Exploration of Temperature Constraints for Thermal Aware Mapping of 3D Networks on Chip

Parisa Khadem Hamedani[1], Shaahin Hessabi[1], Hamid Sarbazi-Azad[1,2], Natalie Enright Jerger[3]
[1]Sharif University of Technology, Tehran, Iran, [2]Institute for Research in Fundamental Sciences (IPM), Tehran, Iran, [3]University of Toronto, Toronto, Canada

Abstract

This paper proposes three ILP-based static thermal-aware mapping algorithms for 3D Networks on Chip (NoC) to explore the thermal constraints and their effects on temperature and performance. Through complexity analysis, we show that the first algorithm, an optimal one, is not suitable for 3D NoC. Therefore, we develop two approximation algorithms and analyze their algorithmic complexities to show their proficiency. As the simulation results show, the mapping algorithms that employ direct thermal calculation to minimize the temperature reduce the peak temperature by up to 24% and 22%, for the benchmarks that have the highest communication rate and largest number of tasks, respectively. This comes at the price of a higher power-delay product. This exploration shows that considering power balancing early in the mapping algorithms does not affect the chip temperature. Moreover, it shows that considering the explicit performance constraint in the thermal mapping has no major effect on performance.

1. Introduction

With rapid advances in integrated circuit manufacturing technology, the number of processing cores in systems-on-chip is continuously increasing. As a result of technology scaling, interconnect becomes a bottleneck for performance, even though the performance of gates improves compared with older technologies. Furthermore, the rapid increase in the capability and complexity of applications increases the density and complexity of on-chip interconnects. Two solutions to alleviate the problem of interconnects are NoCs [1] and 3D integrated circuits (3D ICs) [2][3]. By combining these two techniques, we take advantage of their different benefits [2]. In a 3D IC, various layers of components are stacked vertically on each other; communication among these layers is through vertical interconnects. Some of the advantages of 3D ICs are abatement of overall interconnection length, scaling of chip dimensions, decrease of interconnect power and the possibility of mixing different technologies (e.g. logic, analog, DRAM) on one chip. Although a 3D IC has many advantages, it also has some disadvantages [2][3]. The most important disadvantage is the thermal management problem that is more significant than when using 2D ICs. Although the power consumption of a 3D IC is less than that of the equivalent 2D chip due to the decrease in average interconnection length, power density of a 3D IC is higher due to smaller chip area and the existence of the vertical layers. Also, the decrease in overall interconnection length can improve the operating frequency of the 3D chip, which can also contribute to the growth of the chip power density [2][3].

In addition, the increase in the number of the transistors and the higher operating frequency of the processors increase the chip power consumption, which results in a higher power density in the chip, which in turn increases the chip temperature. High temperature has several undesirable effects on nano-scale designs, such as performance degradation of transistors, increased static power consumption, increased RC delay in interconnects, and reliability issues such as Mean Time To Failure (MTTF) decline and thermo-migration [4][5]. As a result of these effects, designing and utilizing temperature control algorithms for SoCs is essential.

One method to control the chip temperature is mapping. The mapping algorithm decides how various tasks are assigned to cores based on the criteria that should be optimized [6]. These criteria can include minimizing the total network communication rate [7], minimizing the network communication energy [8], decreasing the network congestion [9] and network

This work is partially supported by a research grant from Iran Telecommunication Research Center (ITRC), the University of Toronto and the Natural Science and Engineering Research Council.

thermal balancing [10][11][12]. Genetic algorithms have been proposed for thermal aware mapping to reduce the peak temperature in 2D NoCs [10][11]; thermal- and communication-aware mapping for 3D NoCs can be performed using genetic algorithms [12]. This heuristic search algorithm speeds up the process of finding reasonably good solutions; however, we favor an approximation algorithm, for which we can analytically verify the performance, and unlike heuristic algorithms, can guarantee that the solutions are close to optimum [13].

This paper proposes three mapping algorithms and employs integer linear programming (ILP) to answer the following questions: Is it important to the chip temperature that the mapping algorithm consider thermal constraints, such as power or temperature? Which thermal constraints are sufficient to achieve an acceptable chip temperature: implicit constraints such as power estimation, or explicit constraints such as temperature? And how much do these constraints affect the chip temperature? Do these constraints have any negative effects on performance and power consumption? Is an implicit constraint such as router power consumption, which is used to estimate the temperature of a tile, sufficient to obtain the optimal power and performance?

The rest of the paper is organized as follows. Section 2 presents three thermal aware mapping algorithms with the mapping problem formulation, and gives the optimal and approximation solutions using an ILP approach. In Section 3, evaluation and experimental results are shown. Section 4 summarizes the main contributions.

