph, $G(V, E)$, where each vertex $s \in V$ represents a core and a directed edge $s \rightarrow d (d \in V)$ represents the *communication flow* from core s to core d. Each edge $s \rightarrow d$ has two attributes, denoted by b_{sd} and v_{sd}, where the former represents the required bandwidth, and the latter the total volume of communication data between s and d. Fig. 1 shows an example of CCG consisting of 5 vertices and 6 edges.

3.2. NoC Architecture

Topology: We consider the *2-D mesh* NoC architecture as an instance of the communication architecture, because they have several desirable properties, such as regularity, concurrent data transmissions, and controlled electrical parameters[1][4]. Fig. 1 shows an example of the 2-D mesh NoC architecture, consisting of 16 tiles, 16 switches, and 16 (unidirectional) links, where each tile is a square surrounded by four switches and links.

Switch: Each switch is connected to its neighboring switches via two opposite unidirectional links. The mCSM allows a switch to have *multiple* internal links connected to the cores placed at its adjacent tiles. Let T_i be the set of tiles that switch i can connect to. For example, $T_i = \{t_1, t_2, t_3, t_4\}$, in case of Fig. 2 (a). Since a switch can have up to four adjacent tiles in 2-D mesh topologies($|T_i| \leq 4$), the size of a switch varies from 4×4 to 8×8.

Multiple packets may contend for the same output port of a switch. To prevent the packet loss by this contention, we assume that each switch has small buffers which are implemented with registers.

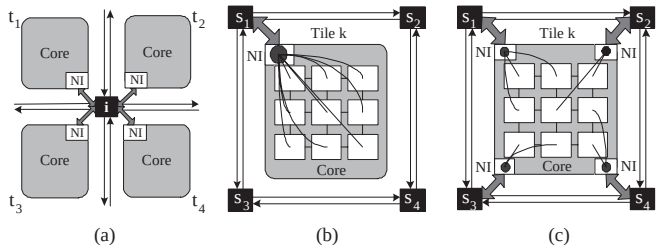

Figure 2. Example of mCSM

Core: A core is a resource that is placed onto a tile. It can be an IP, DSP, processor, memory, or arbitrary combinations of these blocks as shown in Fig. 2. To communicate with others, a core should have access points to the on-chip network, named the *network interfaces(NIs)*. The NI is placed between a core and a switch and has the responsibility of packetizing and depacketizing the communication data.

If a core with large traffic volume has a single NI as shown in Fig. 2 (b), it can easily be the performance bottleneck. In [9], a distributed-memory system has been proposed to reduce the communication overhead in multiprocessor communications, but it is still vulnerable to the bandwidth bottleneck at the *single* NI. On the other hand, the mCSM allows a core to have *multiple* NIs, each connected to a different adjacent switch. Let S_k be the set of adjacent switches that a core at tile k can directly connect to, where $S_k = \{s_1, s_2, s_3, s_4\}$ in Fig. 2 (c). Having multiple NIs also enables a core to transmit/receive multiple packets concurrently, and reduces the average hop distance of the network. The drawback of the mCSM is that it requires an additional area, control logic, and the possible change in core-design methodology, but these overheads can be alleviated by the wrapper-based NI design[10].

Routing: We assume *static XY wormhole routing* in this paper, because 1) it is easily implemented on the switches; 2) it does not require packet reordering buffers at the destination NI; 3) it significantly reduces the size of buffers at each switch; and 4) it is free of deadlock and livelock[5][11].

3.3. Energy Model

Energy minimization is the utmost goal of state-of-the-art SoCs in many cases. In [12], Ye *et al.* first defined the *bit energy* metric of a router as the energy consumed when a single bit of data goes through the router. In [5], Hu *et al.* modified the bit energy model so that it is suitable for 2-D mesh NoC architectures. In this model, the bit energy of a data from switch i to switch j is given by

$$E_{bit}^{i,j} = (h_{ij} + 1) \times E_{S_{bit}} + h_{ij} \times E_{L_{bit}}, \qquad (1)$$

where $E_{S_{bit}}$ and $E_{L_{bit}}$ are the energy consumed on the switches and the links, respectively. The variable h_{ij} denotes the hop distance between switch i and j, which is defined as the number of *links* on the shortest path.

