

Design Planning with 3D-Via Optimization in Alternative Stacking Integrated Circuits

CHAO-HUNG LU, HUNG-MING CHEN* AND CHIEN-NAN JIMMY LIU

Department of Electrical Engineering

National Central University

Taoyuan, 320 Taiwan

E-mail: {chlu; jimmy}@ee.ncu.edu.tw

**Department of Electronics Engineering*

National Chiao Tung University

Hsinchu, 300 Taiwan

E-mail: hmchen@mail.nctu.edu.tw

Billions of transistors are placed in one single chip (SoC) with advanced manufacturing technology. Further development is obstructed by the ability to the manufacture of SoC and the signal integrity. Stacking IC is an alternative choice when we design a high-performance high-density chip. Design flow (especially physical design) is facing different issues when compared with 2D IC design. The location of the I/Os seriously affect the number of 3D-Vias and their total area in the stacking IC. This paper proposes a Stacking IC architecture and the corresponding design flow to solve the I/O and 3D-Via problems. In this flow, we have developed a system partition approach to minimize the number of 3D-Vias and balance the I/O number of each tier, and modified one traditional floorplan method to optimize the I/O and module locations. The experimental results are encouraging in the GSRC benchmarks. Compared with greedy and intuitive methods, our framework reduces the number of 3D-Vias by 30.02% on the average and can balance the I/O count of each tier. The dead space of the final floorplan is reduced by 14.13%.

Keywords: stacking IC, partition, floorplanning, TSV, 3D-via

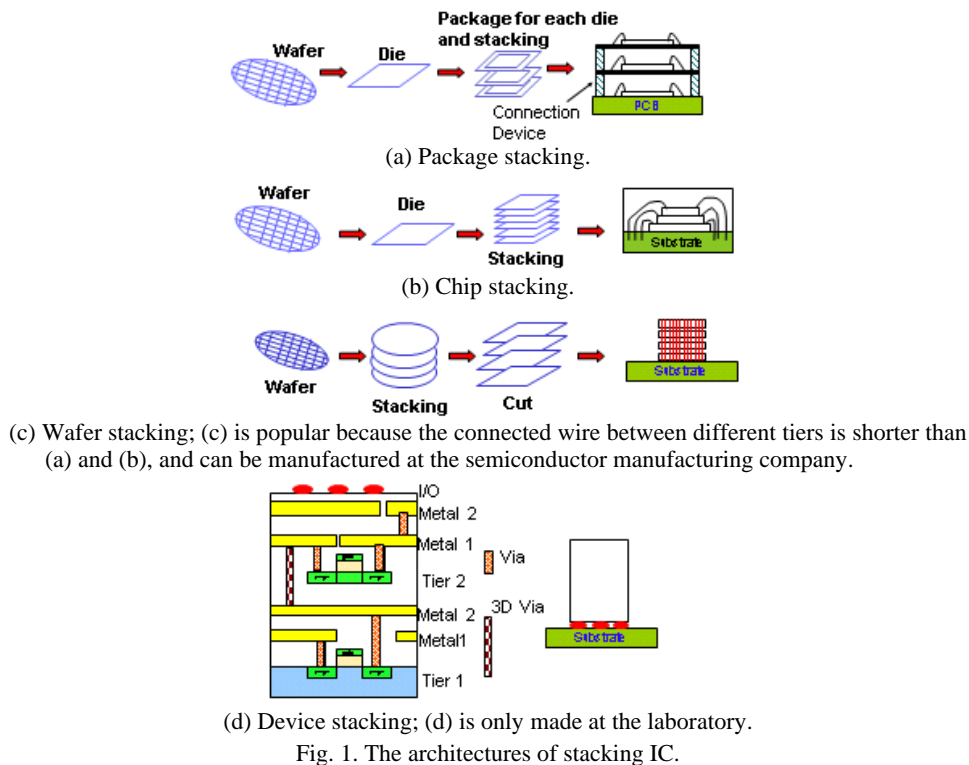
1. INTRODUCTION

The trend in the chip design is to integrate multi-function into one chip, and simultaneously improve the size, power, performance and cost of the chip. To accomplish this purpose, semiconductor manufacturing companies need to continually increase the per square inch transistor number on integrated circuits and the manufacturing technology needs to be continually advanced. The resistance in the connected wire is raised substantially as the Very Large Scale Integration (VLSI) technology enters the nanometer era. The resistance in the wire seriously impacts the performance of the chip if the semiconductor manufacturing company uses the advanced technology to make 2-D IC.

Stacking Integrated Circuit (IC) is promising to the development of a high-density high-performance IC. Stacking technology stacks a die (chip, wafer) over another die (chip, wafer). The total wirelength and size of the chip shrink by vertical interconnecting when using Stacking IC with 2-D IC. The performance and power consumption can be greatly improved because the resistance of the total wirelength is decreased. Stacking ICs in modern researches are called Three-dimensional (3-D) ICs [16] or System in Package

Received January 5, 2009; revised June 17, 2009; accepted August 20, 2009.

Communicated by Yao-Wen Chang.



(SiP) [17]. Stacking ICs can be classified into four types: (1) package stacking; (2) chip stacking; (3) wafer stacking; (4) device stacking. The difference between each type is shown in Fig. 1. The chip in package stacking is packaged before stacking, as Fig. 1 (a) shows. Chip stacking [19] ICs stack dies before packaging, as Fig. 1 (b) shows. Wafer stacking fabrication ([1, 11]) stacks the wafer before cutting, as Fig. 1 (c) shows. The size of wafer stacking IC is smaller than chip stacking IC. The size and the performance of device stacking ICs are better than wafer stacking ICs and the architecture as shown in Fig. 1 (d). Because modern manufacturing technologies for device stacking [8] ICs are inexperienced, it cannot be manufactured by semiconductor manufacturing companies.