2. Static thermal aware mapping

In order to develop our algorithms, we make some assumptions about the systems that are to be managed by those algorithms. The system is composed of similar processing elements, which is a common assumption in chip multiprocessors. As a result, none of them is specialized for a particular task. Furthermore, we assume that the applications that run on the chip can be broken down into tasks where the communication requirements are known at design time.

The mapping algorithm takes information related to the tasks, their communication and the topology structure in order to find a processing element (PE) for each task to run on, considering special metrics such as performance, power consumption or temperature. In the remainder of this section, we introduce three thermal-aware mapping algorithms. Our mapping algorithms take the communication task graph (CTG) as the input to find the mapping of tasks to PEs such that the chip temperature is optimized. The CTG = G (T, E) is a directed acyclic graph where each $T_i \in T$ and each $e_{i,j} \in E$ indicate the task (T) and communication dependence (E) between T_i and T_j, respectively [6].

2.1. Algorithm1: optimal thermal aware mapping

In this algorithm, we exploit ILP to map the tasks on a NoC such that the maximum temperature of the chip is minimized, considering the bandwidth and performance constraints. Our optimal temperature-aware mapping proceeds as follows:

- The parameters are technology information, such as the thickness of wires, interlayer dielectric (ILD), glue layer, silicon layer, topology dimensions and the routing algorithm. Based on this information, the first phase extracts the 3D NoC information, including the neighbors of each PE, all paths between each two PEs in the routing algorithm, and thermal resistances for all PEs.
- The second phase is thermal-aware mapping that maps all tasks to the PEs. For this phase, the NoC's physical information is required, such as the channel width between routers, bit-power consumption of the router, and the operating frequency of the NoC. The exact bit-power consumption of each router is not known until the end of the third phase, therefore, we use an estimate from the Orion2 library [14].
- Next, simulation estimates the power consumption of each router, the total power consumption, and the total latency. The simulation uses the output of the second phase, as well the information used in the first phase.
- Finally, the temperature of each tile (PE and its associated router) is calculated, as discussed in Section 2.1.2.

2.1.1. Parameters and variables. The parameters and variables that are used in the LP formulae (Figure 1) are explained below.

- The variables are shown as the binary matrix of M with $|PE|\times|T|$ elements, where $|PE|$ and $|T|$ are the number of PEs and the number of tasks in the CTG, respectively. Each element of the M matrix, $M_{T_m}^{Pe_i}$, will be 1 or 0 depending on whether the task T_m is mapped on the Pe_i or not.
- The parameter w_{T_i,T_j} is the communication rate between the tasks T_i and T_j .
- The parameter BW_{L_k} is the maximum available bandwidth of link L_k on NoC.

- The parameter $Virt_{Pe_i}$ is the maximum number of tasks that can run simultaneously on the Pe_i.
- The parameter CC_{th} is the average distance in the NoC topology, where the average distance of a 3D mesh is calculated as follows:

$$CC_{th} = \frac{1}{3} \sum_{\forall (T_i,T_j) \in E} w_{T_i,T_j} \left(d_1 - \frac{1}{d_1}\right)\left(d_2 - \frac{1}{d_2}\right)\left(d_3 - \frac{1}{d_3}\right)$$

  where $d_1, d_2,$ and d_3 are dimensions of 3D Mesh
- The set of links between Pe_i and Pe_j, based on the NoC routing algorithm, is $Link(Pe_i, Pe_j)$.

2.1.2. Objective function. Our objective is to minimize the maximum temperature that is caused by running the tasks on PEs. Since thermal capacitances are not required to calculate the steady state temperature of each PE, we use an existing thermal model [15] as follows:

$$Th_{i,j,k} = T_{Amb} + \sum_{m=1}^{k} \frac{R_{i,j,m}}{A} \times \left(\sum_{s=m}^{n} \left(P_{i,j,s} + PR_{i,j,s}\right) \right)$$

where $P_{i,j,s} = \sum_{\forall T_k \in E} p_{T_k} \times M_{T_i}^{Pe_{i,j,s}}$

where $PR_{i,j,s} = \sum_{\forall L_k \text{ that connected to the } R_{i,j,s}} \lambda_{L_k} \times PRpB$

Parameters:

- $Th_{i,j,k}$ shows the temperature of the PE at the position (*i, j, k*) in 3D NoC.
- The ambient temperature is shown with T_{Amb}.
- $R_{i,j,m}$ is the thermal resistance of the PE at the position (*i, j, m*) in 3D NoC.
- The area of each PE is shown with *A* in the formulae for temperature calculation.
- $P_{i,j,s}$ is the average power consumption of the PE at the position (*i, j, s*) in 3D NoC, which is calculated based on the average power consumption of the task(s) that runs on it. We assign a random power to each task based on the average power consumption of the core (at the range of 10 to 60 W/cm^2) at the specific technology [16].
- $PR_{i,j,s}$ is the average power consumption of the router at the position (*i, j, k*) in the 3D NoC. The router power is calculated based on the router power consumption per bit and the amount of the data which is routed from this router. We consider the average power consumption of cores and routers, since we intend to calculate the steady state temperature which is sufficient for static mapping.