Since Eq. (1) is a linear equation of variable h_{ij} and constants $E_{S_{bit}}$ and $E_{L_{bit}}$, we assert that minimizing the average hop distance yields the same mapping results as minimizing the energy consumption, regardless of the constant values. Hence, in the following, we first optimize the hop-distance metric instead of the energy consumption, and then calculate the latter using Eq. (1).

Note that the bit energy consumed at a switch $E_{S_{bit}}$ depends on the number of its ports. Therefore, increasing the number of internal links may result in an adverse effect to the energy minimization. But, as will be shown later, this adverse effect is insignificant, since energy consumption of the NoCs tends to be dominated by $E_{L_{bit}}$ rather than $E_{S_{bit}}$.

4. Problem Formulation

This section presents full MILP formulation for the mCSM using the following notations(See also Fig. 1):

- s and d denote the *source* and the *destination* core of a communication flow, respectively.
- k and l denote the *tiles* onto which core s and d lie, respectively.
- i and j denote the *switches* that are connected to core s and d, respectively.
- x and y denote two endpoints of an link on the XY route between switch i and j.

4.1. Parameters

The given parameters are listed in the following.

- Number of tiles/switches in the network = N
- Number of cores = $|V|$ ($|V| \leq N$)
- Traffic volume of a communication from s to d = v_{sd}
- Required bandwidth of the communication flow from core s to d = b_{sd}
- Capacity of a link in the network = C (Bytes/sec)
- Maximum number of switches connected to a core = D_C ($1 \leq D_C \leq 4$).
- Maximum number of cores connected to a switch = D_S ($1 \leq D_S \leq 4$).
- *Traffic routing*: The constant r_{xy}^{ij} = 1, if link $x \rightarrow y$ is on the XY route of the communication flow from switch i to j; r_{xy}^{ij} = 0, otherwise.
- Hop distance of the XY route from switch i to j = h_{ij}

4.2. Variables

We here define some variables for core-tile mapping, core-tile-switch mapping, communication flow, link bandwidth, and the maximum link bandwidth.

- *Core-tile mapping*: The variable m_{sk} = 1, if core s is placed on tile k; m_{sk} = 0, otherwise.
- *Core-tile-switch mapping*: The variable m_{sk}^{i} = 1, if core s is placed on tile k and connected to switch i; m_{sk}^{i} = 0, otherwise.
- *Communication flow*: The variable f_{skdl}^{ij} = 1, if core s is mapped onto tile k, connected to switch i, core d mapped onto tile l, connected to switch j, and traffic volume from core s to d is larger than 0; f_{skdl}^{ij} = 0, otherwise.

- *Link bandwidth*: The variable B_{xy} denotes the sum of required bandwidth of flows that traverse link $x \rightarrow y$.
- *Maximum link bandwidth*: The variable $B_{\max}$ denotes the bandwidth of the maximally congested link of the on-chip network.

4.3. Objectives

The two objective functions of our MILP formulation are

1. *Minimize the average hop distance*:

$$\text{Minimize: } \frac{1}{\sum_{\forall s,d} v_{sd}} \sum_{\forall s,d} v_{sd} \sum_{\forall k,l} \sum_{\forall i \in S_k, \forall j \in S_l} h_{ij} f_{skdl}^{ij} \quad (2)$$

Remark: The objective function minimizes the average hop distance in the on-chip network. The motivation for choosing this objective is that both of the energy consumption and the communication delay are proportional to the average hop distance.

2. *Minimize the maximum link bandwidth*:

$$\text{Minimize: } B_{\max} \quad (3)$$

Remark: The objective function minimizes the bandwidth of the maximally congested link in the on-chip network. Since traffic load on link $x \rightarrow y$ can be interpreted as the ratio of B_{xy} to C, this objective minimizes the queueing delay of the maximally congested link. In addition, as the required bandwidth scales up, minimizing the congestion will support the largest bandwidth growth, and therefore maximizes the throughput of the communication architecture.

Note that minimizing the congestion may indirectly reduce the average hop distance experienced by communication flows, and vice versa. Thus, these two objective functions are strongly correlated.

