Accurate TSV Number Minimization in High-Level Synthesis

Chih-Hung Lee, Shih-Hsu Huang, and Chun-Hua Cheng
Department of Electronic Engineering
Chung Yuan Christian University
Chung Li, Taiwan, R.O.C.
Email: shhuang@cycu.edu.tw

ABSTRACT
Recent progress in process technology makes it possible to vertically stack multiple integrated chips. In three dimensional integration circuits (3D ICs), through silicon vias (TSVs) are used to communicate signals between layers. However, TSVs act as obstacles during placement and routing and have a negative impact on chip yield. Therefore, TSV number minimization is an important topic in 3D IC design. However, previous high-level synthesis approach only tries to maximize the number of same-layer operation-level data-transfers. In fact, a TSV should correspond to a cross-layer resource-level data-transfer. Therefore, in this paper, we propose an integer linear programming (ILP) approach to perform TSV number minimization by minimizing the number of cross-layer resource-level data-transfers. Experimental results consistently show that our approach is more effective than the previous approach in TSV number minimization.

Keywords:
Electronic Design Automation, High-Level Synthesis, Three Dimensional Integration Circuits, Layer Assignment, Through Silicon Via.

Correspondence:
Prof. Shih-Hsu Huang
Department of Electronic Engineering,
Chung Yuan Christian University,
Chung Li, TAIWAN, R.O.C.
TEL: 886-3-2654611
FAX: 886-3-2654699
Email: shhuang@cycu.edu.tw
1. INTRODUCTION

Three dimensional integrated circuits (3D ICs) technologies can highly integrate systems by vertically stacking and connecting various materials, technologies and functional units together. In 3D ICs, through silicon vias (TSVs) are used to communicate signals between layers. However, TSVs act as obstacles during the placement and routing stage [1,2]. Furthermore, TSVs also have a negative impact on chip yield. Therefore, TSV number minimization becomes an important topic in the design of 3D IC.

In high-level synthesis [3], a behavior-level description is represented by a data flow graph (DFG), in which each node corresponds to an operation, and each directed edge corresponds to a dependency relation (i.e., an operation-level data-transfer). Take the DFG shown in Figure 1(a), called ex, for illustration. This DFG has eleven operations, including o₁, o₂, o₃, o₄, o₅, o₆, o₇, o₈, o₉, o₁₀, and o₁₁. This DFG has nine dependency relations, including o₁→o₃, o₂→o₃, o₃→o₄, o₄→o₅, o₂→o₆, o₆→o₇, o₇→o₅, o₈→o₉, and o₁₀→o₁₁.

The high-level synthesis of 3D IC includes three main tasks: operation scheduling, resource binding, and layer assignment. Operation scheduling [4-6] is to assign each operation to a control step to start its execution (under dependency constraints and resource constraints). Resource binding [7,8] is to assign each operation to a functional unit that can execute it (under operation lifetime constraints and resource constraints). Layer assignment [9,10] is to assign each functional unit to a layer that can

---

1 Although TSVs result in many negative impacts, they are helpful to the reduction of heat dissipation. Therefore, some physical design researches [15,16] studied the addition of dummy TSVs for the reduction of heat dissipation. However, in the high-level synthesis stage, the design of 3D IC still focuses on the minimization of TSV number.

2 In this paper, the term “dependency relation” and the term “data-transfer” are used interchangeably.
accommodate it (under the constraint on the number of active device layers and the constraint on footprint area\(^3\)).

![Diagram](image_url)

**Figure 1:** The solution sol1 of example DFG ex. (a) Operation scheduling result of previous approach. (b) Resource binding and layer assignment results of previous approach.

Take DFG ex as an example. Figure 1(a) gives an operation scheduling result. Operations \(o_1\), \(o_2\), and \(o_{10}\) are assigned to control step 1, operations \(o_3\) and \(o_6\) are assigned to control step 2, operations \(o_4\), \(o_7\), and \(o_8\) are assigned to control step 3, operations \(o_5\), \(o_9\), and \(o_{11}\) are assigned to control step 4. Figure 1(b) gives a resource binding result. Operations \(o_9\) and \(o_{10}\) share adder \(A_1\), operations \(o_4\) and \(o_5\) share subtractor \(S_1\), operation \(o_1\), \(o_3\), and \(o_8\) share multiplier \(M_1\), operation \(o_2\), \(o_6\), and \(o_7\) share multiplier \(M_2\), and operation \(o_{11}\) is assigned to comparator \(C_1\). Figure 1(b) also gives a layer assignment result. Adder \(A_1\) and comparator \(C_1\) are assigned to layer 1, multiplier \(M_1\) is assigned to layer 2, and multiplier \(M_2\) and subtraction \(S_1\) are assigned to layer 3.

Mukherjee et al. [9] propose an integer linear programming (ILP) approach to perform the simultaneous application of operation scheduling, resource binding, and layer assignment for

\(^3\)The footprint area constraint means that the constraint on the area of each layer.
minimizing the number of TSVs. Because the input data lacks of the connection of resources, it is difficult to directly formulate the number of TSVs in high-level synthesis. Therefore, they [9] use an approximate method, in which a complementary objective function is used to maximize the number of same-layer operation-level data-transfers (i.e., to minimize cross-layer operation-level data-transfers). Their approximate method is reasonable but inaccurate, because a TSV should correspond to a cross-layer resource-level data-transfer instead of a cross-layer operation-level data-transfer. Actually, minimizing cross-layer operation-level data-transfers does not minimize the number of TSVs. As a result, their formulation [9] does not guarantee to minimize the TSV number.

In this paper, we also study the simultaneous application of operation scheduling, resource binding, and layer assignment. The main advantage of our approach is that we directly formulate the number of TSVs in our integer linear programming (ILP) formulation. Note that our formulation counts the TSV number according to cross-layer resource-level data-transfers. As a result, our approach guarantees minimizing the TSV number. Compared with the previous work [9], experimental data show that our approach can reduce 44.1% TSV number.

Besides, two following related works should be mentioned.

1. Mukherjee et al. [10] study physical-aware high-level synthesis of 3D ICs. But their objective is to reduce the total interconnect length. Therefore, their problem definition is different from ours.

1. Krishnan et al. [11] propose a high-level design flow for 3D ICs. But their design flow focuses on resource binding and floorplanning. Operation scheduling is not considered in their design flow. Therefore, their problem definition is different from ours.

The remainder of this paper is organized as follow. Section 2 demonstrates our motivation. In Section 3, we propose our ILP approach. In Section 4, we make a generation. Section 5 reports our experimental results. Finally, in Section 6, we make some concluding remarks.
2. MOTIVATION

The previous work [9] aims at the minimization of TSV number. However, their objective function is to maximize same-layer operation-level data-transfers. In the following, we demonstrate that their objective function does not minimize the number of TSVs.

