Thermal-aware task mapping for communication energy minimization on 3D NoC

Lili Shen\textsuperscript{1,2a)}, Ning Wu\textsuperscript{1)}, Gaizhen Yan\textsuperscript{1,3)}, Fen Ge\textsuperscript{1)}
\textsuperscript{1} College of Electronic and Information Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China
\textsuperscript{2} Jincheng college, Nanjing University of Aeronautics and Astronautics, 211156, Nanjing, China
\textsuperscript{3} School of Electrical and Electronic Engineering, Anhui Science and Technology University, Chuzhou, 233100, China
a) shirely Ivy@163.com

Abstract: Three dimensional Network-on-chip (3D NoC) is regarded as an attractive architecture delivering high communication performance. However, due to its high power density and strong vertical thermal correlation, thermal issues in 3D NoC are critical. In this paper, we propose a thermal-aware task mapping algorithm (3D-TTM) to reduce peak temperature meanwhile minimize the communication energy consumption. Experimental results show that the proposed scheme can achieve significant peak temperature reduction (up to 5.75K) when compared to other methods. Moreover, our proposed algorithm achieves up to 63.34% communication energy consumption reduction.

Keywords: 3D Network-on-chip, thermal, task mapping
Classification: Integrated circuits

References

1 Introduction

In comparison with traditional two dimensional implementations, three dimensional Network-on-chip (3D NoC) is an attractive solution to provide shorter interconnect lengths, higher packing density, and more flexible integration [1, 2]. However, it has the side effect of high power density and long heat dissipation paths in the vertical direction [3], which result in the increase of overall temperature. The temperature increase may exacerbate existing hotspots and create new hotspots within the chip that lead to shorter chip lifetime and worse system reliability. For example, every 10° rise of temperature will increase interconnect delay by 5% [4]. One efficient method to control the system temperature is task mapping. In NoC design, task mapping determines the placement of tasks to cores, that has a great influence on the system performance.

Various papers have presented thermal-aware task mapping and scheduling method for 3D NoC. Zhou et al. [5] proposed an OS-level scheduling algorithm for thermal balancing on 3D chip multiprocessor. Fuzzy logic-based task mapping methodology is presented in [6], which utilizes maximum heat emission capability of chip and reduces communication energy consumption by decreasing average transmission delay to achieve a significant reduction in peak temperature of cores. In [7] a 3D matrix synthesis problem based thermal-aware mapping approach is proposed to realize temperature equilibrium. Despite several previous works that have proposed thermal-aware task mapping algorithm, it is still difficult to achieve a high-efficiency task mapping. They either only consider the task power consumption, or they only consider communication energy consumption. In order to address the shortcoming, Our prior work [8] proposed a cluster-based thermal-aware task allocation algorithm to lower down the system peak temperature by clustering tasks. However, this approach increases the complexity when the tasks
increase. Moreover, it ignores the earlier tasks running on the core before a new task mapping.

Therefore, this paper further introduces genetic algorithm to find optimized allocation of thermal-aware task mapping approach for 3D NoC. The proposed algorithm takes task power consumption, communication energy consumption and heat emission capabilities of cores into consider to reduce peak temperature in a faster way. Besides, a detailed comparison has been accomplished with other thermal-aware task mapping algorithms.

2 Thermal model and Communication Energy Model of 3D NoC

2.1 Thermal model

Fourier heat flow analysis is the standard method to model heat conduction for circuit-level and architecture-level IC chip thermal analysis [9]. Fig.1 shows the thermal model of 3D NoC.

In this model, the temperatures of core 1 and core 2 can be calculated as follows:

\[ T_1 = T_{\text{amb}} + (P_F + P_D)R_{\text{amb}} \]  
\[ T_2 = T_1 + P_D R_V = T_{\text{amb}} + (P_F + P_D)R_{\text{amb}} + P_D R_V \]

Where \( P_D \) and \( P_F \) represent the power consumption of node D and F, respectively. \( R_V \) is the thermal resistance between different nodes in adjacent layers and \( R_H \) is the thermal resistance between different nodes in the same layer. \( R_{\text{amb}} \) represents the thermal resistance between node and the ambient environment. \( T_{\text{amb}} \) is the ambient temperature.

With the constant thermal resistance parameters, we know that the temperature of the core is mainly decided by the power consumption. Furthermore, due to \( R_H = 16 R_V \), the thermal correlation between vertical direction is much stronger than the horizontal layer that cores in the same stack should be considered together. Intuitively, we regard a set of vertically aligned cores...
Fig. 2. Example of Super-cores

as a super-core which has cores with similar temperatures. In this way, the dimension of the system can be reduced from 3D to 2D, that reduces the complexity of the task mapping algorithm, as shown in Fig.2.