4.4. Constraints

1. *Core-tile mapping constraints*:

$$\sum_{\forall k} m_{sk} = 1, \qquad \sum_{\forall s} m_{sk} \leq 1 \quad (4)$$

Remark: The above constraints ensure *one-to-one* mapping between a core and a tile. Since the number of cores is less than or equal to that of tiles($|V| \leq N$), each core must lie on exactly one tile by the first equation. The second equation represents that each tile has at most one core. By this, we are assured that no two or more cores will be mapped onto the same tile.

2. *Core-tile-switch mapping constraints*:

$$m_{sk} \leq \sum_{\forall i \in S_k} m_{sk}^i \leq D_C \, m_{sk}, \;\; 0 \leq \sum_{\forall k \in T_i} \sum_{\forall s} m_{sk}^i \leq D_S \quad (5)$$

Remark: The above constraints ensure that, depending on D_C and D_S, the mapping between cores and switches can be one-to-one, one-to-many, many-to-one, or many-to-many. The first equation restricts the number of switches connected to a core; If core s lies on tile k, it has at most D_C internal links connected to its adjacent switches. On the other hand, the second equation limits the number of cores connected to a switch; Switch i has at most D_S internal links connected to the cores lying on its adjacent tiles.

3. *Communication-flow constraints*:

$$f_{skdl}^{ij} \leq m_{sk}^i, \qquad f_{skdl}^{ij} \leq m_{dl}^j \quad (6)$$

$$m_{sk} + m_{dl} - 1 \leq \sum_{\forall i \in S_k, j \in S_l} f_{skdl}^{ij} \leq \frac{m_{sk} + m_{dl}}{2} \quad (7)$$

Remark: Eq. (6) constrains the problem such that flow f_{skdl}^{ij} can exist only if core s is connected to switch i and core d to switch j. We here consider static XY routing in which only one of the shortest path is used for a flow. In other words, given that core s is placed on tile k and core d on tile l, only a single pair is used for communication flow $s \rightarrow d$ among all-possible pairs of their adjacent switches, i.e.,

$$\sum_{\forall i \in S_k, j \in S_l} f_{skdl}^{ij} = \begin{cases} 1, & \text{if } m_{sk} = 1 \text{ and } m_{dl} = 1; \\ 0, & \text{otherwise.} \end{cases} \quad (8)$$

Since Eq. (8) is a nonlinear equation of variables m_{sk} and m_{dl}, we need an alternative *linear* equation for this constraint. To this end, we devise Eq. (7), where the summation in the middle is equal to 1, if m_{sk} = 1 and m_{dl} = 1, or 0, otherwise.[1]

4. *Link-bandwidth constraint*:

$$\sum_{\forall s,d} \sum_{\forall k,l} \sum_{\forall i,j} b_{sd} r_{xy}^{ij} f_{skdl}^{ij} = B_{xy} \quad (9)$$

$$B_{xy} \leq B_{\max}, \qquad B_{\max} \leq C \quad (10)$$

Remark: The above equations ensure that the total bandwidth of any link is less than or equal to $B_{\max}$, and that the bandwidth of the maximally congested link cannot exceed the link capacity C.

5. Decomposition Approach

In the previous section, we formulated the full CSM problem as an MILP problem, however, it may not be computationally feasible to solve the MILP problem, particularly for large-scale CSM problems. Therefore, it is necessary to find more efficient ways to solve the CSM problem. This section presents a decomposition method that divides the CSM problem into two subproblems: the core-tile mapping(CTM) subproblem and the tile-switch mapping(TSM) subproblem. Since the size of each subproblem is smaller than the full CSM problem, we can improve the computational efficiency. But, in general, the decomposition method would not yield the optimal solution for the CSM problem.

1 The sum of f_{skdl}^{ij} in Eq. (7) should be 0 for m_{sk} = 0 or m_{dl} = 0, because f_{skdl}^{ij} is a *non-negative* binary variable.