Let’s use the DFG shown in Figure 1(a) for illustration. Suppose that we are given four control steps to execute these operations. Suppose that we are given one adder, called $A_1$, one subtractor, called $S_1$, two multipliers, called $M_1$ and $M_2$, and one comparator, called $C_1$; the areas of adder, subtractor, multiplier, and comparator are 600 $\mu$m$^2$, 600 $\mu$m$^2$, 1000 $\mu$m$^2$, and 600 $\mu$m$^2$, respectively; the power dissipations of adder, subtractor, multiplier, and comparator are 1 mW, 1mW, 2 mW, and 1 mW, respectively. Furthermore, suppose that the number of active device layers is 3 and the footprint area constraint (i.e., the constraint on the area of each layer) is 1600 $\mu$m$^2$. By applying the previous approach [9], we may have the solution sol1 as shown in Figure 1, in which operation scheduling result is displayed in Figure 1(a) and the resource binding and layer assignment results are displayed in Figure 1(b).\footnote{From the layer assignment result, we know that the areas of layer 1, layer 2, and layer 3 are 12, 10, and 16, respectively. Therefore, the footprint area constraint is satisfied.}

We analyze this solution sol1 below.

(1) The number of same-layer operation-level data-transfers (i.e., the objective function of the previous approach). From Figure 1(b), we find that the operation-level data-transfer $o_{10} \rightarrow o_{11}$ occurs at layer 1, the operation-level data-transfer $o_1 \rightarrow o_3$ occurs at layer 2, and the operation-level data-transfers $o_2 \rightarrow o_6$, $o_4 \rightarrow o_5$, $o_6 \rightarrow o_7$, and $o_7 \rightarrow o_5$ occur at layer 3. Therefore, the number of same-layer operation-level data-transfers is 6.
(2) The number of TSVs (i.e., the number of cross-layer resource-level data-transfers). From Figure 1(b), we find that there are one TSV from multiplier M\textsubscript{1} to adder A\textsubscript{1} is 1, one TSV from multiplier M\textsubscript{2} to multiplier M\textsubscript{1} is 1, and one TSV from multiplier M\textsubscript{1} to subtractor S\textsubscript{1} is 1. Therefore, the number of TSVs is 3.

Note that, in the solution sol1, the number of same-layer operation-level data-transfers is maximized. Therefore, the solution sol1 may be obtained by the previous approach [9]. We find that, in the solution sol1, the number of same-layer operation-level data-transfers is 6 and the number of TSVs is 3. However, in fact, there exists another solution in which the number of TSVs is only 2. That is to say, the previous approach [9] does not guarantee minimizing the number of TSVs. Let’s consider the solution sol2 as shown in Figure 2:

(1) Operation scheduling result. The scheduling result is displayed in Figure 2(a).

(2) Resource binding result. Operations o\textsubscript{9} and o\textsubscript{10} share adder A\textsubscript{1}, operations o\textsubscript{4} and o\textsubscript{5} share subtraction S\textsubscript{1}, operation o\textsubscript{1}, o\textsubscript{3}, and o\textsubscript{7} share multiplier M\textsubscript{1}, operation o\textsubscript{2}, o\textsubscript{6}, and o\textsubscript{8} share multiplier M\textsubscript{2}, and operation o\textsubscript{11} is assigned to comparator C\textsubscript{1}. The resource binding result is displayed in Figure 2(b).

(3) Layer assignment result. Comparator C\textsubscript{1} is assigned to layer 1, multiplier M\textsubscript{2} and adder A\textsubscript{1} are assigned to layer 2, and multiplier M\textsubscript{1} and subtraction S\textsubscript{1} are assigned to layer 3. The layer assignment result is displayed in Figure 2(b).

We analyze the solution sol2 below. From Figure 2(b), we find that there are one TSV from adder A\textsubscript{1} to comparator C\textsubscript{1} and one TSV from multiplier M\textsubscript{2} to multiplier M\textsubscript{1}. Therefore, in the solution 2, the number of TSVs is only 2. Compared with the solution of the previous approach sol1, the number of TSVs in the solution sol2 is less.
From the above discussions, we know that maximizing the number of same-layer operation-level data-transfers and minimizing the number of TSVs are not exactly equivalent. Therefore, the previous approach [9] does not guarantee minimizing the number of TSVs. The main reason is that the previous approach [9] lacks of formulating the resource-level data-transfers. As a result, their objective function is not really to minimize the TSV number. Actually, we cannot know the number of TSVs obtained by the previous approach [9], unless we perform an additional calculation process to count the TSV number (based on the result of the previous approach [9]). Thus, we are motivated to develop a new ILP formulation to accurately minimize the number of TSVs and directly report the number of TSVs.

**Figure 2:** The solution sol2 of example DFG ex. (a) Scheduled DFG of our approach. (b) Resource binding and layer assignment of our approach.

### 3. THE PROPOSED APPROACH

In this section, we present an ILP approach to perform the simultaneous application of operation scheduling, resource binding, and layer assignment for the architecture of 3D ICs. Given a DFG, a constraint on the number of layers, a constraint on the footprint area, a constraint on the number of control steps, and constraints on the resources, our objective is to minimize the number of TSVs.
Different from the previous work [9], we successfully formulate resource-level data-transfers. As a result, our ILP approach can accurately minimize the number of TSVs.

First, for the convenience of readers, Table I tabulates the notations used in our ILP formulation.