In [2], the authors directly utilized traditional 2-D tools to design a stacking IC. This method cannot be used in modern design because the vertical factors are ignored in traditional 2-D tools. [4] proposed a better design flow of stacking IC. It use complexity method to design stacking ICs. The role of EDA (Electronic Design Automatic) tools in stacking IC is described in [5, 6]. Comparing stacking ICs with 2-D ICs, front-end design flows are the same and the physical design flow should be reformed ([10, 24]). In the stacking IC, physical design methodologies can be classified into two categories: innovation and improving existing approaches.

The former denotes that the problem does not exist in the 2-D design, *e.g.*, 3D-Via placement, thermal via placement. The latter one denotes that problems arise in the 2-D design and the difficulty is increased in stacking ICs, *e.g.*, area-driven floorplan, noise-driven routing.

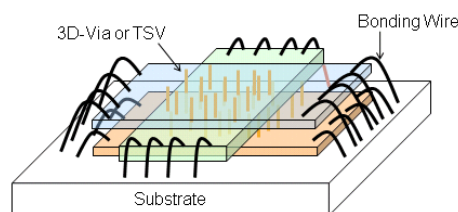


Fig. 2. The alternative stacking architecture. All I/O pins are placed at sides in this architecture.

Wafer stacking and device stacking ICs use the vertical vehicle (Through Silicon Via (TSV) or 3D-Via) for connecting different tiers and the locations of these vertical vehicles are set under the die. The tier denotes the number of wafers in the chip. The performance and power delivery attenuate when the data and power are delivered from the bottom tier to the top tier because the resistance of the vertical vehicle is larger than the traditional via. In our previous paper [25], we propose a stacking IC architecture to improve these problems. A defect of this architecture is that the bonding wires of the lower tiers are often obstructed. This paper proposes the alternative stacking IC to improve these problems and the architecture is shown in Fig. 2. The architecture must be used for multi-tier ICs (> 3). There are several advantages to this architecture. The power dissipation from substrate to CMOS is smaller than the traditional architecture and the performance increases because the resistance decreases.

When developing stacking IC tools ([20, 21]), we should simultaneously consider many problems, such as area, wirelength, congestion, signal integration and the number of vertical vehicles. In the alternative stacking architecture, the I/O pad balance for each tier is considered since it critically affects the total area. [12] directly balances the number of I/Os for each tier, and integrates hMetis [13] and the simulated annealing [15] method to optimize the number of 3D-Via. Unfortunately, the effect between the vertical vehicle and the I/O location is ignored.

This paper enhances the method proposed in [25] to improve the solution quality. It proposes a design flow of the stacking IC and develops a system partition methodology based on the Fiduccia-Mattheyees (F-M) [9] algorithm to plan I/Os and modules. The system partition methodology simultaneously considers the number of vertical vehicles and the number of I/Os for each tier. The modern floorplan method is successfully combined in this design flow. The contributions of this paper are summarized as follows,

- This paper proposes a practical design flow in stacking ICs. The vertical factor is divided into several 2-D problems to simplify the complexity of the stacking IC design.
- This paper presents a system partition method to balance the area and minimize the 3D-Via number improving the area and the yield.
- This paper shows that the aspect ratio of floorplanning is an important factor to obtain a better area result in stacking IC.

The rest of the paper is organized as follows. Section 2 describes our design flow in stacking IC and problem formulation for the system partition. The system partition approach is presented in section 3. Experimental results are shown in section 4, and section 5 presents conclusions.

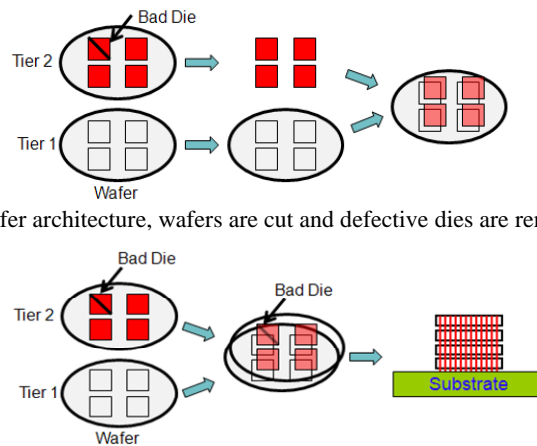
2. STACKING IC MODELS AND DESIGN FLOW

Stacking ICs have different architectures. This section briefly describes the diversity of all stacking ICs. The alternative stacking IC is used to improve the performance of wafer stacking IC. The alternative stacking IC has special characteristics. This study focuses primarily on the effect of these characteristics. The goal of this paper is defined in the last subsection.

2.1 3D-Via and Stacking IC Models

The 2-D chip design uses via to connect different metal layers. A similar vehicle is used to connect different tiers in stacking IC, as mentioned in the following works: VILIC (vertical inter-layer interconnects) in [22], 3D-Via in [12], Through Silicon Via in [23]. These vehicle devices are called 3D-Via in this paper.

Stacking IC can be classified into four types, as shown in Fig. 1. Compared package stacking IC with others, the chip area is biggest because all chips should be packaged before stacking, as Fig. 1 (a) illustrates. The technology of package stacking IC is more mature than others, and the yield is the highest. Chip stacking ICs stack up dies before packaging, as Fig. 1 (b) illustrates. The performance and the size are better than package stacking IC. The I/Os are placed around the core. The performance and size of wafer stacking IC are better than package stacking and chip stacking ICs. Besides, 3D-Via could be utilized to connect different tiers. Wafer stacking is subdivided into two types in [18]: (1) chip-to-wafer and (2) wafer-to-wafer. The difference is shown in Fig. 3. In the manufactured process of chip-to-wafer ICs defective dies are removed before stacking [11], therefore increasing the yield. The disadvantage of the chip-to-wafer stacking is that the lower tier area is larger than the higher tiers and the spacing between die to die is wider than wafer-to-wafer IC. Wafers should be aimed before stacking in wafer-to-wafer ICs [1]. The disadvantage of the wafer-to-wafer stacking is that the defective die is used in the stacking



(a) In the chip-to-wafer architecture, wafers are cut and defective dies are removed before stacking.