2.1.3. Constraints. The constraints are shown in Figure 1.

- The first constraint is to control the task assignment such that each task is mapped to only one PE.

Minimize H

where $H = \max\{Temp_p : p = 1 \cdots m, \text{for a system of m cores}\}$

Subject to :

1) $\forall T_m \epsilon T: \sum_{\forall Pe_i \epsilon PE} M_{T_m}^{Pe_i} = 1$

$M_{T_m}^{Pe_i} = \begin{cases} 1 & \text{if map}(T_m) = Pe_i \\ 0 & \text{else} \end{cases}$

2) $\forall Pe_i \epsilon PE: \sum_{\forall T_m \epsilon T} M_{T_m}^{Pe_i} < Virt_{Pe_i}$

3) $\forall L_k \in \text{Link}: \lambda_{L_k} \leq BW_{L_k}$

$\lambda_{L_k} = \sum_{\forall (T_m,T_n)} \sum_{i=1}^{|PE|} \sum_{j=1}^{|PE|} w_{T_m,T_n} \times LE_{L_k}^{Pe_i,Pe_j} \times M_{T_m}^{Pe_i} \times M_{T_n}^{Pe_j}$

where $LE_{L_k}^{Pe_i,Pe_j} = \begin{cases} 1 & \text{if } L_k \epsilon \text{Link}(Pe_i, Pe_j) \\ 0 & \text{else} \end{cases}$

4) $\text{Comm. Cost} \leq CC_{th}$

$\text{Comm. Cost} = \sum_{\forall (T_m,T_n)} \sum_{i=1}^{|PE|} \sum_{j=1}^{|PE|} w_{T_m,T_n} \times \text{dist}(Pe_i, Pe_j) \times M_{T_m}^{Pe_i} \times M_{T_n}^{Pe_j}$

where $\text{dist}(Pe_i, Pe_j)$ = manhatan distance between Pe_i and Pe_j

Figure 1. LP formulae of optimal thermal aware mapping

- We use the second constraint to control the number of tasks that can be run on each PE.
- The third constraint is related to bandwidth. The total communication rate of the tasks assigned to each PE should be less than the total available bandwidth of the link.
- We use the fourth constraint as the performance constraint. To simplify the model, we use the zero load latency and ignore the delay of buffers in routers. However, we will consider the delay of buffers when evaluating our design in Section 3.

2.1.4. Linearization. In formulae related to the objective and the third constraint, variables appear as multiplication. Thus, these constraints cannot be solved using LP. In order to transform them into the LP model, we insert the following extra variables [9].

$$M_{T_u}^{Pe_i} + M_{T_v}^{Pe_j} - 1 \leq M_{T_u}^{Pe_i} \times M_{T_v}^{Pe_j} \leq \frac{M_{T_u}^{Pe_i} + M_{T_v}^{Pe_j}}{2}$$

$$0 \leq M_{T_u}^{Pe_i} \times M_{T_v}^{Pe_j} \leq 1$$

The first formula determines a lower and an upper bound for the quasi-convex function. The lower bound underestimates the non-convex function, which is derived from multiplication of two convex functions [17]. Also, we can impose the upper bound on them since they are binary variables.

2.1.5. Algorithm complexity. As branch and bound is used to solve the ILP model, the complexity of the mapping problem increases exponentially with the number of variables. The complexity of the simplex

Minimize Cut

$$\text{where } Cut = \sum_{\forall (T_i,T_j)\in E} w_{T_i,T_j} \times P_{T_i}^{Cl_0} \times P_{T_j}^{Cl_1} + w_{T_i,T_j} \times P_{T_i}^{Cl_1} \times P_{T_j}^{Cl_0}$$