<table>
<thead>
<tr>
<th>Notation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$vl_{a_{k_1,k_2}}$</td>
<td>An integer variable that denotes the number of TSVs from resource $k_1$ to resource $k_2$. Note: $vl_{a_{k_1,k_2}} = 0$ when $k_1$ and $k_2$ are assigned to the same layer.</td>
</tr>
<tr>
<td>$D_{k_1,k_2}$</td>
<td>A binary variable. If the output of resource $k_1$ is the input of resource $k_2$, then $D_{k_1,k_2} = 1$; otherwise, $D_{k_1,k_2} = 0$.</td>
</tr>
<tr>
<td>$N_{layer}$</td>
<td>A constant that denotes the number of active device layers in the design.</td>
</tr>
<tr>
<td>$R_{max}$</td>
<td>A constant that denotes the total number of given resources.</td>
</tr>
<tr>
<td>$E_i$</td>
<td>A constant that denotes the earliest control step of operation $i$ (according to the ASAP schedule [6]).</td>
</tr>
<tr>
<td>$L_i$</td>
<td>A constant that denotes the latest control step of operation $i$ (according to the ALAP schedule [6]).</td>
</tr>
<tr>
<td>$R(i)$</td>
<td>The set of resources that can execute operation $i$.</td>
</tr>
<tr>
<td>$A_k$</td>
<td>A constant denotes the area of resource $k$ in the design.</td>
</tr>
<tr>
<td>$A_{max}$</td>
<td>A constant denotes the footprint area.</td>
</tr>
<tr>
<td>$p_k$</td>
<td>A constant denotes the average power dissipation of resource $k$.</td>
</tr>
<tr>
<td>O</td>
<td>The set includes all the operations in the DFG</td>
</tr>
<tr>
<td>$x_{i,j,k}$</td>
<td>A binary variable. If operation $i$ is scheduled in control step $j$ and bound to resource $k$, then $x_{i,j,k} = 1$; otherwise, $x_{i,j,k} = 0$.</td>
</tr>
<tr>
<td>$x_{i,j}$</td>
<td>A binary variable that denotes operation $i$ is scheduled in control step $j$. In other words, $x_{i,j} = \sum_{k=1}^{R_{max}} x_{i,j,k}$.</td>
</tr>
<tr>
<td>$x_{i,k}$</td>
<td>A binary variable that denotes operation $i$ is bound to resource $k$. In other words, $x_{i,k} = \sum_{j=E_i}^{L} x_{i,j,k}$.</td>
</tr>
<tr>
<td>$r_{k,l}$</td>
<td>A binary variable. If resource $k$ is assigned to layer $l$, then $r_{k,l} = 1$; otherwise, $r_{k,l} = 0$.</td>
</tr>
</tbody>
</table>
We use the binary variable $D_{k_1,k_2}$ to represent the data-transfer from resource $k_1$ to resource $k_2$. If the value of $D_{k_1,k_2}$ is 1 (i.e., the output of resource $k_1$ is the input of resource $k_2$), then the value of $\text{via}_{k_1,k_2}$ is \[ \sum_{l=1}^{N_{\text{layer}}} l \times r_{k_1,l} - \sum_{l=1}^{N_{\text{layer}}} l \times r_{k_2,l}. \] On the other hand, if the value of $D_{k_1,k_2}$ is 0 (i.e., no data-transfer from resource $k_1$ to resource $k_2$), then the value of $\text{via}_{k_1,k_2}$ is 0.

Now we use Figure 3 as an example. Suppose that resource $\alpha$ is assigned to layer 2, resource $\beta$ is assigned to layer 5, resource $\gamma$ and $\delta$ are assigned to layer 1, and resource $\epsilon$ is assigned to layer 3. Then, we have $r_{\alpha,l}=1$ and $r_{\alpha,l}=0$ for $l \neq 2$, $r_{\beta,l}=1$ and $r_{\beta,l}=0$ for $l \neq 5$, $r_{\gamma,l}=1$ and $r_{\gamma,l}=0$ for $l \neq 1$, $r_{\delta,l}=1$ and $r_{\delta,l}=0$ for $l \neq 1$, $r_{\epsilon,l}=1$ and $r_{\epsilon,l}=0$ for $l \neq 3$. Since there are data-transfers from resource $\alpha$ to resource $\beta$, from resource $\beta$ to resource $\alpha$, from resource $\alpha$ to resource $\delta$, from resource $\delta$ to resource $\gamma$, and from resource $\epsilon$ to resource $\beta$, we have $D_{\alpha,\beta}=1$, $D_{\beta,\alpha}=1$, $D_{\alpha,\delta}=1$, $D_{\delta,\gamma}=1$, and $D_{\epsilon,\beta}=1$. On the other hand, we have $D_{\alpha,\epsilon}=0$, $D_{\beta,\epsilon}=0$, $D_{\gamma,\delta}=0$, $D_{\delta,\gamma}=0$, $D_{\epsilon,\alpha}=0$, ... and so on.

Thus, we have \[ \text{via}_{\alpha,\beta} = |2 \times r_{\alpha,2} - 5 \times r_{\beta,5}| = 3, \quad \text{via}_{\beta,\alpha} = |5 \times r_{\beta,5} - 2 \times r_{\alpha,2}| = 3, \quad \text{via}_{\alpha,\delta} = |2 \times r_{\alpha,2} - 1 \times r_{\delta,1}| = 1, \]
\[ \text{via}_{\delta,\gamma} = |1 \times r_{\delta,1} - 1 \times r_{\gamma,1}| = 0, \quad \text{and} \quad \text{via}_{\epsilon,\beta} = |3 \times r_{\epsilon,3} - 5 \times r_{\beta,5}| = 2. \] Besides, for the pair of resources $k_1$ and $k_2$ in which $D_{k_1,k_2}=0$, we have $\text{via}_{k_1,k_2}=0$. Note that the number of TSVs can be obtained by summing the value of $\text{via}_{k_1,k_2}$ for each pair of resources $k_1$ and $k_2$. Take the example shown in Figure 3 for illustration. In this example, the number of TSVs is 9 (i.e., $3+3+1+2 = 9$). Since our goal is to minimize the TSV number, our objective function is to minimize \[ \sum_{k_1=1}^{R_{\text{max}}} \sum_{k_2=1}^{R_{\text{max}}} \text{via}_{k_1,k_2}. \]

---

\[ From \ this \ expression, \ if \ D_{k_1,k_2}=1 \ and \ k_1=k_2, \ the \ value \ of \ \text{via}_{k_1,k_2} \ is \ 0. \]
Next, we describe the constraints. Consider a dependency relation \( i_1 \rightarrow i_2 \) in the DFG. If operation \( i_1 \) is assigned to resource \( k_1 \) and operation \( i_2 \) is assigned to resource \( k_2 \), then the value of binary variable \( D_{k_1,k_2} \) must be 1. Thus, for each dependency relation \( i_1 \rightarrow i_2 \) and each pair of resources \( k_1 \rightarrow k_2 \) (which means the data-transfer from resource \( k_1 \) to resource \( k_2 \)), we have the following constraint:

\[
x_{i_1,k_1} + x_{i_2,k_2} \leq 1 + D_{k_1,k_2}.
\]

(Formula 1)
We use Figure 4 as an example. Figure 4(a) gives the viewpoint of operation-level data-transfer. There is a data-transfer from operation $i_1$ to $i_2$ in the DFG. Figure 4(b) gives the viewpoint of resource-level data-transfer. Adders $A_1$ and $A_2$ are two binding candidates for operation $i_1$, and multipliers $M_1$, $M_2$ and $M_3$ are three binding candidates for operation $i_2$. Then, from Formula 1, the following six constraints will be generated for handling the relation between operation-level data-transfers and resource-level data-transfers:

$$x_{i_1, A_1} + x_{i_1, M_1} \leq 1 + D_{A_1, M_1}, \quad x_{i_1, A_2} + x_{i_1, M_2} \leq 1 + D_{A_1, M_2}, \quad x_{i_1, A_3} + x_{i_1, M_3} \leq 1 + D_{A_1, M_3},$$

$$x_{i_2, A_1} + x_{i_2, M_1} \leq 1 + D_{A_2, M_1}, \quad x_{i_2, A_2} + x_{i_2, M_2} \leq 1 + D_{A_2, M_2}, \quad x_{i_2, A_3} + x_{i_2, M_3} \leq 1 + D_{A_2, M_3}.$$
0 ≤ \( \text{via}_{k_1,k_2} \). \hspace{1cm} \text{(Formula 2)}

This constraint is applied to give a lower bound on the number of TSVs from resource \( k_1 \) to resource \( k_2 \).

