Floorplanning Driven Network-on-Chip Synthesis for 3-D SoCs

Wei Zhong, Song Chen, Fei Ma, Takeshi Yoshimura, and Satoshi Goto
Graduate School of Information Production and Systems, Waseda University
Kitakyushu 808-0135, Japan,
{wzhong@ruri., chensong@aoni., mafei712@ruri., t-yoshimura@, goto}@waseda.jp

Abstract—As technology advances, 3-D stacking of silicon layers is emerging as a promising approach to address the integration challenges faced by current System-on-Chips (SoCs). Designing efficient Network-on-Chips (NoCs) is necessary to handle the 3-D interconnect complexity. In this paper, we present a four-stage synthesis approach to determine the power-performance efficient 3-D NoC topology for the application. First, we propose an algorithm to explore optimal clustering of cores during 3-D floorplanning. Then, an Integer Linear Programming (ILP) algorithm is proposed to place switches and network interfaces on the 3-D floorplan. Thirdly, a power and timing aware path allocation algorithm is carried out to determine the connectivity across different switches. Last, a min-cost max-flow based algorithm is proposed for Through-Silicon Via (TSV) assignment to minimize the link power consumption. Experimental results show the effectiveness of the proposed algorithm.

I. INTRODUCTION

The enhanced integration densities predicted by Moore's law may be achieved through new technologies such as 3-D ICs, in which multiple device layers are stacked above each other and vertically interconnected by through-silicon vias (TSVs) [1]. As compared to 2-D designs, 3-D ICs can significantly reduce interconnect delay and achieve high circuit performance.

In nanoscale technologies, the interconnects for 3-D ICs have evolved from simple vertical links connecting buses in different layers to a more scalable Network-on-Chip (NoC) solution [2]. NoCs are composed of switches and links and use circuit or packet switching technology to transfer data inside a chip. They provide better structure, modularity and scalability when compared to traditional interconnect solutions.

There are several significant challenges for the topology synthesis of 3-D NoCs. A 3-D NoC with few switches will lead to higher link power consumption. On the other hand, a large number of switches will lead to a large switch power consumption. Thus, in order to determine the best power-performance efficient 3-D NoC topology, proper switch numbers need to be determined. In 3-D ICs, the number of TSVs strongly depends on the underlying 3-D fabrication technology. For technologies where a large number of TSVs are available, cores across different 3-D layers can share the same switches, as more vertical links can be established. On the other hand, with a tight TSV constraint, a core on a layer may need to connect to a switch in the same layer. Thus, depending on the TSV constraint, the topology synthesized can be very different. Moreover, TSV macros need to be introduced in each layer for a vertical link and placement of switches, network interfaces and TSV macros need to be performed.

A lot of works have been done to synthesize application-specific NoC topology. In [3], a novel NoC topology generation algorithms were presented, however their solutions only consider topologies based on a slicing structure where switch locations are restricted to corners of cores. In [4] a partition-driven floorplanning algorithm is proposed. But the authors assume optimal switch number is given as an input, and apply min-cut partitioning every iteration in simulated annealing. In [5], synthesis approaches for designing power-performance efficient 3-D NoC topology are proposed. But the authors assume 3-D floorplan result as input and in order to obtain the optimal switch number, they explore the designs with several different partition numbers and cores only in the same layers can share the same switch. Moreover, as the switches located by the authors resulting in overlaps with cores, they have to reuse the floorplanner to remove the overlaps. In [6], a tool for 3-D NoC topology synthesis is proposed. However, the number of switches in the design is varied from 1 to the number of cores. The locations of switches and TSVs cause overlaps with cores and the overlaps are removed one by one. In [7], the authors present an efficient floorplan-aware 3D NoC synthesis algorithm, based on simulated allocation. But the positions of TSV macros and network interfaces are not taken into consideration.

In this paper, partitioning is integrated into the multi-layer floorplanning phase to explore the optimal switch number for clustering the cores. An Integer Linear Programming (ILP) method is also proposed to determine the optimal positions of the switches and network interfaces on the 3-D floorplan. Then, a power and timing aware path allocation algorithm (PA) [8] is carried out to determine the connectivity across the different switches. Finally, a min-cost max-flow based algorithm is proposed to assign TSV macros on the 3-D floorplan with minimized link power consumption.