2.2 Communication Energy Model

The total system energy can be divided into computation energy and communication energy. According to [10], the communication energy can take up to 20% to 36% of total energy. Hence, it is essential to consider communication energy.

For calculating the total communication energy, as in [11], the bit energy consumed on sending one bit from core $i$ to core $j$ is defined as

$$E_{bit}^{i,j} = \left( n_{\text{horizontal hops}} + 1 \right) E_{\text{router}}^{bit} + n_{\text{horizontal hops}} E_{\text{horizontal bit}}^{bit} + n_{\text{vertical}} E_{\text{vertical bit}}^{bit}$$ (3)$$

Where $n_{\text{horizontal hops}}$ represents horizontal distance between node $i$ and $j$, $n_{\text{vertical}}$ represents vertical distance. $E_{\text{horizontal bit}}^{bit}$ and $E_{\text{vertical bit}}^{bit}$ represent the energy consumed on each horizontal link and vertical link when transmitting one bit respectively, while $E_{\text{router}}^{bit}$ represents the energy consumed by the router when transmitting one bit.

The total communication energy consumed by a task graph with $m$ communication transactions can be defined as

$$E_{\text{comm}} = \sum_{k=1}^{m} w_{i,j} E_{bit}^{i,j}$$ (4)$$

Where $w_{i,j}$ indicates the amount of communication data volume between node $i$ and $j$.

As described in [11], the bit transmission energy on TSV is only 7.5% of that on the horizontal link. This shows that the tasks with more communication should be assigned into one super-core to reduce the communication energy.
3 The proposed thermal-aware task allocation algorithm

Our proposed thermal-aware task mapping algorithm for 3D NoC (3D-TTM) has two stages. The first stage is communication-aware group mapping, which allocates the communication intensive tasks into task groups while balancing the power consumption among each group, and maps the task groups to the super-cores considering the lowest communication energy. The second stage is thermal-aware task scheduling that moves some tasks to be executed on different layer to lower the temperature of system.

3.1 Communication-aware group mapping

Based on the analysis of the thermal model of 3D NoC, we view a core stack as a super-core, that we could apply the thermal-aware task mapping algorithm designed for 2D-NoC. According to the dependence of the tasks, the communication-aware group mapping has two steps as depicted in Fig3. In first step, we assume that the communication energy between the tasks in the same task group is negligible and the communication distance between tasks from different task group is the average Manhattan distance, which can be calculated as $NAD = (X+Y)/3*(1-1/(X*Y))$ , where X, Y are dimensions of 2D Mesh. Based on the simplification, we decide the task groups for each task so that the system communication is minimized. In second step, using the actual communication distance based on NoC architecture model, we try to further minimizing the communication energy by finding an optimal binding of each task group to a super-core. Genetic algorithm is employed to find optimized allocation for the above two steps.

![Fig. 3. Communication-aware group mapping](image)

3.1.1 First genetic algorithm

Following steps are used to decide the task groups for each task based on genetic algorithm.

1) Generate a random population of chromosomes, and each chromosome
represents a randomly design of tasks mapping on task groups. The length of every chromosome is equal to the number of tasks in the task graph, and the chromosome is encoded into integer strings. Every chromosome is a series of genes, and every gene contains an integer that indicates a randomly chosen task group.

2) Evaluate the fitness of every chromosome as the reciprocal of its communication energy(eq(4)). Here the average Manhattan distance is used as horizontal distance.

3) Generate a better population by using three operators (selection, crossover and mutation) similar to natural selection operators. The selection operator chooses two parent chromosomes from the population according to the better fitness. The crossover operator randomly chooses cross points from the parent chromosomes to create a new generation. For mutation, we exchange two genes in the new chromosome that randomly selected.

4) Repeat the above two steps to create a new population till a fixed number of loop iterations.

5) Select some best chromosomes from the last population as the inputs of the 2nd genetic algorithm.

3.1.2 Second genetic algorithm
In order to get the best system performance, after mapping tasks into task groups, another genetic algorithm will be used to find an optimal binding of each task group to a super-core. This genetic algorithm is similar to the first genetic algorithm except the following points:

1) Every chromosome represents a randomly design of task groups mapping on super-cores. The length of every chromosome is equal to the number of task groups, and every gene contains an integer that indicates a randomly chosen super-core.

2) Fitness function is also set as the reciprocal of communication energy, but actual communication distance is used.

3) Finally we only choose the best chromosome as the result.

In our implementation, we set the population size to be 20, maximum generation to be 100, crossover probability to be 0.9 and mutation probability to be 0.01.

3.2 Thermal-aware task scheduling
After allocating all task groups on the super-cores, the thermal-aware task scheduling is performed to adjust the tasks in super-core to lower the system temperature. For all cores in a super-core, the distances to the heat sink are different, cores farther from the heat sink are hotter [5]. Therefore, this stage
considers the power consumption and heat dissipation efficiency of each core to minimize peak temperature.