As mentioned in the beginning of this section, if the output of resource \( k_1 \) is the input of resource \( k_2 \) (i.e., \( D_{k_1,k_2} = 1 \)), then the value of \( \text{via}_{k_1,k_2} \) is \( \left\lfloor \sum_{l=1}^{N_{\text{layer}}} l \times r_{k_1,l} - \sum_{l=1}^{N_{\text{layer}}} l \times r_{k_2,l} \right\rfloor \). In order to use ILP to formulate it, for each pair of resources \( k_1 \rightarrow k_2 \), we have the following two constraints on the value of \( \text{via}_{k_1,k_2} \):

\[
\sum_{l=1}^{N_{\text{layer}}} l \times r_{k_1,l} - \sum_{l=1}^{N_{\text{layer}}} l \times r_{k_2,l} \leq \text{via}_{k_1,k_2} + (1 - D_{k_1,k_2}) \times N_{\text{layer}}. \hspace{1cm} \text{(Formula 3)}
\]

\[
\sum_{l=1}^{N_{\text{layer}}} l \times r_{k_2,l} - \sum_{l=1}^{N_{\text{layer}}} l \times r_{k_1,l} \leq \text{via}_{k_1,k_2} + (1 - D_{k_1,k_2}) \times N_{\text{layer}}. \hspace{1cm} \text{(Formula 4)}
\]

Now we use Figure 3 to illustrate Formula 3 and Formula 4. We discuss three possible cases below.

**Case 1 (the condition of no resource-level data-transfer):** Consider the number of TSVs from resource \( \alpha \) (at layer 2) to resource \( \epsilon \) (at layer 3). Since there is no data-transfer from resource \( \alpha \) to resource \( \epsilon \), the value of variable \( D_{\alpha,\epsilon} \) is 0. Due to Formula 3 and Formula 4, we have the constraint \( 2 \times r_{\alpha,2} - 3 \times r_{\epsilon,3} = 2 - 3 = -1 \leq \text{via}_{\alpha,\epsilon} + N_{\text{layer}} \) and the constraint \( 3 \times r_{\epsilon,3} - 2 \times r_{\alpha,2} = 3 - 2 = 1 \leq \text{via}_{\alpha,\epsilon} + N_{\text{layer}} \), in which \( N_{\text{layer}} \) is greater than 5. As a result, from these two constraints, we have \( -5 \leq \text{via}_{\alpha,\epsilon} \). Furthermore, due to Formula 2, we have \( 0 \leq \text{via}_{\alpha,\epsilon} \).

Since the objective function is to minimize \( \sum_{k_1=1}^{R_{\text{max}}} \sum_{k_2=1}^{R_{\text{max}}} \text{via}_{k_1,k_2} \), the value of \( \text{via}_{\alpha,\epsilon} \) will be 0. Thus, the number of TSVs will be 0, if there is no data-transfer between resources.
Case 2 (the condition of same-layer resource-level data-transfer): Consider the number of TSVs from resource $\delta$ (at layer 1) to resource $\gamma$ (at layer 1). Since there is a data-transfer from resource $\delta$ to resource $\gamma$, the value of variable $D_{\delta,\gamma}$ is 1. Due to Formula 3 and Formula 4, we have the constraint $1 \times r_{\delta,1} - 1 \times r_{\gamma,1} = 1 - 1 = 0 \leq \text{via}_{\delta,\gamma} + (1-1) \times N_{\text{layer}} = \text{via}_{\delta,\gamma}$ and the constraint $1 \times r_{\gamma,1} - 1 \times r_{\delta,1} = 1 - 1 = 0 \leq \text{via}_{\delta,\gamma} + (1-1) \times N_{\text{layer}} = \text{via}_{\delta,\gamma}$. As a result, from these two constraints, we have $0 \leq \text{via}_{\delta,\gamma}$. Furthermore, due to Formula 2, we have $0 \leq \text{via}_{\delta,\gamma}$. Since the objective function is to minimize $\sum \sum_{k_1=1}^{R_{\text{max}}} \sum_{k_2=1}^{R_{\text{max}}} \text{via}_{k_1,k_2}$, the value of $\text{via}_{\delta,\gamma}$ will be 0. Thus, the number of TSVs will be 0, if the resource-level data-transfer occurs at the same layer.

Case 3 (the condition of cross-layer resource-level data-transfer): Consider the number of TSVs from resource $\epsilon$ (at layer 3) to resource $\beta$ (at layer 5). Since there is a data-transfer resource $\epsilon$ to resource $\beta$, the value of variable $D_{\epsilon,\beta}$ is 1. Due to Formula 3 and Formula 4, we have the constraint $3 \times r_{\epsilon,3} - 5 \times r_{\beta,5} = 3 - 5 = -2 \leq \text{via}_{\epsilon,\beta} + (1-1) \times N_{\text{layer}} = \text{via}_{\epsilon,\beta}$ and the constraint $5 \times r_{\beta,5} - 3 \times r_{\epsilon,3} = 5 - 3 = 2 \leq \text{via}_{\epsilon,\beta} + (1-1) \times N_{\text{layer}} = \text{via}_{\epsilon,\beta}$. As a result, from these two constraints, we have $\text{via}_{\epsilon,\beta} \geq 2$. Since the objective function is to minimize $\sum \sum_{k_1=1}^{R_{\text{max}}} \sum_{k_2=1}^{R_{\text{max}}} \text{via}_{k_1,k_2}$, the value of $\text{via}_{\epsilon,\beta}$ will be 2. Thus, the number of TSVs will be correct and exact, if the resource-level data-transfer occurs at different layers.