The rest of this paper is organized as follows. Section II presents the approach used for topology synthesis. Section III discusses the details of the 4-stage 3-D NoC topology synthesis method. Experimental results and conclusions are presented in Sections IV and V, respectively.

II. DESIGN APPROACH

Fig.1 shows the design approach used to determine the power-performance efficient 3-D NoC topology. The input of the synthesis procedure is a Core Communication Graph (CCG), which could be represented by a directed graph $G = (V, E)$. Each vertex $v_i \in V$ represents a core and the edge $e_{ij}$ with the weight $w_{ij}$ represents the communication requirement between core $c_i$ and $c_j$. In the core specification file, the name and size of different cores are obtained as inputs. In addition, technology constraints such as the maximum number of TSVs across adjacent layers, NoC operating frequency and latency constraints are obtained. For the synthesis procedure, the area, power and timing models of the NoC and vertical links are also taken as inputs. The output of the synthesis procedure is an optimized application-specific 3-D NoC topology with pre-determined paths on network to route the traffic flows and the 3-D floorplan result of cores, switches, network interfaces and TSV macros with minimized power consumption.

1This research was supported by a grant of Knowledge Cluster Initiative 2nd stage implemented by Ministry of Education, Culture, Sports, Science and Technology(MEXT) and CREST (Core Research for Evolutional Science and Technology) JST, Japan.
III. 3-D NoC TOPOLOGY SYNTHESIS (FIPT)

As the topology synthesis problem is NP-Hard [9], we present efficient heuristics, a four-stage synthesis approach, to synthesize the best power-performance efficient topology for the application.

A. Floorplanning integrated with Cluster Generation (FCG)

This algorithm integrates the partitioning and TSV aware multi-layer floorplanning to explore the optimal clustering of cores with minimized power consumption. Fig.2 shows the flow of the proposed algorithm. The initial solution of 3-D floorplan is generated by a multi-layer fixed-outline floorplanning tool IARFP [10], which drives the floorplan with the objective evaluated by the linear combination of the area costs, wirelength and the number of TSVs.

For the required operating frequency and TSV constraints of the NoC, the maximum size of the switch max-sw_size and the maximum number of TSVs max-TSV_num are obtained as an input. Initial Partitioning applies a recursive min-cut bi-partitioning algorithm on Core Communication Graph (CCG) in Fig.1, according to the communication requirements and physical locations of the cores, until each cluster has the core number smaller than max_sw_size. In order to ensuring those cores with larger communication requirements and less distances are assigned to the same cluster and reduce the number of vertical links, during Initial Partitioning, we define new edge weight $w'_{ij}$ in CCG as:

$$w'_{ij} = \alpha_w \times \frac{w_{ij}}{\max_w} + (1 - \alpha_w) \times \frac{\min_{dis}}{\max_{dis}}$$  \hspace{1cm} (1)

where $w_{ij}$ denotes communication requirement between core $i$ and core $j$, $\max_{dis}$ denotes distance between core $i$ and $j$, $\max_w$ is the maximum communication requirement over all flows and $\min_{dis}$ is minimum distance among cores. We also reduce the weights of inter-layer flows, depending on the scaling parameter $\theta$.

Once the initial partition is generated, the next step, TSV Aware Multi-layer Floorplanning and Clustering, is invoked to explore optimal clustering of cores during multi-layer floorplanning. This step is integrated with a multi-layer floorplanning tool IARFP [10], which elaborates a perturbation method based on the technique of block position enumeration. To obtain designs with low power and inter-layer links, when a core $c_{mi}$ is inserted into the selected insertion point, the cost is evaluated as:

$$\Phi = \lambda_a A + \lambda_w W + \lambda_c C + \lambda_{tsv} TSV + \lambda_p P$$  \hspace{1cm} (2)

where $A$ denotes area cost, $W$ is wire-length, $C$ is the size of its candidate cluster, $TSV$ is the number of TSVs. $P$ is the total link power, which is evaluate as:

$$P = \sum_{i \leq n_t \leq n_c} \sum_{j \leq n_t} c_{r\_core} c_{i\_j} \cdot (|x_i - x_j| + |y_i - y_j| + |l_i - l_j| \cdot w_l)$$  \hspace{1cm} (3)

where $c_{r\_core}, c_{i\_j}, (x_i, y_i), (x_j, y_j)$ and $l_i$ are the coordinate and layer of core $c_i$ respectively. $n_t$ represents the number of cores and $w_l$ is the layer thickness. The parameters $\lambda_a$, $\lambda_w$, $\lambda_c$, $\lambda_{tsv}$ and $\lambda_p$ can be used to adjust the relative weighting between the contributing factors.