5.1. The CTM Subproblem

In the CTM subproblem, we only consider how to place the cores over the tiles and ignore the issue of how to connect the cores to adjacent switches to meet the bandwidth constraints. Then, the CTM subproblem can be formulated as follows:

Minimize

$$\sum_{\forall s,d} v_{sd} h_{sd}$$

Subject to

$$\sum_{\forall k} m_{sk} = 1, \qquad \sum_{\forall s} m_{sk} \leq 1$$
$$f_{sd}^{kl} \leq m_{sk}, \qquad f_{sd}^{kl} \leq m_{dl}$$
$$\sum_{\forall k,l} f_{sd}^{kl} = 1, \qquad \sum_{\forall k,l} d_{kl} \cdot f_{sd}^{kl} = h_{sd}$$

where $h_{s,d}$ is defined as the smallest hop distance between two cores after their placement to tiles are determined, d_{kl} is the smallest hop distance between tile k and l, and f_{sd}^{kl} = 1, if core s is placed on tile k, core d on l, and core s has traffic to send to d; f_{sd}^{kl} = 0, otherwise. The hop-distance objective is employed in the CTM subproblem, since it may reduce the maximum link bandwidth as noted before. We omit the description of the constraints, because they are similar to those in section 4.

5.2. The TSM Subproblem

Given the solution of the CTM problem, our interest is how to find the optimal tile-switch mapping solution such that minimizes the two objective functions:

Minimize

1. $\dfrac{1}{\sum_{\forall s,d} v_{sd}} \sum_{\forall s,d} \sum_{\forall k,l} v_{sd}^{kl} \sum_{\forall i \in S_k, \forall j \in S_l} h_{ij} f_{kl}^{ij}$
2. $B_{\max}$

Subject to

$$1 \leq \sum_{\forall i \in S_k} m_{ki} \leq D_C, \qquad 0 \leq \sum_{\forall k \in T_i} m_{ki} \leq D_S$$
$$f_{kl}^{ij} \leq m_{ki}, \qquad f_{kl}^{ij} \leq m_{lj}$$
$$\sum_{\forall i \in S_k, \forall j \in S_l} f_{kl}^{ij} = 1, \qquad \sum_{\forall k,l} \sum_{\forall i \in S_k, \forall j \in S_l} b_{sd} r_{xy}^{ij} f_{kl}^{ij} = B_{xy}$$
$$B_{xy} \leq B_{\max}, \qquad B_{\max} \leq C$$

where the constant $v_{sd}^{kl} = v_{sd}$, if core s is placed on tile k and core d on tile l; v_{sd}^{kl} = 0, otherwise. The variable m_{ki} = 1, if core s on tile k has a connection with switch i; m_{ki} = 0, otherwise. The variable f_{kl}^{ij} = 1, if core s transmit packets via switch i and it arrives at core d via switch j; f_{kl}^{ij} = 0, otherwise.

We finally note that our decomposition method can also be used to obtain a complementary solution to the existing one-to-one CSM algorithms[5],[7], which will be explained in the next section.

$E_{L_{bit}}$	$E_{S_{bit}}$				
	4×4	5×5	6×6	7×7	8×8
5.445 pJ	0.43 pJ	0.52 pJ	0.61 pJ	0.69 pJ	0.78 pJ

Table 1. Bit-energy values for a link and switches with different sizes

6. Experimental Results

This section presents the results of our MILP formulations obtained by the CPLEX optimization package[14]. The results are also compared with those of three mapping algorithms, i.e. the PMAP algorithm[7], the NMAP algorithm[7], and the PBB algorithm[5].

6.1. Model Parameters

We first address the model parameters that are used to evaluate the energy consumption based on the LP optimization results. Table 1 shows the bit-energy values for a link and switches of different size, assuming 0.18 μm technology. The parameters used to calculate $E_{L_{bit}}$ are 1) length of a link = 2 mm; 2) capacitance of a wire = 0.5 fF/μm; and 3) voltage swing = 3.3 V. The bit-energy values of 4×4 and 8×8 switches are taken from the results of the fully-connected switches in [12]. Those of the others are estimated by linearly interpolating these two values. Given the bit-energy values and the optimization results, the energy consumed in the NoC architecture is expressed as

$$E_{NoC} = \sum_{\forall s,d} v_{sd} \sum_{\forall k,l} \sum_{\forall i \in S_k, \forall j \in S_l} E_{bit}^{ij} f_{skdl}^{ij}. \qquad (11)$$