Subject to:

$$1)\ \forall T_m \in T: P_{T_m}^{Cl_0} + P_{T_m}^{Cl_1} = 1$$

$$\text{where } P_{T_m}^{Cl_i} = \begin{cases} 1 & \text{if Partition}(T_m) = Cl_i \\ 0 & \text{else} \end{cases}$$

$$2)\ \sum_{\forall T_m \in T} P_{T_m}^{Cl_0} = K_0$$

$$3)\ \sum_{\forall T_m \in T} P_{T_m}^{Cl_1} = K_1$$

$$4)\ \sum_{\forall (T_i,T_j)\in E} w_{T_i,T_j} \times P_{T_i}^{Cl_0} \times P_{T_j}^{Cl_1} \leq W_{\text{cut}_{\text{LtoR}} \text{ or } \text{cut}_{\text{DtoU}}}$$

$$5)\ \sum_{\forall (T_i,T_j)\in E} w_{T_i,T_j} \times P_{T_j}^{Cl_0} \times P_{T_i}^{Cl_1} \leq W_{\text{cut}_{\text{RtoL}} \text{ or } \text{cut}_{\text{UtoD}}}$$

$$6)\ \sum_{\forall (T_i,T_j)\in E} w_{T_i,T_j} \times P_{T_i}^{Cl_0} \times P_{T_j}^{Cl_0} \leq W_{\text{flow}_{\text{Cl}_0}}$$

$$7)\ \sum_{\forall (T_i,T_j)\in E} w_{T_i,T_j} \times P_{T_i}^{Cl_1} \times P_{T_j}^{Cl_1} \leq W_{\text{flow}_{\text{Cl}_1}}$$

$$8)\ \sum_{\forall T_i \in T} \text{Pow}_{T_i} \times P_{T_i}^{Cl_0} \leq \text{Avg}_{\text{power}} + \chi$$

$$9)\ \sum_{\forall T_i \in T} \text{Pow}_{T_i} \times P_{T_i}^{Cl_1} \leq \text{Avg}_{\text{power}} + \chi$$

Figure 2. LP formulae of partitioning-based mapping

method used to solve the LP problem is polynomial. Thus, the number of variables has an explicit effect on the algorithm execution time. As mentioned above, the variable matrix M has $|PE|\times|T|$ elements. In addition, to linearize the problem we add extra variables. The total number of these variables is on the order of $|PE|^2 \times|E|$, because nonlinear variables appear by tensor product of M and M. As a result, it is not efficient to use this model for the next generation of chips; e.g., for 128 PEs and 100 edges in the CTG, the number of variables is O(1638400). Due to this complexity, we propose the following two mappings.

2.2. Algorithm2: partitioning based mapping

In this algorithm, we map the tasks onto PEs using hierarchical partitioning. At each step, we employ ILP to partition the tasks in order to find the minimum cut between two partitions, taking into account the bandwidth and power consumption balancing constraints. Simultaneously, we cluster the PEs with horizontal and vertical cuts (X and Y dimensions) based on the level of task partitioning in the current step. We proceed accordingly until the number of PEs in each partition is less than or equal to the number of layers in the third (or Z) dimension. Then, we sort and map the tasks in each partition based on their power consumptions; i.e., the task with the highest power consumption is mapped on the PE nearest the heat sink. The LP formulae are shown in Figure 2.

The number of variables in the first iteration of this model leads to a complexity of O($|E|$+$|T|$), where $|E|$ and $|T|$ are the number of edges and tasks in the communication task graph, respectively.

2.2.1. Parameters and Variables. The parameters and variables used in Figure 2 are explained below.

- Cl_i shows the ith partition.
- The binary matrix of P with $|Cl|\times|T|$ elements is the variable matrix of this model, where $|Cl|$ and $|T|$ are the number of partitions created at each step and the number of the tasks in the CTG, respectively. Each element of the P matrix, $P_{T_m}^{Cl_i}$, will be 1 or 0 depending on whether the task T_m is assigned to the partition Cl_i or not.
- The parameter w_{T_i,T_j} is the communication rate between the tasks T_i and T_j.
- The parameters K_0 and K_1 are the number of the tasks that can be assigned to the partition.
- The parameters $W_{cut_{LtoR} or\ cut_{DtoU}}$ and $W_{cut_{RtoL} or\ cut_{UtoD}}$ are the total available link flow between PE clusters from left (up) to right (down) and right (down) to left (up), respectively.
- The parameters $W_{Flow_{cl0}}$and $W_{Flow_{cl1}}$ are the total available link flows inside of each PE cluster. Sometimes it is impossible to map cores such that all bandwidth requirements are satisfied. To handle such cases, we increase the total available intra and inter PE clusters link flow to let the mapping process proceed. This will model performance degradation that will occur in practice.
- The parameter Pow_{T_i} is the average power consumption of the task T_i if it is run on the PE.
- The parameters Avg_{power} and χ are the mean and the standard deviation of power consumption of all tasks that should be partitioned. This bound has been specified empirically, since our experiments showed that amounts smaller than one standard deviation make the LP model infeasible, while larger values make this upper bound so large that the related constraint cannot bound the LP model.

2.2.2. Objective function. The objective is minimizing the cut between two partitions (Figure 2).

2.2.3. Constraints.

- The first constraint is to control the task assignment such that each task belongs to only one partition.