Next, we introduce the layer assignment constraint. A resource must be assigned to one and only one layer. Thus, for each resource $k$, we have the following constraint:

$$\sum_{l=1}^{N_{\text{layer}}} r_{k,l} = 1.$$  
(Formula 5)
The summation of the areas of resources at the same layer must be less than or equal to the given footprint area. Thus, for each layer \( l \), we have the following constraint:

\[
\sum_{k=1}^{R_{\text{max}}} A_k \times r_{k,l} \leq A_{\text{max}} .
\]  \hspace{1cm} (Formula 6)

A decreasing power gradient must be maintained from the lowest layer (i.e., layer \( N_{\text{layer}} \)) to the highest layer (i.e., layer 1) in order to control the thermal gradient in the 3D IC. Thus, for each pair of adjacent layers, the average power of resources in the upper layer must be less than that of the lower layer. As a result, for each pair of adjacent layers \( l_1 \) and \( l_2 \), where \( l_1 < l_2 \), we have the following constraint:

\[
\sum_{k=1}^{R_{\text{max}}} p_k \times r_{k, l_1} < \sum_{k=1}^{R_{\text{max}}} p_k \times r_{k, l_2} .
\]  \hspace{1cm} (Formula 7)

Now we introduce the lifetime constraint for the resource binding problem of high-level synthesis. Due to two operations cannot share the same resource at the same control step, for each control step \( j \) and resource \( k \), we have the following constraint:

\[
\sum_{i \in O} x_{i,j,k} \leq 1 .
\]  \hspace{1cm} (Formula 8)

To handling the resource binding problem of high-level synthesis, an operation must be assigned to one and only one resource. Thus, for each operation \( i \), we have the following constraint:

\[
\sum_{k \in R(i)} x_{i,k} = 1 .
\]  \hspace{1cm} (Formula 9)

To handling the scheduling problem of high-level synthesis, each dependency relation in the DFG must be preserved. Thus, for each dependency relation \( i_1 \rightarrow i_2 \) in the DFG, we have the following constraint:
\[ \sum_{j = E_i}^{L_i} j \times x_{i,j} < \sum_{j = E_{i_2}}^{L_{i_2}} j \times x_{i_2,j}. \]  

(Formula 10)

Control Step

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>![Node 1]</td>
</tr>
<tr>
<td>3</td>
<td>![Node 2]</td>
</tr>
<tr>
<td>4</td>
<td>![Node 1]</td>
</tr>
<tr>
<td>5</td>
<td>![Node 2]</td>
</tr>
<tr>
<td>6</td>
<td>![Node 2]</td>
</tr>
<tr>
<td>7</td>
<td></td>
</tr>
</tbody>
</table>

**Figure 5**: Illustration for the perseverance of dependency relation.

We use Figure 5 to illustrate the perseverance of dependency relation. Suppose that there is a dependency relation \( i_1 \rightarrow i_2 \) in the DFG, the earliest control step of operation \( i_1 \) (i.e. \( E_i \)) is 2, the latest control step of operation \( i_1 \) (i.e. \( L_i \)) is 4, the earliest control step of operation \( i_2 \) (i.e. \( E_{i_2} \)) is 3, and the latest control step of operation \( i_2 \) (i.e. \( L_{i_2} \)) is 6. Figure 5 shows all the feasible scheduling results. We find that the constraint \( 2 \times x_{i_2,2} + 3 \times x_{i_1,3} + 4 \times x_{i_1,4} < 3 \times x_{i_2,3} + 4 \times x_{i_2,4} + 5 \times x_{i_2,5} + 6 \times x_{i_2,6} \), which is derived
from Formula 10, can exactly model all the feasible scheduling results. If operation $i_1$ is scheduled in control step 2 (i.e., $x_{i_1,2} = 1$), operation $i_2$ can be scheduled in control step 3 (i.e., $x_{i_2,3} = 1$), control step 4 (i.e., $x_{i_2,4} = 1$), control step 5 (i.e., $x_{i_2,5} = 1$), or control step 6 (i.e., $x_{i_2,6} = 1$); if operation $i_1$ is scheduled in control step 3, operation $i_2$ can be scheduled in control step 4, control step 5, or control step 6; if operation $i_1$ is scheduled in control step 4, operation $i_2$ can be scheduled in control step 5 or control step 6.

From the objective function, our goal is to minimize the number of TSVs. From Formula 1 to Formula 4, we accurately formulate the number of TSVs. From Formula 5 to Formula 7, we formulate the constraints on layer assignment. From Formula 8 to Formula 9, we formulate the constraints on resource binding. In Formula 10, we formulate the constraints on operation scheduling. Therefore, from Formula 1 to Formula 10, we have provided sufficient and necessary constraints for the simultaneous application of operation scheduling, resource binding, and layer assignment. As a consequence, our ILP formulation guarantees minimizing the TSV number under the simultaneous application of operation scheduling, resource binding, and layer assignment.

Finally, let’s use the example given in Section 2 to completely demonstrate our ILP formulation. The objective function is:

$$\text{minimize } (\text{via}_{A_i,S_i} + \text{via}_{A_i,C_i} + \text{via}_{A_i,M_i} + \text{via}_{A_i,M_2} + \text{via}_{S_1,A_i} + \text{via}_{S_1,C_i} + \text{via}_{S_1,M_i} + \text{via}_{S_1,M_2} + \text{via}_{C_1,S_i} + \text{via}_{C_1,A_i} + \text{via}_{C_1,M_i} + \text{via}_{C_1,M_2} + \text{via}_{M_2,A_i} + \text{via}_{M_2,S_i} + \text{via}_{M_2,C_i} + \text{via}_{M_2,M_i} + \text{via}_{M_2,M_2}).$$

Due to Formula 1, we have the following constraints:
\[x_{o_1,M_1} + x_{o_1,M_2} \leq 1 + D_{M_1,M_2},\]
\[x_{o_2,M_1} + x_{o_1,M_1} \leq 1 + D_{M_2,M_1},\]
\[x_{o_2,M_2} + x_{o_1,M_2} \leq 1 + D_{M_1,M_2},\]
\[x_{o_2,M_2} + x_{o_2,M_1} \leq 1 + D_{M_2,M_1},\]
\[\text{L} \quad \text{and so on.}\]

Due to Formula 2, we have the following constraints:
\[0 \leq \text{via}_{A_i,S_i},\]
\[0 \leq \text{via}_{A_i,C_i},\]
\[0 \leq \text{via}_{A_i,M_1},\]
\[0 \leq \text{via}_{A_i,M_2},\]
\[0 \leq \text{via}_{S_i,A_i},\]
\[0 \leq \text{via}_{S_i,C_i},\]
\[\text{L} \quad \text{and so on.}\]

Due to Formula 3, we have the following constraints:
\[1 \times r_{A_1,1} + 2 \times r_{A_1,2} + 3 \times r_{A_1,3} - 1 \times r_{S_1,1} - 2 \times r_{S_1,2} - 3 \times r_{S_1,3} \leq \text{via}_{A_i,S_i} + (1 - D_{A_i,S_i}) \times 3,\]
\[1 \times r_{A_1,1} + 2 \times r_{A_1,2} + 3 \times r_{A_1,3} - 1 \times r_{C_1,1} - 2 \times r_{C_1,2} - 3 \times r_{C_1,3} \leq \text{via}_{A_i,C_i} + (1 - D_{A_i,C_i}) \times 3,\]
\[\text{L} \quad \text{and so on.}\]