(b) In the wafer-to-wafer architecture, defective dies are removed after the wafers are cut.

Fig. 3. Wafer stacking ICs can be subdivided into the chip-to-wafer architecture and the wafer-to-wafer architecture.

Table 1. The comparison table of all classified stacking ICs.

	Package Stacking	Chip Stacking	Wafer Stacking		Device Stacking
			Chip-to-wafer	Wafer-to-wafer	
Yield	Highest	High	Normal	Low	Lowest
Size	Highest	Huge	Normal	Small	Smallest
Performance	Slowest	Slow	Normal	Normal	Fastest

even though the defective die is found, as shown in Fig. 3 (b). Device stacking [8] ICs are complex and hard, according to current manufacturing technology, and exist only in the laboratory. The illustration is shown in Fig. 1 (d). A comparison of all classified stacking ICs is shown in Table 1.

In wafer stacking ICs, the locations of I/Os are set under the die, as the flip-chip architecture. The flip-chip architecture is a mesh structure, and the VDD and GND bumps are uniformly distributed across the die with signal bumps in a fixed interspersed location. This architecture can make a high-performance IC in the 2-D design. The main reason is that the distance from I/O to the connection point of the core is shorter than the traditional peripheral I/O architecture. The performance of the chip is substantially improved.

A defect results when the locations of I/Os are set below the die in wafer stacking ICs. The resistance of 3D-Via is greater than the traditional via. The resistance and the noise gradually increase when the data and power are delivered from the bottom tier to the top tier by 3D-Via. The alternative stacking IC solves the resistance and noise problems and the architecture is shown in Fig. 2. The I/Os are placed on sides of the chip in alternative stacking IC. The data and the power can be directly delivered to the higher tier by the bonding wire, besides, the number of 3D-Vias is greatly decreased.

2.2 Design Flow of Alternative Stacking IC

To successfully design high-efficacy stacking ICs, the traditional 2-D design flow should be slightly modified. The proposed design flow is shown in Fig. 4 (a). The vertical factor is ignored before the partition step. Traditional 2-D tools, (such as hMetis [13]), are still in use. The connected relationship between modules and I/Os can be built during the partition step. The diagram of the connected relation is shown in Fig. 4 (b). In the tradition physical design, the floorplan step follows the partition step. The authors in [10] believe that the floorplanning tool should thoroughly consider the vertical factor in stacking ICs. A modern floorplan tool should simultaneously consider many problems, such as area, wirelength, congestion, and signal integration. The burden of the floorplan greatly increases if the vertical factor is added. To solve this problem, we add one system partition step before the floorplan step, as shown in Fig. 4 (a). The goal of the system partition is to transform the stacking problem into several 2-D IC problems.

Compared stacking ICs with 2-D ICs, the planning of I/Os and modules are complex because I/Os and modules should be planned into several tiers. Due to the planning of I/Os and modules in stacking IC, all modules and I/Os in the system partitioning step should be placed in a suitable tier to minimize the nets cut, and then the 2-D floorplanning tool is used for each tier. Most existing 2-D partitioning approaches, Kernighan-Lin [14], Fiduccia-Mattheyes (F-M) [9], CLIP/CDIP [7] and hMetis [13], help to achieve this work.

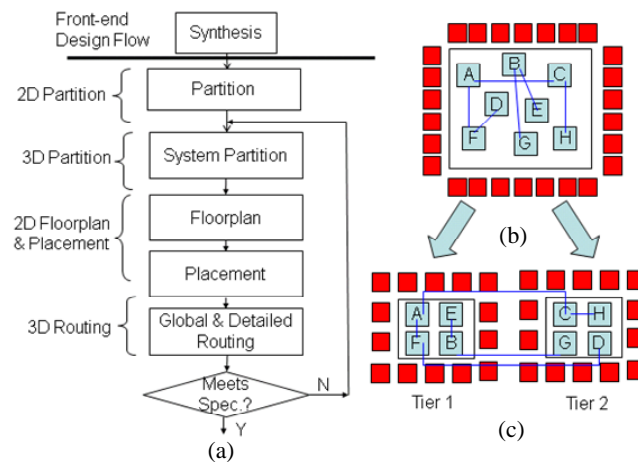


Fig. 4. Compared stacking ICs with 2-D ICs, the difference of the design flow is at the partition step [24]. This paper proposes a design flow for stacking ICs, as shown in (a). This design flow adds one system partitioning step before floorplanning. The module and the I/O are assigned to several tiers, as shown in (b) and (c). The complexity of stacking IC design can be greatly decreased.

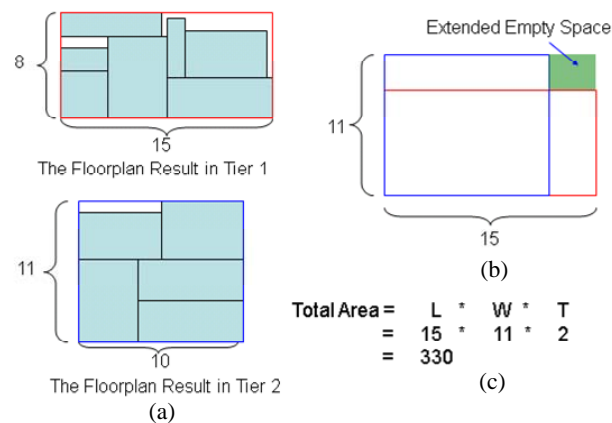


Fig. 5. The effect of aspect-ratio in the floorplan; (a) A floorplan result; (b) The maximum length and the maximum width used to compute the total area. If the difference between the maximum length and the maximum width is slightly increased, the empty space of the chip increases; (c) The area computation.

hMetis is the popular tool to provide a high quality partition result and short run time. hMetis can handle the number of 3D-Via and the module area of each tier, but the I/O balance is not considered. hMetis should be modified if it is used to place all components. The action and the performance are not expected when hMetis is modified. Therefore, we propose a system partition method to solve these problems. The number of I/Os and the module area for each tier are obtained after system partitioning. The next step is to plan the locations of modules. The 2-D chip design usually utilizes the simulated annealing [15] and guided moves to find the better results, and the aspect ratio factor is usually ignored.