6.2. Experiments with Random Benchmarks

In our first experiment, we consider four random applications, each consisting of 9 cores which are mapped onto a 3×3 2-D mesh NoC architecture. Their CCGs are randomly generated, such that 1) A certain fraction CA of directed edges connect two different vertices among all possible edges; 2) The required bandwidth of an edge(flow) is uniformly distributed over the range [0, 100 MBytes/s]; and 3) The traffic volume of an edge is uniformly distributed over the range [0, 1 Gbits]. In all figures, "mCSM" refers to the results of 4:4 mCSM, "CSM" 1:1 CSM, and "CTSM" the 1:1 decomposition method.

Fig. 3 shows the communication costs for 4 random applications(CA = 0.2, C = 1 GB/s). From the figure, we make three major observations. First, 1:1 CSM perform well for all benchmarks. More specifically, our MILP formulation can achieve 34.8 % energy savings and 34.4 % bandwidth savings, both on average, when compared to the heuristic algorithms. Second, the decomposition method gives competitive mapping solutions from the viewpoint of the energy consumption, but its bandwidth requirements are 18.4 % higher than other heuristic algorithms on average. This is because the CTM MILP formulation optimizes

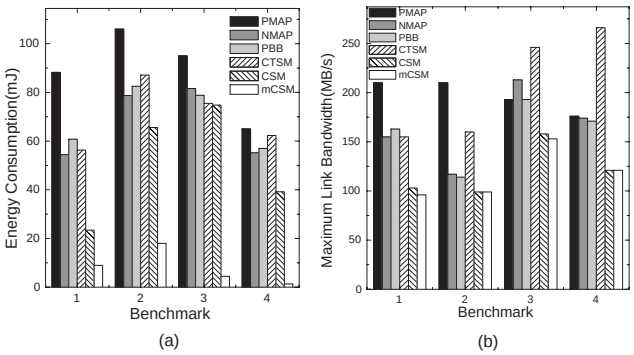

Figure 3. Comparison of six mapping algorithms with four random benchmarks

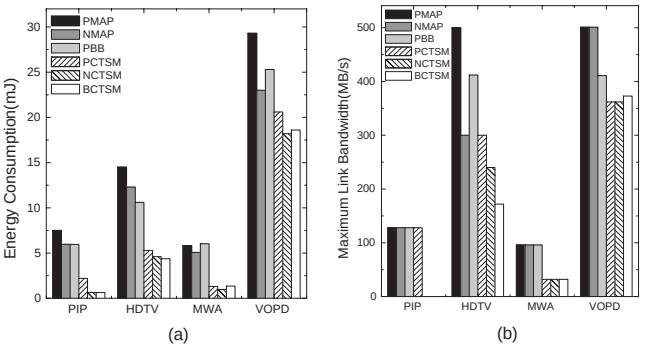

Figure 4. Comparison of heuristic mapping algorithms with four video applications

the average hop distance without considering the bandwidth constraints. Third, compared to 1:1 CSM, 4:4 mCSM saves approximately 81.2 % of energy consumption and 2.5 % of bandwidth requirement, both on average. This observation demonstrates the fact that providing multiple access points to the NoC architecture not only reduces the average hop distance, but also efficiently distributes the communication flows over themselves.

We also note that, depending on the benchmark, the CTM-TSM MILP runs $10 \sim 10^3$ times faster than the full MILP, but it can easily overwhelm the current computing resources when applied to the mapping problems with several tens of cores/tiles. We found that the running time of the decomposition method is dominated by that of the CTM subproblem, while the TSM subproblem is usually solved within 0.1 sec.