Due to Formula 4, we have the following constraints:
\[1 \times r_{S_1,1} + 2 \times r_{S_1,2} + 3 \times r_{S_1,3} - 1 \times r_{A_1,1} - 2 \times r_{A_1,2} - 3 \times r_{A_1,3} \leq \text{via}_{A_i,S_i} + (1 - D_{A_i,S_i}) \times 3,\]
\[1 \times r_{C_1,1} + 2 \times r_{C_1,2} + 3 \times r_{C_1,3} - 1 \times r_{A_1,1} - 2 \times r_{A_1,2} - 3 \times r_{A_1,3} \leq \text{via}_{A_i,C_i} + (1 - D_{A_i,C_i}) \times 3,\]
\[\text{L} \quad \text{and so on.}\]

Due to Formula 5, we have the following constraints:
\[r_{A_1,1} + r_{A_1,2} + r_{A_1,3} = 1,\]
\[r_{S_1,1} + r_{S_1,2} + r_{S_1,3} = 1,\]
\[r_{C_1,1} + r_{C_1,2} + r_{C_1,3} = 1,\]
\[\text{L} \quad \text{and so on.}\]
Suppose that the areas of adder, subtractor, multiplier, and comparator are 600 $\mu m^2$, 600 $\mu m^2$, 1000 $\mu m^2$, and 600 $\mu m^2$, respectively, and the footprint area constraint is 1600 $\mu m^2$. Due to Formula 6, we have the following constraints:

\[
600 \times r_{A_1} + 600 \times r_{S_1} + 600 \times r_{C_1} + 1000 \times r_{M_1} + 1000 \times r_{C_1} \leq 1600,
600 \times r_{A_2} + 600 \times r_{S_2} + 600 \times r_{C_2} + 1000 \times r_{M_2} + 1000 \times r_{C_2} \leq 1600,
600 \times r_{A_3} + 600 \times r_{S_3} + 600 \times r_{C_3} + 1000 \times r_{M_3} + 1000 \times r_{C_3} \leq 1600.
\]

Suppose that the power dissipations of adder, subtractor, multiplier, and comparator are 1 mW, 1 mW, 2 mW, and 1 mW, respectively. Due to Formula 7, we have the following constraints:

\[
1 \times r_{A_1} + 1 \times r_{S_1} + 2 \times r_{M_1} + 2 \times r_{C_1} \leq 1 \times r_{A_2} + 1 \times r_{S_2} + 2 \times r_{M_2} + 2 \times r_{C_2},
1 \times r_{A_2} + 1 \times r_{S_2} + 2 \times r_{M_2} + 2 \times r_{C_2} \leq 1 \times r_{A_3} + 1 \times r_{S_3} + 2 \times r_{M_3} + 2 \times r_{C_3}.
\]

Due to Formula 8, we have the following constraints:

\[
x_{o_1, M_1} + x_{o_2, M_1} + x_{o_3, M_1} + x_{o_4, M_1} + x_{o_5, M_1} \leq 1,
\]
\[
x_{o_1, M_2} + x_{o_2, M_2} + x_{o_3, M_2} + x_{o_4, M_2} + x_{o_5, M_2} \leq 1.
\]

Due to Formula 9, we have the following constraints:

\[
x_{o_1, M_1} + x_{o_2, M_1} = 1,
\]
\[
x_{o_2, M_2} + x_{o_3, M_2} = 1,
\]
\[
x_{o_3, M_1} + x_{o_4, M_2} = 1,
\]
\[
x_{o_4, S_1} = 1,
\]

Due to Formula 10, we have the following constraints:

\[
1 \times x_{o_1, i} + 2 \times x_{o_2, i} + 3 \times x_{o_3, i} + 4 \times x_{o_4, i} < 1 \times x_{o_1, i} + 2 \times x_{o_2, i} + 3 \times x_{o_3, i} + 4 \times x_{o_4, i},
1 \times x_{o_2, i} + 2 \times x_{o_3, i} + 3 \times x_{o_4, i} < 1 \times x_{o_1, i} + 2 \times x_{o_2, i} + 3 \times x_{o_3, i} + 4 \times x_{o_4, i},
\]

\[
\text{L}, \text{ and so on.}
\]
After solving the ILP formulation, we obtain the solution sol2: the operation scheduling result displayed in Figure 2(a) and the resource binding and layer assignment results displayed in Figure 2(b). Therefore, the number of TSVs obtained by our ILP approach is only 2. Compared with the solution sol1, which may be obtained by the previous work [9], the number of TSVs is reduced from 3 to 2. Note that our ILP approach can accurately minimize the number of TSVs (i.e., guarantee minimizing the number of TSVs) and directly report the number of TSVs.

4. GENERALIZATION

Following the assumption of previous approach [9], in Section 3, we do not consider the TSVs for the propagations of primary inputs and outputs. However, in fact, our approach can be easily generalized to consider the TSVs for the propagations of primary inputs and outputs. In this section, we make the generalization.

Without loss of generality, we use the TSV model given in [12] for explanation. From [12], there are two types of TSVs described as follows:

1. TSV_SIGNAL cell (with TSV_LAND cell): it is used to propagate the data from one layer to the neighboring layer (i.e., the upper layer or the lower layer).
2. TSV_IO cell: it is used to connect a primary input or output to a package pin. Only the first layer (i.e., layer 1) can contain TSV_IO cells.

Since the number of TSV_IO cells is fixed during the high-level synthesis stage, our optimization goal can be reduced to minimize the number of TSV_SIGNAL cells. Furthermore, strictly speaking, the term “TSV” used in Section 3 corresponds to the term “TSV_SIGNAL cell”\(^6\). Therefore, even if

\(^6\) The value of variable \(\text{via}_{k_1,k_2}\) (in our ILP formulation) corresponds to the number of TSV_SIGNAL cells from resource \(k_1\) to resource \(k_2\).
the propagations of primary inputs and outputs are considered, it is still not necessary to modify our ILP formulation (including the objective function and the constraints).

To fit primary inputs and outputs into our ILP formulation, we define virtual resources below. For each primary input (each primary output), we assume that there is a corresponding virtual resource to generate it (to receive it). Each virtual resource is assumed to be placed at layer 1. The area of each virtual resource is assumed to be 0. The power consumption of each virtual resource is also assumed to be 0. By using the concept of virtual resource, we can count the number of TSVs for the propagations of primary inputs and outputs (without any influence on the calculation of area and power consumption).