The chip area would be greatly increased if the aspect ratio is continually ignored in the proposed stacking IC.

An example can help to present the significance of the aspect ratio in alternative stacking IC. The tier number is set as two. The modules are averagely planned into these two tiers, and the area-driven floorplan results are shown in Fig. 5 (a). The area in Tier 1 is $120\mu\text{m}^2$ and the area in Tier 2 is $110\mu\text{m}^2$. The intuitive computation of total area is $120 + 110 = 230\mu\text{m}^2$. The real computation of the total area in stacking IC is shown as follows,

$$\begin{aligned} \text{Total_Area} &= L \times W \times T, \\ L &= \max(\text{Tier_Lengt}_i, 1 \leq i \leq T), \\ W &= \max(\text{Tier_Width}_i, 1 \leq i \leq T), \end{aligned} \tag{1}$$

where T denotes the tier number. Based on the calculation, the empty space is increased at the corner, as Fig. 5 (b) shows. The total area of the chip is $330\mu\text{m}^2$, as Fig. 5 (c) shows. If we use the aspect-ratio-driven floorplan to plan the module locations, the extended empty space can be greatly improved.

Traditional placement tools are still used in the placement stage because the vertical problem is solved at partitioning and floorplanning. Routing complexity is then increased, due to the large 3D-Via resistance and the increased number of routing layers.

2.3 The Impact of I/O Location in Alternative Stacking IC

Peripheral-I/Os have minimal area constraint in the chip design. In Fig. 6, the size of I/O is 3×3 and the area constraint is 18×18 . If the core area is smaller than this constraint, the area is fixed at 24×24 . In alternative stacking IC, this problem is more critical. The area would not be extended if the number of I/Os for each tier is balanced, as Fig. 7 (a) shows. The area must be extended if the number of I/Os for each tier is unbalanced, as shown in Fig. 7 (b). The reason for the extension is that I/Os of the odd and the even tier cannot overlap. The result of the extension is an enlargement of the chip area. Besides the area problem, I/O locations affect the number of 3D-Via when the locations of I/Os and the connected components are placed at different tiers.

This paper sets a constraint to control I/O locations and to balance the I/O number for each tier. The constraint is that the difference of the I/O amount between any two tiers is smaller than $x - 1$, where x denotes the number of tiers.

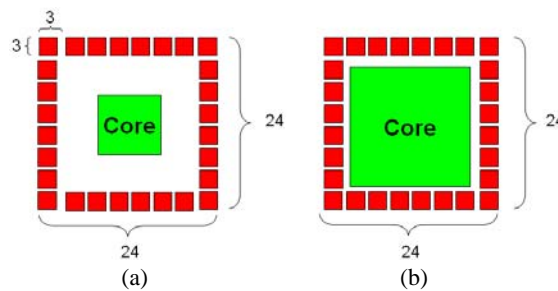


Fig. 6. In the 2-D chip design, all I/Os should be placed into one tier. Therefore, the area of I/O is fixed. The size of the core in (a) is smaller than that in (b), but the areas are both the same.

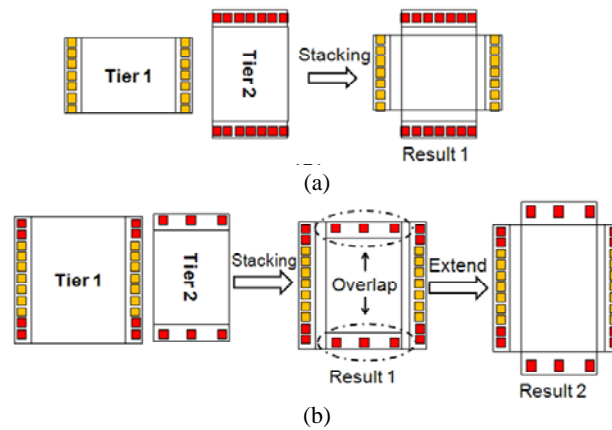


Fig. 7. In alternative stacking IC, all I/Os could be placed into several tiers. If the I/O number for each tier is balanced, as (a) shows, the area can be fully utilized by the core. If not, as shown in (b), the area will be extended.

2.4 Problem Formulation

The area and the 3D-Via number in alternative stacking IC increases when I/Os and modules are placed in an unsuitable location. The goal of our system partition is to plan I/Os and modules in suitable tiers and to balance each tier area. The problem is formulated as follows.

Given a set of modules, M_1, M_2, \dots, M_x , the number of tiers, T , a set of I/Os, IO_1, IO_2, \dots, IO_y , and the connection information of each nets, N_1, N_2, \dots, N_z , find a solution such that all modules and all I/Os assign to the suitable tier to obtain the minimal 3D-Via number. At the same time, the I/O number and area for each tier must be balanced.

3. I/Os AND MODULES PLANNING WITH MINIMAL 3D-VIA NUMBER IN ALTERNATIVE STACKING ICs

This paper develops a two-step methodology to plan modules and I/Os at the system partition. We firstly use the global planning method to plan modules and I/Os. This method balances the I/O number and the area for all tiers, and reduces the 3D-Via number. We then use a Congestion-driven Planning and Iterative Refinement (CPIR) method to plan the I/Os into the suitable tier to minimize the 3D-Via number.