First, we sort the tasks of the task group in descending order according to the task power consumption and allocate the task which has highest power to each position of the super-cores. Second, we define a cost function to evaluate the power consumption of each core in the super-core.

The cost function for scheduling a task $v$ to a core is defined as

$$p = P_{\text{init}} + n_{\text{bottom}} \times P_t$$

In the equation, $P_{\text{init}}$ is the initial power consumption on core, which is determined by the previous tasks, $P_t$ is the power consumption of task $v$. In the same super-core, the heat dissipation ability of different positions can be denoted by the hop $n_{\text{bottom}}$ to reach the heat sink.

When assign the task to core, the scheduling algorithm always picks the core with the lowest cost. After the tasks in the task group are all scheduled, the algorithm schedules for other super-core and repeats the procedures until all the tasks in super-cores are allocated.

4 Experiments

To evaluate the efficiency of the proposed algorithm, we use the well-know E3S benchmarks [12] in experiments. Besides, a series of random tasks producing by TGFF [13] as benchmark as well. In this paper, comparisons have been made with the temperature-balancing algorithm (TBA) [5] in the communication energy consumption and the peak temperature. Besides, the proposed algorithm is compared with our prior work [8] as well, which is denoted as CTTA.

HotSpot 5.02 [14] is used as the thermal simulation tool which supports thermal simulation on 3D chip with a grid model. The configuration information of the HotSpot 5.02 is shown in Table I. The other configuration followed the default setting of HotSpot.

<table>
<thead>
<tr>
<th>Item</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bulk Si thickness of bottom die(next to heat sink)</td>
<td>150 µm</td>
</tr>
<tr>
<td>Bulk Si thickness of other die</td>
<td>50 µm</td>
</tr>
<tr>
<td>Si thermal conductivity</td>
<td>100.0 W/(m-K)</td>
</tr>
<tr>
<td>Heat sink thermal conductivity</td>
<td>400.0 W/(m-K)</td>
</tr>
<tr>
<td>HotSpot grid resolution</td>
<td>64x64</td>
</tr>
</tbody>
</table>

4.1 Random Benchmark and Results

In this experiment, the topology architecture of the simulation platform is $4 \times 4 \times 3$ 3D Mesh NoC. We use TGFF to generate six task graphs randomly (tgff1-tgff6). Each task graph contains 48 tasks, and the power consumption of each task obeys random distribution. Meanwhile, the amount of
communication between tasks are also generated randomly, and ensure the communication power consumption is less than 30% of the total power consumption of the system.

We first run the task mapping algorithm to schedule tasks and then the power traces of power consumption of all cores were generated. Finally, HotSpot simulates the temperature of the 3D NoC according to the power traces.

In Fig.4, a comparison of TBA, CTTA and 3D-TTM on communication energy consumption is given. Our proposed approach 3D-TTM saves 54.18% communication energy consumption on average than TBA, while the communication energy consumption of 3D-TTM and CTTA are nearly approximate. That is because 3D-TTM and CTTA consider communication energy consumption between tasks. However, 3D-TTM still saves 6.50% communication energy consumption on average than CTTA. Peak temperature comparison is shown in Fig.5. The result seems that our 3D-TTM algorithm achieves
significant peak temperature reduction. We set the temperature value of 3D-TTM as zero for clearly. The average peak temperature of 3D-TTM is lower than TBA by 3.51K, and the maximum reduction is 4.54K. While CTTA shows unstable capability to optimize due to that it ignores the earlier tasks running on the core before.

4.2 E3S Benchmark Suite and Results
In this experiment, the topology architecture of the simulation platform is $2 \times 2 \times 3$ 3D Mesh NoC, and the test case is E3S. The normalized communication energy consumption and peak temperature of TBA, CTTA and 3D-TTM with different test cases are shown in Fig.6 and Fig.7.

Then we can calculate the reduction percentage of the communication energy consumption and peak temperature compared with TBA and CTTA. It can be seen that 3D-TTM obtains 51.17% average reduction in communication energy consumption than TBA, and reduces the peak temperature up
to 5.75K compared to the other two algorithms.

5 Conclusion
In this paper, we propose a thermal-aware task mapping algorithm for reducing the peak temperature of the system meanwhile having minimal communication energy consumption. Experimental results show that, compared with other thermal-aware task mapping algorithms, the propose algorithm can achieve up to 63.34% reduction in communication energy consumption and up to 5.75K reduction in peak temperature respectively.

Acknowledgments
This work is supported by the National Natural Science Foundation of China (61376025,61774086), the Anhui Scientific Research Funds for University (KJ2017A501) and the Natural Science Foundation of Jiangsu Province (BK20160806).