- We use the second and third constraints to control the number of tasks that are assigned to each partition based on the number of PEs in each cluster.
- The fourth to seventh constraints are bandwidth constraints. The total communication rate of the tasks assigned to each partition should be less than the total available bandwidth inside of the cluster; also, the total communication rate of the tasks between two partitions should be less than the total available bandwidth between clusters.
- We use the eighth and ninth constraints to balance the power consumption of tasks between partitions.

2.2.4. Linearization. In the 4th to 7th constraints, variables appear as the multiplication. We use a method similar to that of the algorithm 1 and insert the following extra variables for linearization.

$$P_{T_u}^{Cl_i} + P_{T_v}^{Cl_j} - 1 \le P_{T_u}^{Cl_i} \times P_{T_v}^{Cl_j} \le \frac{P_{T_u}^{Cl_i} + P_{T_v}^{Cl_j}}{2}$$

$$0 \le P_{T_u}^{Cl_i} \times P_{T_v}^{Cl_j} \le 1$$

2.3. Algorithm3: approximation mapping

Algorithm2 reaches its final result quickly because it does not consider detailed information for thermal estimation and its main emphasis is on power balancing. To see how important this parameter is in achieving a balanced temperature, we propose another algorithm which takes more information into account, similar to Algorithm1, but can be solved much faster.

To gain better system performance, this algorithm has two phases: partitioning and merging. In the partitioning phase, we employ algorithm2 to attain several smaller sub-graphs from the CTG. In the merging phase, we use algorithm1 to map and merge the sub-graphs created in the first phase. Before running algorithm1, we search within the sub-graphs to find the partition whose sum of internal communication rate is the highest in all sub-graphs. After we find its mapping using algorithm1, we search on the rest of the sub-graphs to find one that has the highest communication rate with the first already mapped sub-graph. Then, we merge these two sub-graphs into a larger sub-graph and employ algorithm1 to map the remaining unmapped tasks. We proceed with the merging process until all tasks in the CTG are mapped. At the merging phase, by merging the sub-graphs, the number of undetermined variables is not increased, because before each merging in the second phase, all tasks in the sub-graphs are mapped.

Table 1. Technology Information (45nm) [21]

No. of metal layers	12	V_{dd}	1.1
TSV area(m^2)	25e-12	T_{Amb} (℃)	25
Width(m)		Thickness(m)	
Cu local	45e-9	Cu local	81e-9
Cu semi-global	45e-9	Cu semi-global	81e-9
Cu global	67.5e-9	Cu global	162e-9
Thermal conductivity(w/mk)		ILD local	72e-9
Si	150	ILD semi-global	72e-9
Cu	400	ILD global	148e-9
Glue	0.25	Glue layer	2e-6
ILD	0.2	Si layer	10e-6
		Si bulk	500e-6

3. Evaluation

The approximation algorithm takes the technology parameters, topology information, and system-level information as input, and formulates them as a LP model, and solves them using Integer Linear Programming. We have implemented the algorithm in C++, and use the LPSolve package [18] to solve the linear model. Also, we have written a branch and bound algorithm to solve the ILP, based on the LP model results. The LPSolve package uses Simplex Method to solve the linear model.

For simulating the NoC, we use Booksim [19]. Power consumption for interconnects and NoC routers with 2 GHZ frequency and 128 flit size is modeled by adding Orion2 [14] to Booksim. Orion2 is designed for estimating the NoC's power consumption at the architecture level [14]. This library estimates the power consumption of the NoC router's pipeline stages, taking into account the architecture- and circuit-level parameters of the NoC, such as fabrication technology, operating frequency, etc. Power modeling in Orion2 has been devised for 2D NoCs, therefore, estimation of

Table 2. Benchmarks Information

Name	|T|	|E|	NoC x,y,z	Name	|T|	|E|	NoC x,y,z	Name	|T|	|E|	NoC x,y,z
GSM Coder[22]	53	81	4,4,4	MMS[23]	29	23	3,4,3	VOPD[24]	12	15	2,3,2
GSM Decoder[22]	34	48	3,4,3	MPEG4[24]	12	23	2,3,2	MWD[24]	12	13	2,3,2