3.1 Global Planning for I/Os and Modules

Here we use an example to explain the relation between I/O locations and the 3D-Via number. In Fig. 8, all **blackened** blocks should be connected together. This example requires the spending of two 3D-Via to connect the I/Os of Tier 2 and four 3D-Via to connect the I/O of Tier 3 if the core module is set at Tier 1 and the I/O block is fixed. The total 3D-Via number is six. If the core module sets into Tier 2, it only spends four 3D-Vias to connect all blocks. Similarly, the number of 3D-Via could improve when the core module

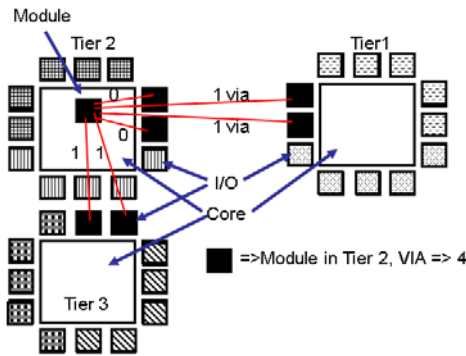


Fig. 8. The locations of the modules would affect the number of 3D-Via. In this case, I/Os are fixed and the optimal location of the module is on Tier 2. If the module is placed on Tier 1 or Tier 3, we should use six vias to connect the modules and I/Os.

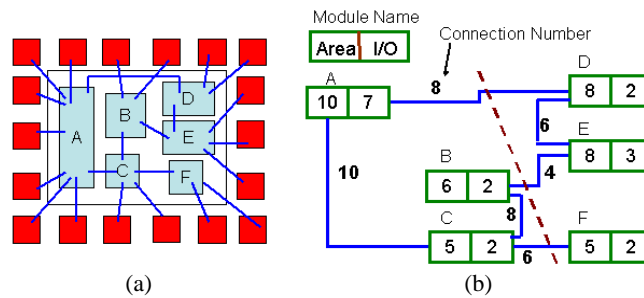


Fig. 9. The connection graph of the circuit built by the netlist file, as shown in (a); (b) denotes the equivalent graph of (a).

is fixed and the location of I/O blocks changes. To decrease the 3D-Via number, we formulate a rule to help plan I/Os and modules. The rule is defined as follows.

Rule 1: If a module is placed at Tier α , connected I/Os should be placed at Tier β , $\max(\alpha - 1, 1) \leq \beta \leq \min(\alpha + 1, T)$.

In other words, I/Os are placed on three consecutive tiers and the connected module is placed on the middle tier. The core is set on Tier 1 in Fig. 8, I/Os can only be placed on Tiers 1 and 2 only. If the core is set at Tier 2, I/Os could be placed on Tiers 1, 2, and 3.

The connected graph of the circuit can be built after the synthesis and partition, as shown in Fig. 9 (a). We extract the information for areas, interconnections, and I/Os to construct the equivalent graph, as Fig. 9 (b) illustrates. We then use the F-M algorithm [9] to plan modules into each tier. The module area balance factor for all tier and 3D-Via number is considered into our FM algorithm. After tier location of all modules is obtained, we use Rule 1 to plan the I/O location.

After global planning, final module locations and initial I/O locations can be obtained. Because **Rule 1** is a simply assigned method for I/Os, we propose another method, Congestion-driven Planning and Iterative Refinement (CPIR), to optimize I/O locations and to minimize the 3D-Via number.

3.2 I/O Allocation by Congestion-driven Planning and Iterative Refinement

Here we propose the Congestion-driven Planning and Iterative Refinement (CPIR) method to improve the previous result and minimize the 3D-Via number, as illustrated in Fig. 10. In CPIR, we firstly build one connected graph (line 1), and then sort the I/Os priority (line 2). The I/Os priority is inversely proportional to the connected tier number. In section 3.1, we explain the relation between the 3D-Via number, I/O and module. If the I/O and the connected module are plan at the same tier, we do not need to spend extra 3D-Via to build the interconnection and the cost of 3D-Via is zero. The zero cost candidates are less when I/O has less connected tier number. These I/Os which has less connected tiers should be planned first. The higher priority could be obtained when I/O has less connection tiers. We then explain the “available I/O space”. According to our input information, the tier and I/O number can be obtained. The ideal I/O number for each tier is (I/O number)/(Tier Number). If the allocated I/O number of the tier is smaller than the ideal I/O number, this tier has available I/O space. In the next step, I/Os would be allocated into suitable tiers to minimize the 3D-Via number. The allocated I/O is called I/O_x . If connected tiers of I/O_x have available I/O spaces, I/O_x is assigned to the optimal tier (line 5). The optimal tier denotes that the available I/O spaces of the tier are more than other tiers. If connected tiers of I/O_x do not have available I/O spaces, we would try to modify previous I/Os, I/O_z , $1 \leq z \leq x-1$, to make another assigned result (lines 7-12). The total number of the connected via is constant and the I/O locations are slightly modified. I/O_x and the connected module are placed on the same tier, and the number of the 3D-Via is constant. The time complexity for CPIR is $O(n \times m)$, where n denotes the I/O number and m denotes the tier number.

An example can help to explain CPIR. In this example, the tier number is set as three, the module number is set as eight and the I/O number is six, No. 1, No. 2, ..., No. 6. The goal of our planned method is to averagely place all I/O into three tiers. The optimal number

```

CPIR Method
Input : a set of I/Os,  $IO_1, IO_2, \dots, IO_y$ , Tier number, the location of all modules,
        Connected relations
Output : I/O locations, module locations
Objection : minimize the via number and balance the I/O number for each tier
1. According to input information, builds a relation graph.
2. Sorting ( I/Os(connected tier number) )
3. For each  $I/O_x$ 
4.   If ( connected tiers have available I/O spaces )
5.      $I/O_x$  assigns into  $max(\text{available I/O spaces})\text{tier}$ 
6.   Else
7.     If ( connected tier of previous ( $I/O_z, 1 \leq z \leq x-1$ ) > 1 )
8.       iterative re-plan previous ( $I/O_z, 1 \leq z \leq x-1$ ) until  $I/O_x$  can be
9.       inserted into the tier
10.    Else
11.       $I/O_x$  inserts into the nearest tier
12.    End If
13.  End If
14. End For

```

Fig. 10. The pseudo code of congestion-driven planning and iterative refinement. The 3D-Via number can be greatly decreased by this algorithm.