6.3. Experiments with Video Applications

To further improve the computational efficiency of the decomposition method, we utilize the existing mapping algorithms[5],[7] to obtain a CTM solution and then combine it with our TSM MILP formulation. In this section, we compare the quality of the solution generated by the combined decomposition method with the pure heuristic mapping algorithms. Fig. 4 shows the communication costs for the following video applications: *Picture-In-Picture*(PIP - 8 cores)[7], *HDTV video processor*(HDTV - 9 cores)[15], *Multi-Window Application*(MWA - 14 cores)[7] and *Video Object Plane Decoder*(VOPD - 16 cores)[7]. In the figure, "PCTSM" refers to the results of PMAP-CTM + 4:4 TSM heuristic, "NCTSM" NMAP-CTM + 4:4 TSM heuristic, and "BCTSM" PBB-CTM + 4:4 TSM heuristic. We observe that the combined decomposition method achieves 55.5 % and 49.2 % reduction in the energy consumption and the bandwidth requirement, respectively. These results demonstrate that, combined with the existing algorithms, our decomposition method can obtain more competitive solutions for medium- to large-scale CSM problems.

7. Conclusion

In this paper, we proposed a many-to-many core-switch mapping in a 2-D mesh NoC architectures. We presented exact mixed-integer linear programming(MILP) formulation, including the optimal choice of the core placements, switches for each core, and network interfaces for communication flows. We also presented a decomposition method that can obtain near optimal solutions with improved computational efficiency. From the results, we demonstrate that the mCSM is highly effective to reduce both the energy consumption and the congestion of an NoC architecture.

References

[1] S. Kumar *et al*., "A Network on Chip Architecture and Design Methodology," in *Proc. ISVLSI'02*, pp. 105-112, April 2002.

[2] K. Lahiri, A. Raghunathan, and S. Dey, "Efficient Exploration of the SoC Communication Architecture Design Space," in *Proc. IEEE/ACM ICCAD'00*, pp. 424-430, 2000.

[3] L. Benini and G. D. Micheli, "Powering Networks on Chips," in *Proc. Int'l Symp. System Synthesis*, pp. 33-38, Montreal, Canada, 2002.

[4] W. J. Dally and B. Towles, "Route Packets, Not Wires: On-Chip Interconnection Networks," in *Proc. DAC'01*, pp. 684-689, June 2001.

[5] J. Hu and R. Marculescu, "Energy-Aware Mapping for Tile-based NoC Architectures Under Performance Constraints," in *Proc. ASP-DAC'03*, pp. 233-239, Jan 2003.

[6] J. Hu and R. Marculescu, "Exploiting the Routing Flexibility for Energy/Performance Aware Mapping of Regular NoC Architectures," in *Proc. DATE'03*, pp. 688-693, 2003.

[7] S. Murali and G. D. Micheli, "Bandwidth-Constrained Mapping of Cores onto NoC Architectures," in *Proc. DATE'04*, pp. 896-901, Feb. 2004.

[8] T. Lei and S. Kumar, "A Two-step Genetic Algorithm for Mapping Task Graphs to Network on Chip Architecture," in *Proc. DSD'03*, pp. 180-187, Sept. 2003.

[9] P. Steenkiste, M. Hemy, T. Mummert, and B. Zill, "Architecture and Evalution of a High-Speed Networking Subsystem for Distributed-Memory Systems," in *Proc. 21st Ann. Int'l Symp. on Computer Architecture*, pp. 154-163, April 1994.

[10] P. Bhojwani and R. Mahapatra, "Interfacing Cores with On-chip Packet-switched Networks," in *Proc. VLSI'03*, pp. 382-387, Jan. 2003.

[11] C. J. Glass and L. M. Ni, "The Turn Model for Adaptive Routing," in *Proc. 19th Ann. Int'l Symp. Computer Architecture*, pp. 278-287, May 1992.

[12] T. T. Ye, L. Benini, and G. D. Micheli, "Analysis of Power Consumption on Switch Fabrics in Network Routers," in *Proc. DAC'02*, pp. 524-529, June 2002.

[13] D. Bertsekas and R. Gallager, *Data Networks*, Prentice-Hall, 1992.

[14] ILOG CPLEX Division. (2004) CPLEX optimization package. [Online]. Available: www.cplex.com

[15] H. Yamauchi *et al*., "Single Chip Video Processor for Digital HDTV," *IEEE Trans. on Consumer Electronics*, vol. 47, no. 3, pp. 394-404, Aug. 2001.