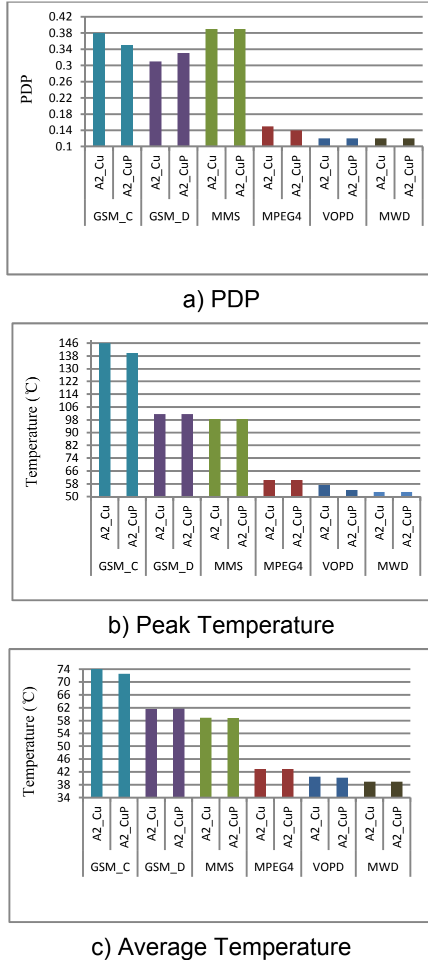


Figure 3. Simulation Results for A2_Cu, A2_CuP

interconnect power consumption is based on calculating the capacitances of the horizontal interconnects, and adding buffers to them. The data and the method used in Orion2 are not sufficient for modeling the 3D NoC power, due to the existence of vertical interconnects and their distinction in structure and size, compared to horizontal interconnects. Therefore, we have added the extra technology information, such as TSV size (Table 1), and formulae to calculate TSV capacitance [20].

To answer the questions asked in the introduction, we examine the different constraints and objectives of the second and third mapping algorithms. We do not evaluate Algorithm1 due to its algorithmic complexity, which makes it unsuitable for regular-sized systems.

3.1. Effect of task power balancing on temperature

To answer the question "How important is it for chip temperature to consider power balancing during the mapping algorithm based on partitioning?" we performed a series of simulations on two mapping algorithms based on algorithm2. The first mapping, *Algorithm2_CutPower (A2_CuP),* is similar to algorithm2 where both the cut minimization and task power balancing are taken into account to partition the CTG. In the other algorithm, *Algorithm2_Cut* (*A2_Cu*), we find the mapping based only on the cut minimization and omit the power balancing constraints.

We run these algorithms on the benchmarks listed in Table 2. The results are shown in Figure 3 as power delay product (PDP), peak and average temperatures.

Comparing the peak and the average temperatures, there are no major differences between the results. The maximum peak temperature difference is 6% for VOPD, where the A2_CuP result is better than that of A2_Cu.

As our results show, considering the task power balancing in the partitioning-based mapping has no major effect on the chip temperature. This is due to the fact that we do not have enough information about the final mapping in the early stages of the mapping algorithm. Additionally, the power balancing constraint can cause the LP model to become infeasible, if its upper bound is selected incorrectly. As a result, we do not recommend considering task power balancing for thermal management.

3.2. Effect of performance and explicit thermal constraints on temperature

It is important to know the effects of the explicit thermal constraint on the performance and temperature. Also, we need to know whether the implicit performance constraint in thermal mapping affects the performance in a way comparable to that of the explicit performance constraint. In order to investigate these issues, we perform simulations on three mapping algorithms based on algorithm3. In the first algorithm, referred to as *Algorithm3_Comunication (A3_C*), we change the algorithm3 objective of optimal mapping to minimize the zero load latency, as introduced in the fourth constraint in Section 2.1 (Figure 1). In this algorithm, we do not consider the temperature constraint. The second one of these three algorithms, referred to as *Algorithm3_ComunicationTemperature (A3_CT)*, is similar to the algorithm3, in that the objective of optimal mapping is to minimize the peak temperature and the performance, considering zero load latency. Finally, in the third algorithm, referred to as *Algorithm3_Temperature (A3_T*), the performance constraint is omitted; therefore, it finds the mapping solely based on the objective of minimizing the maximum temperature. For the partitioning phase of

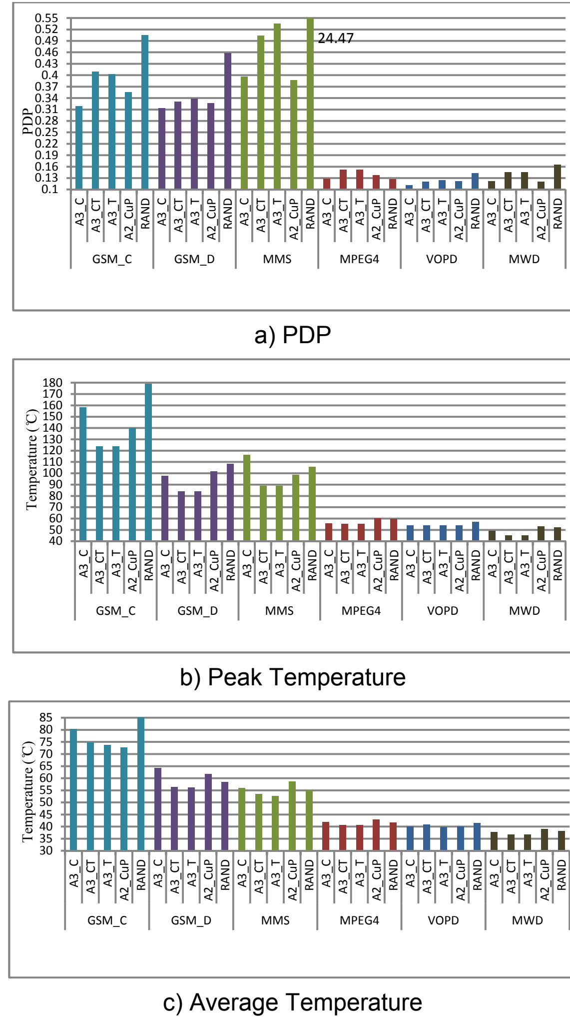
these algorithms, we run the partitioning based on cut and task power balancing.