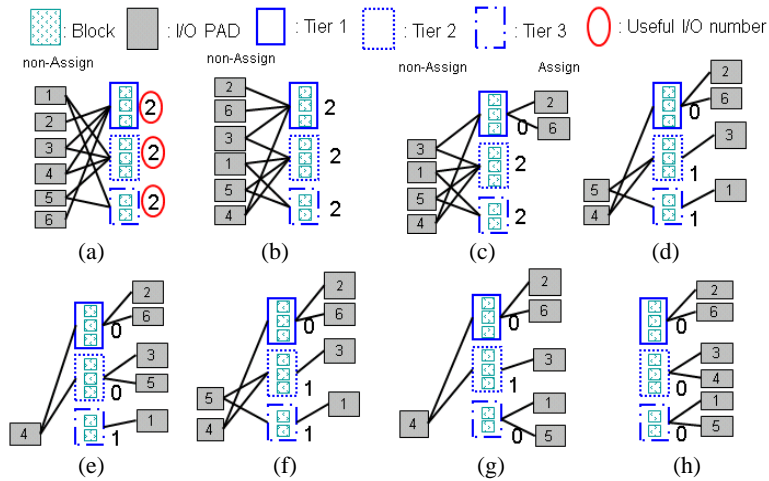


Fig. 11. The example of the CPIR method; (a) The first step builds the relation graph; (b) The connection tier number decides the I/O priority; The congestion method assigns the I/O location, as (c) and (d); If the tier does not have enough space, we would use the iterative re-plan method to change previous I/O at the same tier, as shown in (e) to (h).

of I/Os for each tier is two. We firstly build the relation graph as shown in Fig. 11 (a). Then, according to the number of the connection tier, the I/O priority is decided, as shown in Fig. 11 (b). If the number of connected tier of I/O is small, the higher priority would be obtained. Because No. 2 and No. 6 have only one connected tier, we assign No. 2 and No. 6 to Tier 1, as shown in Fig. 11 (c). Now, two I/O pins are assigned to Tier 1. According to the relation graph, No. 3 could be assigned to Tier 1 or Tier 2. If No. 3 is assigned to Tier 1, the I/O number is bigger than the optimal number. Therefore, we assign No. 3 into Tier 2. No. 1 could be assigned into Tier 2 or Tier 3. Because the space of Tier 3 is more than Tier 2, No. 1 is assigned to Tier 3. The illustration of the assigned result is shown in Fig. 11 (d). No. 5 could be assigned into Tier 2 or Tier 3. Because the available space of Tiers 2 and 3 are the same, we randomly choose the assignable tier. The assigned result of No. 5 is shown in Fig. 11 (e). According to the relation graph, No. 4 could be assigned into Tier 1 or Tier 2. Tiers 1 and 2 do not have the spare space. We use the iterative re-plan method to modify I/O locations. In this case, we change the location of No. 5, as shown in Figs. 11 (f) and (g). Therefore, No. 4 could be assigned to Tier 2 and the via number is constant, as shown in Fig. 11 (h).

Finally, we summarize our modules and I/Os planning method. Firstly, the global plan method is used to plan the final location of modules and initial location of I/Os. This method uses the F-M method and Rule 1 to implement. The description is shown in section 3.1. We then use the CPIR method to improve the global plan result. Therefore, all modules and I/O can be planned at the suitable tier and 3D-via number can be improved.

4. EXPERIMENTAL RESULTS

The proposed modules and I/Os planning method and floorplanner have been implemented using C++ language on an Intel Pentium 4 2.4GHz machine with 1G memory.

Five GSRC benchmark circuits, n30, n50, n100, n200 and n300 are used to prove the efficiency of the proposed methodology. Since there is no literature on the I/O planning method in the floorplan level for the stacking chip design, we compared our approach with a greedy method. The greedy method only considers the I/O balance factor, and the 3D-Via number are ignored. The greedy method is explained as follows. The module and the connected I/Os are planned at the same tier. If the real I/O number is more than the ideal I/O number, these redundant I/Os are moved to the second candidate location. The second candidate location is the max (available I/O space of x), where $1 \leq x \leq T$, T is the tier number.

Table 2 compares the total 3D-Vias number. The first row denotes the number of tiers, and the last row denotes the average result for all benchmarks. The **Greedy** column denotes the planned result of the total 3D-Vias number when we use the greedy method to plan modules and I/Os. In **After Global** column, we use the proposed global planning method to assign modules and I/Os. The final planning result is shown in **After CPIR** column. The methodology of the proposed stacking IC reduces the number of 3D-Vias by 69.97% on average.

Table 2. The experimental result of our I/O planning methods targeting stacking architecture with 3, 4, 5 and 8 tiers. The greedy method only considers the I/O balance factor.

GSRC Benchmark	3 Tiers			4 Tiers			5 Tiers			8 Tiers		
	Greedy	After Global	After CPIR	Greedy	After Global	After CPIR	Greedy	After Global	After CPIR	Greedy	After Global	After CPIR
n30	155	145	142	233	202	176	306	285	242	532	455	350
n50	310	298	285	443	408	372	553	532	505	860	750	651
n100	626	538	384	929	814	560	1133	825	719	1922	1647	1312
n200	1206	1023	870	1729	1592	1284	2218	1998	1671	3628	3086	2718
n300	1678	1645	1155	2368	2041	1599	3004	2667	2140	4966	4317	2736
Avg.	100%	91.78%	71.34%	100%	88.68%	69.99%	100%	87.68%	73.36%	100%	86.11%	65.22%

We use B*-tree [3] representation to implement an aspect-ratio-driven and an area-driven floorplan. The area for each tier is considered in the area-driven floorplanning.