Let’s use the example given in Section 2 for illustration. In this example, there are 6 primary inputs and 3 primary outputs. Therefore, we define 9 virtual resources, including $V_1$, $V_2$, $V_3$, $V_4$, $V_5$, $V_6$, $V_7$, $V_8$, and $V_9$. The outputs of virtual resources $V_1$ and $V_2$ are the input of operation $O_1$, the outputs of virtual resources $V_3$ and $V_4$ are the input of operation $O_2$, the outputs of virtual resources $V_5$ and $V_6$ are the input of operation $O_{10}$, the input of virtual resource $V_7$ is the output of operation $O_5$, the input of virtual resource $V_8$ is the output of operation $O_9$, the input of virtual resource $V_9$ is the output of operation $O_{11}$. Then, the objective function becomes to be:

\[
\text{minimize } (\text{via}_{S_1, A_1} + \text{via}_{S_1, C_1} + \text{via}_{S_1, M_1} + \text{via}_{A_1, M_2} + \text{via}_{C_1, S_1} + \text{via}_{C_1, A_1} + \text{via}_{C_1, M_1} + \text{via}_{C_1, M_2} + \text{via}_{M_1, C_1} + \text{via}_{M_1, M_2} + \text{via}_{M_2, A_1} + \text{via}_{M_2, S_1} + \text{via}_{M_2, C_1} + \text{via}_{M_2, M_1} + \text{via}_{M_1, M_2} + \text{via}_{M_2, M_2} + \text{via}_{V_1, M_1} + \text{via}_{V_1, M_2} + \text{via}_{V_2, M_1} + \text{via}_{V_2, M_2} + \text{via}_{V_1, M_2} + \text{via}_{V_1, M_2} + \text{via}_{V_1, M_2} + \text{via}_{V_2, M_2} + \text{via}_{V_3, A_1} + \text{via}_{V_3, V_2} + \text{via}_{A_1, V_6} + \text{via}_{C_1, S_1} ).
\]
Note that all the constraints (from Formula 1 to Formula 10) shown in Section 3 still must remain. In addition, we have some extra constraints, which are derived from Formula 1, Formula 2, Formula 3, and Formula 4, to calculate the number of TSVs for the propagations of primary inputs and outputs. We elaborate these extra constraints below.

Owing to the propagations of primary inputs and outputs, we have the following extra constraints for Formula 1:

\[
1 + x_{vi,M_1} \leq 1 + D_{vi,M_1}, \\
1 + x_{vi,M_2} \leq 1 + D_{vi,M_2}, \\
1 + x_{vi,M_3} \leq 1 + D_{vi,M_3}, \\
1 + x_{vi,M_4} \leq 1 + D_{vi,M_4}, \\
1 + x_{o2,M_1} \leq 1 + D_{vi,M_1}, \\
1 + x_{o2,M_2} \leq 1 + D_{vi,M_2}, \\
1 + x_{o2,M_3} \leq 1 + D_{vi,M_3}, \\
1 + x_{o2,M_4} \leq 1 + D_{vi,M_4}, \\
\vdots, \quad \text{and so on.}
\]

Owing to the propagations of primary inputs and outputs, we have the following extra constraints for Formula 2:

\[
0 \leq via_{vi,M_1}, \\
0 \leq via_{vi,M_2}, \\
0 \leq via_{vi,M_3}, \\
0 \leq via_{vi,M_4}, \\
0 \leq via_{vi,M_5}, \\
0 \leq via_{vi,M_6}, \\
0 \leq via_{vi,M_7}, \\
0 \leq via_{vi,M_8}, \\
\vdots, \quad \text{and so on.}
\]

Since each virtual resource \( V \) is placed at layer 1, we have \( r_{v,j} = 1 \). Owing to the propagations of primary outputs, we have the following extra constraints for Formula 3:
\[1 \times r_{S_{1,1}} + 2 \times r_{S_{1,2}} + 3 \times r_{S_{1,3}} - 1 \leq \text{via}_{S_i, V_i} + (1 - D_{S_i, V_i}) \times 3,\]
\[1 \times r_{A_{1,1}} + 2 \times r_{A_{1,2}} + 3 \times r_{A_{1,3}} - 1 \leq \text{via}_{A_i, V_i} + (1 - D_{A_i, V_i}) \times 3,\]
\[1 \times r_{C_{1,1}} + 2 \times r_{C_{1,2}} + 3 \times r_{C_{1,3}} - 1 \leq \text{via}_{C_i, V_i} + (1 - D_{C_i, V_i}) \times 3.\]

Owing to the propagations of primary inputs, we have the following extra constraints for Formula 4:
\[1 \times r_{M_{1,1}} + 2 \times r_{M_{1,2}} + 3 \times r_{M_{1,3}} - 1 \leq \text{via}_{V_{i, M_i}} + (1 - D_{V_i, M_i}) \times 3,\]
\[1 \times r_{M_{2,1}} + 2 \times r_{M_{2,2}} + 3 \times r_{M_{2,3}} - 1 \leq \text{via}_{V_{i, M_i}} + (1 - D_{V_i, M_i}) \times 3,\]
\[1 \times r_{M_{3,1}} + 2 \times r_{M_{3,2}} + 3 \times r_{M_{3,3}} - 1 \leq \text{via}_{V_{i, M_i}} + (1 - D_{V_i, M_i}) \times 3,\]
\[1 \times r_{M_{4,1}} + 2 \times r_{M_{4,2}} + 3 \times r_{M_{4,3}} - 1 \leq \text{via}_{V_{i, M_i}} + (1 - D_{V_i, M_i}) \times 3,\]
\[1 \times r_{M_{5,1}} + 2 \times r_{M_{5,2}} + 3 \times r_{M_{5,3}} - 1 \leq \text{via}_{V_{i, M_i}} + (1 - D_{V_i, M_i}) \times 3,\]
\[1 \times r_{M_{6,1}} + 2 \times r_{M_{6,2}} + 3 \times r_{M_{6,3}} - 1 \leq \text{via}_{V_{i, M_i}} + (1 - D_{V_i, M_i}) \times 3,\]
\[1 \times r_{M_{7,1}} + 2 \times r_{M_{7,2}} + 3 \times r_{M_{7,3}} - 1 \leq \text{via}_{V_{i, M_i}} + (1 - D_{V_i, M_i}) \times 3,\]
\[1 \times r_{M_{8,1}} + 2 \times r_{M_{8,2}} + 3 \times r_{M_{8,3}} - 1 \leq \text{via}_{V_{i, M_i}} + (1 - D_{V_i, M_i}) \times 3,\]
\[1 \times r_{M_{9,1}} + 2 \times r_{M_{9,2}} + 3 \times r_{M_{9,3}} - 1 \leq \text{via}_{V_{i, M_i}} + (1 - D_{V_i, M_i}) \times 3,\]
\[\ldots\]

5. EXPERIMENTAL RESULTS

We use Extended LINGO Release 11.0 as the ILP solver. The platform is Windows 2003 x64 running on Intel Xeon E5355 CPU. Ten benchmark circuits are used to test the effectiveness of our approach. Circuits HAL, BF, AR, and EWF are adopted from [13], circuits G2, G5, R1, and R2 are adopted from [6], and circuits IDCT2 and Dist2 are adopted from MediaBench high-level synthesis benchmark suite [14]. Table II tabulates the characteristics of benchmark circuits. The column #ops denotes the number of operations in the DFG. The column #vars denotes the number of variables in the DFG. The column #steps denotes the number of control steps.
Table II. Characteristics of benchmark circuits.