If the insertion point shows an improvement, the corresponding $(x, y)$ will be the new position of the core $c_{mi}$, and its the candidate cluster will include the core. Otherwise, an acceptable probability will be calculated.

After multi-layer floorplanning, the clusters with zero core will be ignored and the optimal switch number for clustering the cores is determined. The connectivity between cores and switches is also established, which can support the chip operating frequency and TSV constraint.

B. Switch and Network Interface Insertion

New switches and network interfaces will be included in the NoC topology so their physical positions must be determined to estimate the link power and delay. Due to the restriction that switch and network interface can only be placed within a whitespace, an even grid structure is constructed on each layer and the amount of whitespace in each grid $g_l$ is calculated, denoted as $ws(g_l)$. Let $A$ be the area of a switch or network interface. The capacity $cap(g_l)$ of a grid $g_l$ is defined as $cap(g_l) = \lfloor ws(g_l)/A \rfloor$. 

![Fig. 1. Proposed 3-D NoC Design Approach Overall](image1)

![Fig. 2. Floorplanning integrated with Cluster Generation (FCG)](image2)
We formulate it as an Integer Linear Programming (ILP) problem and simultaneously insert switches and network interfaces into the optimal grid positions with the minimized link power consumption.

Let \( a_{i,m} \) denotes whether grid \( g_i \) is chosen to insert network interface \( n_{im} \) and \( b_{j,k} \) denotes whether to choose grid \( g_j \) to insert switch \( su_k \). \( a_{i,m} = 1 \) if grid \( g_i \) is assigned to \( n_{im} \), otherwise \( a_{i,m} = 0 \). \( b_{j,k} = 1 \) if grid \( g_j \) is assigned to \( su_k \), otherwise \( b_{j,k} = 0 \).

If a network interface \( n_{im} \) is inserted into grid \( g_i \), the Manhattan distance between \( n_{im} \) and the corresponding core \( c_m \) is given by:

\[
d_{nim,cm} = |x_{g_i} - x_{cm}| + |y_{g_i} - y_{cm}| + \lambda_v \cdot \|g_i - lc_m\| \cdot w_l
\]

where \((x_{g_i}, y_{g_i})\) represent the coordinate of grid \( g_i \) and \((x_{cm}, y_{cm})\) is the coordinate of the core \( c_m \). \( l_g \) is the layer of grid \( g_i \), and \( l_{cm} \) is the layer of core \( c_m \). \( w_l \) is the layer thickness. The parameter \( \lambda_v \) is used to adjust the relative weight to ensure network interface is assigned to the same layer as its core.

Let \( n_g \) be the number of grids with non-zero capacity. The distance between network interface \( n_{im} \) and the corresponding core \( c_m \) is calculated as:

\[
d_{nim,cm} = \sum_{i=1}^{n_g} a_{i,m} \cdot d_{nim,cm}
\]

Let \( C_k \) be the set of cores in the \( k \)th cluster. For each network interface \( n_{ie} \) with its core \( c_e \in C_k \), the distance between \( n_{ie} \) and the switch \( su_k \) is denoted as \( d_{nis,e,k} \):

\[
d_{nis,e,k} = \sum_{i=1}^{n_g} a_{i,e} \cdot b_{j,k} \cdot d_{gij,j}
\]

where \( d_{gij,j} \) is the distance between grid \( g_i \) and grid \( g_j \), and can be calculated as:

\[
d_{gij} = |x_{g_i} - x_{g_j}| + |y_{g_i} - y_{g_j}| + \lambda_d \cdot \|g_i - g_j\| \cdot w_l
\]

the parameter \( \lambda_d \) is used to adjust the relative weight between the contributing factors.