In the aspect-ratio-driven floorplan, we add one aspect parameter ψ to control the aspect-ratio of the floorplan. The computation of the aspect-ratio is shown as follow,

$$\max \left(\frac{\text{Tier_Length}_i}{\text{Tier_Width}_i}, \frac{\text{Tier_Width}_i}{\text{Tier_Length}_i} \right), 1 \leq i \leq T. \quad (2)$$

To show the importance of the aspect-ratio of the floorplan in stacking IC, the partition results of n300 are utilized. The ψ is set to 1.25, in other words, the aspect ratio of the resultant floorplan should be smaller than 1.25. The experimental result is shown in Fig. 12. We show the total area and the empty space ratio in this figure. The computation of total area is shown in Eq. (1). The computation of the dead space ratio for one tier (DO) is shown as follows,

	aspect ratio	Area (μm^2)			
	n300	2 Tiers	3 Tiers	4 Tiers	5 Tiers
Aspect_ratio_driven	313110 (1 : 1.24)	378672 (1 : 1.07)	341460 (1 : 1.16)	337480 (1 : 1.21)	
($\psi=1.25$) Dead space ratio for one tier	Tier 1:12.16% Tier 2:12.70% (12.75%)	1:9.01% 2:11.89% 3:3.91% (27.86%)	1:7.25% 2:7.41% 3:6.33% 4:9.15% (19.99%)	1:6.00% 2:3.84% 3:5.81% 4:6.45% 5:6.31% (19.05%)	
Dead space ratio for total area					
Area_driven	326808 (1 : 3.92)	415695 (1 : 2.06)	401292 (1 : 2.21)	340810 (1 : 2.27)	
Dead space ratio for one tier	Tier 1:4.79% Tier 2:4.43% (16.41%)	1:3.20% 2:3.88% 3:3.49% (34.28%)	1:3.02% 2:2.84% 3:3.99% 4:3.76% (31.92%)	1:3.50% 2:3.26% 3:3.22% 4:4.20% 5:3.95% (33.18%)	

Fig. 12. The experimental result of the floorplan. ψ denotes the upper bounds of the aspect ratio. In the aspect-ratio-driven floorplan, L/W or W/L should be smaller than ψ . The results of the aspect-ratio-driven floorplan and the area-driven floorplan are compared. The area of the aspect-ratio-driven floorplan is smaller than the area-driven floorplan in all cases.

$$\begin{aligned}
 Do_i &= \text{empty_area}_i / \text{tier_area}_i, 1 \leq i \leq T, \\
 \text{empty_area}_i &= L_i \times W_i - \Sigma(\text{Module_area}_i), \\
 \text{tier_area}_i &= L_i \times W_i,
 \end{aligned} \tag{3}$$

where Module_area_i denotes the total module area at Tier i , L_i denotes the longest length at Tier i , and W_i denotes the longest width at Tier i . The computation of empty space ratio for total area (ET) is shown as follows,

$$\begin{aligned}
 \text{ET} &= \frac{\Sigma(\text{empty_space}_i)}{\text{Eq. (1)}}, 1 \leq i \leq T, \\
 \text{empty_space}_i &= L \times W - \Sigma \text{Module_area}_i, \\
 L &= \max(\text{Tier_Leight}_i, 1 \leq i \leq T), \\
 W &= \max(\text{Tier_Width}_i, 1 \leq i \leq T).
 \end{aligned} \tag{4}$$

In Fig. 12, the empty space ratio of the aspect-ratio-driven floorplan is larger than that of the area-driven floorplan when we respectively compare the area for each tier. The aspect-ratio-driven floorplan has better results when compared to the final area and the empty space ratio for the total area. The results prove that the aspect ratio is an important factor in alternative stacking ICs.

5. CONCLUSION AND FUTURE WORK

This paper proposes a practical design flow in stacking ICs. The 3-D problem can be divided into several 2-D sub-problems by this design flow. Furthermore, we have proposed

an alternative stacking architecture to improve the performance of wafer stacking ICs. We also improve existing partitioning techniques to reduce the area of the alternative stacking IC and the number of 3D-Via in the early design stage. Experimental results show that the proposed method can reduce the number of 3D-Via, and the I/O number of all tiers is balanced. The modern floorplanning methodology is directly applied in our methodology, based on the system partition result. Finally, we show that the aspect ratio is an important factor when considering area cost for stacking IC.

In this paper, we have proposed a global I/O planning method at stacking IC. The tier location of I/Os are decided by our method. Further, this method can combine with a package-aware I/O planning method to optimize the package routing. This co-design method not only saves considerable run-time, but also provides better timing, area and quality of results in stacking ICs.

REFERENCES

1. J. Burns, B. Aull, C. Chen, C. L. Chen, C. Keast, J. Knecht, V. Suntharalingam, K. Warner, P. Wyatt, and D. R. Yost, "A wafer-scale 3-D circuit integration technology," *IEEE Transactions on Electron Devices*, Vol. 53, 2006, pp. 2507-2516.
2. J. Burns, L. McIlrath, C. Keast, C. Lewis, A. Loomis, K. Warner, and P. Wyatt, "Three-dimensional integrated circuits for low-power, high-bandwidth systems on a chip," in *Proceedings of International Solid State Circuits Conference*, 2001, pp. 268-269.
3. Y. C. Chang, Y. W. Chang, G. M. Wu, and S. W. Wu, "B*-trees: A new representation for nonslicing floorplans," in *Proceedings of IEEE/ACM Design Automation Conference*, 2000, pp. 258-263.
4. S. Das, A. Chandrakasan, and R. Reif, "Technology, performance, and computer-aided design of three-dimensional integrated circuits," in *Proceedings of IEEE/ACM International Symposium on Physical Design*, 2004, pp. 108-115.
5. S. Das, A. Fan, and K. N. Chen, "Design tools for 3-dintegrated circuits," in *Proceedings of IEEE Asia and South Pacific Design Automation Conference*, 2003, pp. 53-56.
6. S. Das, A. Fan, and K. N. Chen, "Three-dimensional integrated circuits: performance, design methodology, and cad tools," in *Proceedings of IEEE Computer Society Annual Symposium on VLSI*, 2003, pp. 13-18.
7. S. Dutt and W. Deng, "VLSI circuit partitioning by cluster-removal using iterative improvement techniques," in *Proceedings of IEEE International Conference on Computer Design*, 1996, pp. 194-200.
8. J. Feng, Y. Liu, P. Griffin, and J. Plummer, "Integration of germanium-on-insulator and silicon mosfets on a silicon substrate," *IEEE Electron Device Letters*, Vol. 27, 2006, pp. 911-913.
9. C. Fiduccia and R. Mattheyses, "A linear time heuristic for improving network partitions," in *Proceedings of IEEE/ACM Design Automation Conference*, 1982, pp. 175-181.
10. P. D. Franzon, W. Davis, M. Steer, H. Hao, S. Lipa, S. Luniya, C. Mineo, J. Oh, A. Sule, and T. Thorolfsson, "Design for 3D integration and applications," in *Proceed-*