<table>
<thead>
<tr>
<th>Circuit</th>
<th>#ops</th>
<th>#vars</th>
<th>#steps</th>
</tr>
</thead>
<tbody>
<tr>
<td>HAL</td>
<td>11</td>
<td>8</td>
<td>4</td>
</tr>
<tr>
<td>BF</td>
<td>29</td>
<td>30</td>
<td>8</td>
</tr>
<tr>
<td>AR</td>
<td>28</td>
<td>30</td>
<td>8</td>
</tr>
<tr>
<td>EWF</td>
<td>34</td>
<td>47</td>
<td>14</td>
</tr>
<tr>
<td>G2</td>
<td>24</td>
<td>31</td>
<td>8</td>
</tr>
<tr>
<td>G5</td>
<td>28</td>
<td>31</td>
<td>7</td>
</tr>
<tr>
<td>R1</td>
<td>82</td>
<td>91</td>
<td>11</td>
</tr>
<tr>
<td>R2</td>
<td>108</td>
<td>114</td>
<td>12</td>
</tr>
<tr>
<td>IDCT2</td>
<td>96</td>
<td>200</td>
<td>24</td>
</tr>
<tr>
<td>Dist2</td>
<td>326</td>
<td>325</td>
<td>86</td>
</tr>
</tbody>
</table>

Without loss of generality, in our experiments, we assume that all the resources are 8-bit designs. The modules in the Synopsys DesignWare are used to implement the following resources: adder, subtractor, ALU, multiplier, divisor, selector, and comparator. Further, these resources are targeted to TSMC 0.18μm process technology. The power consumption of adder, subtractor, ALU, multiplier, divisor, selector, and comparator are 428 μW, 557 μW, 572 μW, 1872 μW, 1920μW, 214μW, 528 μW, respectively. The areas of adder, subtractor, ALU, multiplier, divisor, selector, and comparator are 4892 μm², 6326 μm², 6950 μm², 21455 μm², 22840 μm², 2450μm², and 9147 μm², respectively.

We compare our approach with the previous approach [9]. Table III tabulates the comparisons on the number of TSVs and the CPU time. The column Design Constraints include the number of layers (N\text{layer}), the footprint area (A\text{max}), the number of control steps (C\text{step}), and the resource constraints. In circuits HAL, BF, AR, EWF, G2, G5, and R1, we use 4-tuple (#add,#sub,#mul,#com) to represent the resource constraints, where #add, #sub, #mul#, and #com denote the numbers of adders, the number of

---

Note that the previous approach [9] does not consider the propagation of primary inputs and outputs. Therefore, for a fair comparison, in our experiments, our approach also does not take the propagation of primary inputs and outputs into account.
subtractors, the number of multipliers, and the number of comparators, respectively. In circuits R2, IDCT2 and Dist2, we use 5-tuple (#alus,#muls,#divs,#sels,#comps) to represent the resource constraints, where #alus, #muls, #divs, #sels, and #comps denote the number of ALUs, multipliers, divisors, selectors, and comparators, respectively. The column #vias give the number of TSVs. Since our objective function is accurate, in each benchmark circuit, our approach always has less number of TSVs than that of the previous approach [9]. In average, we find that our approach can reduce 44.1% TSV number. The columns Time gives the CPU time. In small circuits, the CPU times of our approach are close to those of previous approach. However, in large circuit, our approach often takes a larger CPU time. The main reason is that, in order to accurately formulate the TSV number, some constraints of our approach, e.g., Formula 3 and Formula 4, are more complex than those of the previous approach. As a result, our approach takes a little CPU time penalty in large circuits. In summary, our approach takes more about 29.0% CPU time penalty than that of the previous approach.
Table III. Comparisons between the previous work [9] and ours approach.

<table>
<thead>
<tr>
<th>Circuit</th>
<th>Design Constraints</th>
<th>Previous Approach [9]</th>
<th>Ours</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Layer</td>
<td>Area (μm²)</td>
<td>C_step</td>
</tr>
<tr>
<td>HAL</td>
<td></td>
<td>3</td>
<td>27781</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4</td>
<td>21455</td>
</tr>
<tr>
<td>BF</td>
<td></td>
<td>3</td>
<td>23608</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4</td>
<td>21455</td>
</tr>
<tr>
<td>AR</td>
<td></td>
<td>2</td>
<td>44490</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4</td>
<td>22245</td>
</tr>
<tr>
<td>EWF</td>
<td></td>
<td>3</td>
<td>42900</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4</td>
<td>21455</td>
</tr>
<tr>
<td>G2</td>
<td></td>
<td>3</td>
<td>24736</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4</td>
<td>21455</td>
</tr>
<tr>
<td>G5</td>
<td></td>
<td>2</td>
<td>17229</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3</td>
<td>11486</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4</td>
<td>9784</td>
</tr>
<tr>
<td>R1</td>
<td></td>
<td>3</td>
<td>90109</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4</td>
<td>67582</td>
</tr>
<tr>
<td>R2</td>
<td></td>
<td>3</td>
<td>78220</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4</td>
<td>92203</td>
</tr>
<tr>
<td>IDCT2</td>
<td></td>
<td>3</td>
<td>48868</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4</td>
<td>67377</td>
</tr>
<tr>
<td>Dist2</td>
<td></td>
<td>3</td>
<td>27943</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4</td>
<td>42245</td>
</tr>
</tbody>
</table>

ratio    | 1   | 1   | 0.559 | 1.290 |

6. CONCLUSIONS

In this paper, we demonstrate that the previous approach does not accurately minimize the TSV number in high-level synthesis of 3D ICs. Then, we propose a new ILP formulation to perform accurate TSV number minimization. Instead of maximizing the number of same-layer operation-level data-transfers, our approach directly minimizes the number of cross-layer resource-level data-transfers. As a result, our approach can accurately minimize the number of TSVs. Compared with the previous
work, benchmark data show that our approach can reduce 44.1% TSV number. However, in order to accurately formulate the TSVs number, some constraints of our approach are more complex so that our approach takes a little CPU time penalty in larger circuits.

Besides, it should be mentioned that, since solving the ILP formulation is an NP-hard problem, there is a demand to develop a heuristic algorithm for large circuits. We leave this topic as our future work.

7. ACKNOWLEDGEMENTS

This work was supported in part by the National Science Council of Taiwan, R.O.C., under grant number NSC 97-2221-E-033-053-MY3.

REFERENCES