The distance between switch \( su_d \) and switch \( su_t \) is denoted as \( d_{sw,d,t} \):

\[
d_{sw,d,t} = \sum_{j=1}^{n_g} b_{d,j} \cdot b_{t,j} \cdot d_{gij,j}
\]

However, the equation for \( d_{nis,e,k} \) and \( d_{sw,d,t} \) above are illegal in an ILP because they are non-linear. As a result, we introduce boolean variables \( \lambda_{ie,j,k} \) and \( \gamma_{id,j,t} \) to replace \( a_{i,e} \cdot b_{j,k} \) and \( b_{d,j} \cdot b_{t,j} \), respectively, and enforce the following artificial constraints:

\[
d_{nis,e,k} = \sum_{i=1}^{n_g} \sum_{j=1}^{n_g} \lambda_{ie,j,k} \cdot d_{gij,j} + a_{i,e} + b_{j,k} - \lambda_{ie,j,k} \leq 1
\]

\[
a_{i,e} - \lambda_{ie,j,k} \geq 0
\]

\[
b_{j,k} - \lambda_{ie,j,k} \geq 0
\]

Because of constraints(9), and the fact that \( d_{nis,e,k} \) appears in the cost function to be minimized, \( \lambda_{ie,j,k} \) will be equal to 0 unless both \( a_{i,e} \) and \( b_{j,k} \) are 1. Similarly, \( d_{sw,d,t} \) can be re-written.

Let \( cr_{core} \) be the communication requirement of the core \( c_m \), and \( cr_{sw2su,d,t} \) be the communication requirement between switch \( su_d \) and switch \( su_t \). To minimize the total power consumption of the links, we need to minimize the length of the links weighted by their communication requirement values, so that higher communication requirements are shorter than lower ones. Formulating the objective function mathematically, we get:

\[
\text{cost} = \sum_{m=1}^{n_c} \sum_{e=1}^{n_e} d_{nis,e,m} \cdot cr_{core,m} + \sum_{k=1}^{n_c} \sum_{e=1}^{n_e} d_{nis,e,k} \cdot cr_{core,e}
\]

\[
+ \sum_{d=1}^{n_d} \sum_{t=1}^{n_t} d_{sw,d,t} \cdot cr_{sw2su,d,t}
\]

where \( n_c \) and \( n_{sw} \) represent the number of cores and switches respectively. The ILP formulation for optimizing switch and network interface positions is as follows:

\[
\text{minimize cost subject to Equations}(4) - (10)
\]

\[
\sum_{i=1}^{n_g} a_{i,e} = 1, \quad \forall e \in \{1 \ldots n_e\}
\]

\[
\sum_{j=1}^{n_g} b_{j,k} = 1, \quad \forall k \in \{1 \ldots n_{sw}\}
\]

\[
\sum_{e=1}^{n_e} a_{i,e} + \sum_{k=1}^{n_{sw}} b_{i,k} \leq cap(g_i), \quad \forall i \in \{1 \ldots n_g\}
\]

\[
a_{i,e}, b_{j,k}, \lambda_{ie,j,k}, \gamma_{id,j,t} \in \{0, 1\}
\]

C. Power and Timing Aware Path Allocation

We take linear combination of power consumption and hop-count as objective to establish physical links and paths for traffic flows. The flows are assigned one by one, by applying Dijkstra’s shortest path Algorithm. In this procedure, the flows with switches across layers will be given higher priorities than the flows with switches in the same layer. Each kind of flows is ordered in decreasing rate requirements, and the bigger flow is assigned first. When opening a new physical link, we also check whether the switch size is small enough to satisfy the particular frequency of operation. In [8], the authors present methods to remove both routing and message dependent deadlocks when computing the paths. We also use the methods to obtain paths that are free of deadlock.

D. Through-Silicon Via Assignment

Assume a vertical link \( v_{lk} \) (vertical interconnects between two switches or network interface to switch) with two terminals on the layer \( m \), \( m < n \). In order to connect the vertical link, we have to introduce \((n - m)\) TSVs, \( tsv_{m+1}, \ldots, tsv_n \), with \( tsv_i \) on the device layer \( i \). Let \( V_Li \) represents the set of vertical links which need TSVs on layer \( i \). So, \( v_{lk} \in V_Li, m < i < n \).