- ings of International Symposium on Signals, Systems and Electronics*, 2007, pp. 263-266.
11. T. Fukushima, Y. Yamada, H. Kikuchi, and M. Koyanagi, "New three-dimensional integration technology using self-assembly technique," in *Proceedings of IEEE International Electron Devices Meeting*, 2005, pp. 348-351
 12. R. Hentschke, G. Flach, F. Pinto, and R. Reis, "An algorithm for I/O partitioning targeting 3D circuits and its impact on 3D-vias," in *Proceedings of IEEE International Conference on Very Large Scale Integration*, 2006, pp. 128-133.
 13. G. Karypis, R. Aggarwal, V. Kumar, and S. Shekhar, "Multilevel hypergraph partitioning: Application in VLSI domain," *IEEE/ACM Design Automation Conference*, 1997, pp. 526-529.
 14. B. Kernighan and S. Lin, "An efficient heuristic procedure to partition graphs," *Bell System Technical Journal*, Vol. 49, 1970, pp. 291-307.
 15. S. Kirkpatrick, C. Gelatt, and M. Vecchi, "Optimization by simulated annealing," *Science*, Vol. 220, 1983, pp. 671-680.
 16. A. Klumpp, R. Merkel, R. Wieland, and P. Ramm, "Chip-to-wafer stacking technology for 3D system integration," in *Proceedings of IEEE Electronic Components and Technology Conference*, 2003, pp. 1080-1083.
 17. S. Lim, "Physical design for 3D system on package," *IEEE Design and Test of Computer*, Vol. 22, 2005, pp. 532-539.
 18. W. C. Lo, Y. H. Chen, J. D. Ko, T. Y. Kuo, Y. C. Shih, and S. T. Lu, "An innovative chip-to-wafer and wafer-to-wafer stacking," in *Proceedings of IEEE Electronic Components and Technology*, 2006, pp. 409-414.
 19. R. Tummala, "Packaging: past, present and future," in *Proceedings of the 6th IEEE International Conference on Electronic Packaging Technology*, 2005, pp. 3-7.
 20. J. Minz and S. Lim, "Module-level 3-D global routing with an application to 3-D packaging," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2006, pp. 2248-2257.
 21. J. M. Minz, E. Wong, M. Pathak, and S. K. Lim, "Placement and routing for 3-D system-on-package designs," *IEEE Transactions on Components and Packaging Technology*, 2006, pp. 644-657.
 22. S. Souri, K. Banerjee, A. Mehrotra, and K. Saraswat, "Multiple SI layer ICS: Motivation, performance analysis, and design implications," *IEEE/ACM Design Automation Conference*, 2006, pp. 213-220.
 23. K. Takahashi and M. Sekiguchi, "Through silicon via and 3-D wafer/chip stacking technology," in *Proceedings of IEEE VLSI Circuits Digest of Technical*, 2006, pp. 89-92.
 24. P. D. Franzon, W. R. Davis, M. B. Steer, S. Lipa, E. C. Oh, T. Thorolfsson, S. Melamed, S. Luniya, T. Doxsee, S. Berkeley, B. Shani, and K. Obermiller "Design and CAD for 3D integrated circuits," in *Proceedings of the 45th IEEE/ACM Annual Design Automation Conference*, 2008, pp. 668-673.
 25. C. H. Lu, H. M. Chen, C. N. Liu, and W. Y. Shih "An I/O planning method for three-dimensional integrated circuits," in *Proceedings of Workshop on Synthesis and System Integration of Mixed Information Technologies*, 2007, pp. 198-202.



Chao-Hung Lu (呂昭宏) received the B.S. degree in Minghsin University of Science and Technology, Taiwan, R.O.C. and the M.S. degree in Chung Hua University, Taiwan, R.O.C. He is currently a candidate for doctor's degree in the Department of Electrical Engineering at National Central University. His research interests include physical design.



Hung-Ming Chen (陳宏明) received the B.S. degree in Computer Science and Information Engineering from the National Chiao Tung University, Hsinchu, Taiwan, R.O.C., in 1993 and the M.S. and the Ph.D. degrees in Computer Sciences from the University of Texas, Austin, in 1998 and 2003, respectively. He is currently an Associate Professor of Electronics Engineering with the National Chiao Tung University. His research interests include very large scale integration of computer-aided design, especially on physical design and system-on-a-chip methodology, design and analysis of algorithms, and combinatorial optimizations.



Chien-Nan Jimmy Liu (劉建男) received the B.S. and Ph.D. degrees in Electronics Engineering from National Chiao Tung University, Taiwan, R.O.C. He is currently an Associate Professor in the Department of Electrical Engineering at National Central University. His research interests include functional verification for HDL designs, high-level power modeling, and analog behavioral models for system verification. He is a member of Phi Tau Phi.