The simulation results for maximum and average temperatures, and PDP are shown in Figure 4. Also, we compare the simulation results of these algorithms with those of A2_CuP introduced in Section 3.1.

As shown in Figure 4, the performance-based algorithms, i.e., A3_C and A2_CuP, have better PDP than those of the temperature-based algorithms; i.e., A3_T and A3_CT. PDP is increased up to 32% for the MMS benchmark, which has the highest communication rate. However, the A3_T and A3_CT algorithms have the best values for peak and average temperature. They achieve a 24% decrease in peak temperature for the MMS benchmark. For the GSM_C benchmark, which has the highest number of tasks, A3_T and A3_CT algorithms improve the peak temperature up to 22%; however, they increase PDP by 26%. Furthermore, the results of A3_CT do not show major differences compared to those of A3_T. Altogether, the thermal constraint can affect the peak and average temperatures.

Since we use branch and bound methods to solve the ILP model, A3_T and A3_CT are faster than A3_C in terms of algorithm run time. In A3_CT and A3_T algorithms, we can cut the branches earlier compared to A3_C, because in A3_C we have to wait for most of the tasks to be mapped in order to conclude that it yields a bad mapping; however in A3_CT and A3_T algorithms, branches are cut early based on the temperature cost. For example, A3_C creates 1622 branches to find the best mapping for MMS benchmark, while A3_T and A3_CT create 119 and 335 branches, respectively; LP solver must run once for each branch.

In order to calculate the tile temperature, we consider the power consumed by each router of NoC, based on the rate of the data it transfers. In order to minimize the peak temperature, the thermal-aware algorithm tries to decrease the data transferred by routers to gain lower temperature. This decrease in data transfer is an implicit performance constraint. As our simulations show, this implicit performance constraint can play the same role on performance management as explicit performance constraints, such as zero load latency. As a result, considering explicit performance constraints is not crucial in the thermal mapping algorithms, neither for achieving good performance, nor for a good temperature result. It is sufficient to solely consider the temperature constraint. It should also be mentioned that adding an explicit performance constraint in LP model can cause it to become infeasible, if its upper bound is selected incorrectly. We also compare results of mapping algorithms with those of a random mapping algorithm. As the results show, random mapping does not achieve suitable results, especially for MMS, GSM_D, and GSM_C. Using the thermal aware mapping becomes crucial when the number of the layers in 3D IC or the rate of task communication increases.



a) PDP

b) Peak Temperature

c) Average Temperature

Figure 4. Simulation Results for A3_C, A3_CT, A3_T

4. Conclusions

The increase in chip power density due to increasing the number of transistors and operating frequency can cause the chip temperature to rise. Employing 3D ICs, as a next generation technology, aggravates this problem. To avoid the negative effects of higher temperatures, such as transistor performance degradation, increased static power consumption, increased RC delay in interconnects, and reliability issues, we need to design thermal management algorithms.

In this paper, we proposed three thermal-aware mappings based on the LP model. We explored the effects of task power balancing early in the mapping, and the effects of performance and explicit thermal constraints on temperature and performance.

As our simulations show, considering the task power balancing in the partitioning-based mapping has no major effect on the chip temperature. Furthermore, in a mapping based on temperature minimization, since we consider the average router power to calculate temperature, considering the explicit performance constraint has no major effect on PDP. Therefore, considering the temperature constraint alone, which yields a simpler LP model, is sufficient. Moreover, our proposed algorithm is capable of reducing the peak chip temperature by up to 24% and 22% for the benchmarks with the highest communication rate and largest number of tasks, respectively.

5. References

[1] A. Jantsch, H. Tenhunen, *Network on Chip*, Kluwer Academic Publishers, Dordrecht, 2003.