For vertical interconnects, we carry out a min-cost max-flow algorithm layer by layer to locate TSV macros into whitespace of the 3D floorplan. As the area of TSV macros for a particular link width is taken as input, denoted as \( A_{sv} \), we calculate the capacity of each grid \( g_i \), defined as \( cap_{sv}(g_i) = [w_{sv}(g_i) / A_{sv}] \). Let \( GRIDS_i \) represents the set of grids with non-zero capacity on layer \( i \). Then network graph for layer \( i \), \( G_i = (V_i, E_i) \), is defined as:

- \( V_i = \{s, t\} \cup V_Li \cup GRIDS_i \)
- \( E_i = \{(s, v_{lk})\} \cup \{(v_{lk}, g_i)\} \forall g_i \in GRIDS_i \) \cup \{(g_j, t)\} \forall g_j \in GRIDS_i \)
- Capacities:
  \( C(s, v_{lk}) = 1, C(v_{lk}, g_j) = 1, C(g_j, t) = cap_{sv}(g_j) \).
Cost: $F(s, v_l k) = 0$, $F(v_l k, v_g) = P_{j,k}$, $F(g, t) = 0$.

where $P_{j,k}$ represents the link power consumption when locate TSV macros into grid $g_j$ on layer $i$ for vertical link $v_l k(\in V_L)$, $P_{j,k}$ can be calculated as the Manhattan distance between the grid $g_j$ and the corresponding two terminals of $v_l k$, weighted by their communication requirements value.

TSV assignment can be done efficiently by applying min-cost max-flow algorithms layer by layer, running in polynomial time [11].

IV. EXPERIMENTAL RESULT

The proposed methods have been implemented in C++ language and run on an IBM workstation (3.2 GHz and 3GB RAM) with Linux OS. We use hMetis [12] as our partitioning tool to generate the initial partition. Besides, we adopted Cbc [13] as our ILP solver. The power consumption is estimated using the power model presented in [7].

Two sets of benchmarks are used to evaluate the proposed algorithm. The first set of benchmarks are three video processing applications obtained from [14], including VOPD, MPEG4, and MWD. The next set of benchmarks are obtained from [3], including 263decmp3dec, 263encmp3dec and mp3encmp3dec. The data are averages of 10 runs.

We synthesize the topologies for the 3D cases (3 device layers). For reference, we also synthesize the topologies for the 2D cases, using our synthesis flow developed earlier [15]. TABLE I lists the experimental results. 3D (Fipt) represents the 3D NoC topologies synthesized by our proposed four-stage synthesis approach FiPT. 2D represents the synthesized 2D NoC topologies [15]. Most of the power savings obtained in 3D are due to the fact that the long horizontal wires in a 2D design are replaced by shorter vertical wires. As is shown, on average, 3D (Fipt) can reduce 31.71% of power consumption, compared to 2D NoCs.

In 3D cases, TSV macros need to be assigned on the 3D floorplan for vertical interconnects. However, under the current technology, TSVs are usually much larger than the vias in metal layers, and are usually placed at the whitespace between the macro blocks. So the assignment of TSVs also influences the link power consumption of 3D NoCs. We evaluate the impact of the TSV position constraints on the total power consumption. TABLE I shows the experimental results. 3D(Ideal) represents the power consumption of ideal assignment for all TSVs, which means that everywhere we want to insert a TSV, there is a whitespace. On the other hand, 3D (Fipt) carries out a min-cost max-flow algorithm to locate TSV macros on the whitespace of the 3D floorplan. As is shown, on average, a 48.64% power reduction can be obtained in the 3D case if TSVs are assigned to ideal positions. The gap of the power reduction between 3D(Ideal) and 3D (Fipt) (48.64%-31.71%=16.93%) is the link power consumption due to the detour routing restricted by TSV positions.

For further demonstrating the effectiveness, we compared the proposed method with another 2D NoC synthesis approach PDF [4], which applies a partition-driven floorplanning and inserts switches and network interfaces separately. TABLE II shows the experimental results. PDF applies min-cut partitioning every iteration in simulated annealing, and uses CBL [16] as the floorplan representation, which uses lots of dummy blocks to ensure good solutions on penalty of longer running time. Compared with PDF, 50.76% of power consumption, 4% of hop-count and 47.99% of running time can be saved by our FiPT 3D NoC synthesis approach.

V. CONCLUSIONS

In this paper, a four-stage synthesis approach FiPT is proposed to determine the power-performance efficient 3-D NoC topologies.