[2] G. De Micheli, “An outlook on design technologies for future integrated systems,” *Transaction on Computer-Aided Design of Integrated Circuits and Systems*, vol. 28, no. 6, IEEE Press, NJ, USA, June 2009, pp. 777 – 790.

[3] V. F. Pavlidis, E. Friedman, *Three-dimensional integrated circuit design*, Morgan Kaufmann Publishers Inc., MA, USA, 2009.

[4] K.N. Tu, “Reliability challenges in 3D IC packaging technology,” *Microelectronics Reliability*, vol. 51, no. 3, Elsevier, March 2011, pp. 517-523.

[5] M. Pedram, S. Nazarian, "Thermal modeling, analysis, and management in VLSI circuits: principles and methods," *Proceedings of the IEEE*, vol.94, no.8, IEEE Press, NJ, USA, Aug. 2006, pp.1487-1501.

[6] R. Marculescu, et al., "Outstanding research problems in NoC design: system, microarchitecture, and circuit perspectives," *Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol.28, no.1, IEEE Press, NJ, USA, Jan. 2009, pp.3-21.

[7] S.Murali, G. De Micheli, “Bandwidth-constrained mapping of cores onto NoC architectures,” Proceedings Design, Automation and Test in Europe Conference and Exhibition, vol.2, Paris, France, Feb. 2004, pp. 896-901.

[8] J. Hu, R. Marculescu, “Energy- and performance-aware mapping for regular NoC architectures,” *Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol.24, no.4, IEEE Press, NJ, USA, April 2005, pp. 551- 562.

[9] C. Chou, R. Marculescu, “Contention-aware application mapping for Network-on-Chip communication architectures,” International Conference on Computer Design, CA, USA, Oct. 2008, pp.164-169.

[10] W. Hung, et al., “Thermal-aware IP virtualization and placement for Networks-on-Chip architecture,” International Conference on Computer Design: VLSI in Computers and Processors, CA, USA, Oct. 2004, pp. 430- 437.

[11] W. Zhou, Y. Zhang, Z. Mao, "Pareto based multi-objective mapping IP cores onto NoC architectures," Asia Pacific Conference on Circuits and Systems, Singapore, Dec. 2006, pp.331-334.

[12] C. Addo-Quaye, "Thermal-aware mapping and placement for 3-D NoC designs," International SOC Conference, VA, USA, Sept. 2005, pp.25-28.

[13] CA. Floudas, PM. Pardalos, *Encyclopedia of Optimization*, Springer, US, 2009.

[14] A. Kahng, B. Li, L.-S. Peh, K. Samadi, “ORION 2.0: a fast and accurate NoC power and area model for early-stage design space exploration,” Proceedings of the Conference on Design, Automation and Test in Europe, Nice, France, 2009, pp. 423 – 428.

[15] S. Im, K. Banerjee, “Full chip thermal analysis of planar (2-D) and vertically integrated (3-D) high performance ICs,” International Electron Devices Meeting, San Francisco, CA, USA, Dec. 2000, pp.727-730.

[16] C. Tsai, S. Kang, “Cell-level placement for improving substrate thermal distribution,” *Transaction on Computer-Aided Design of Integrated Circuits and Systems*, vol. 19, IEEE Press, NJ, USA, 2000, pp. 253–266.

[17] F.A. Al-Khayyal, J.E. Falk, “Jointly constrained biconvex programming,” *Mathematics of Operations Research*, vol. 8, no. 2, INFORMS, MD, USA, May 1983, pp. 273–286.

[18] lpsolve.sourceforge.net/5.5/

[19] W. Dally, B. Towles, *Principles and practices of interconnection networks*, Morgan Kaufmann Publishers Inc., CA, USA, 2003.

[20] R. Weerasekera, et al., “Closed-form equations for through-silicon via (TSV) parasitics in 3-D integrated circuits (ICs)," Proceedings of the Workshop on *3-D* Integration*,* DATE Conference, Nice, France, April 2009.

[21] Semiconductor Association, the International Technology Roadmap for Semiconductors (ITRS), 2009.

[22] M. Schmitz, “Energy minimization techniques for distributed embedded systems,” Ph.D. dissertation, School Electron. Comput. Sci., Univ. Southampton, Southampton, U.K., 2003.

[23] J. Hu, R. Marculescu, “System-level buffer allocation for application-specific Networks on Chip router design*,” Transaction on Computer-Aided Design of Integrated Circuits and Systems*, IEEE Press, NJ, USA, 2006, pp. 2919-2933.

[24] D. Bertozzi, et al., “NoC synthesis flow for customized domain specific multiprocessor Systems-on-Chip,” *Transaction on Parallel and Distributed. Systems*, IEEE Press, NJ, USA, 2005, pp. 113–129.