# 3D Placement with D2D Vertical Connections

Kai-Shun Hu, I-Jye Lin, Yu-Hui Huang, Hao-Yu Chi, and Yi-Hsuan Wu

Synopsys, Inc.

## 0. Revise History

Feb/13 – first version
Apr/20 – update the input/output file keyword sequence in yellow

## 1. Introduction

In the chiplet era, the trend is to split a large die into two or more small dies. By having small chiplet dies with die-to-die vertical connections, we expect to get 1) better yield, 2) better timing (if z-cost can be ignored), and 3) better cost (each die can be fabricated by different technologies).

Fig. 1 showed an example of partitioning a netlist into 2 dies. The interconnections between 2 dies need to be physically connected to complete the logic of the original netlist.

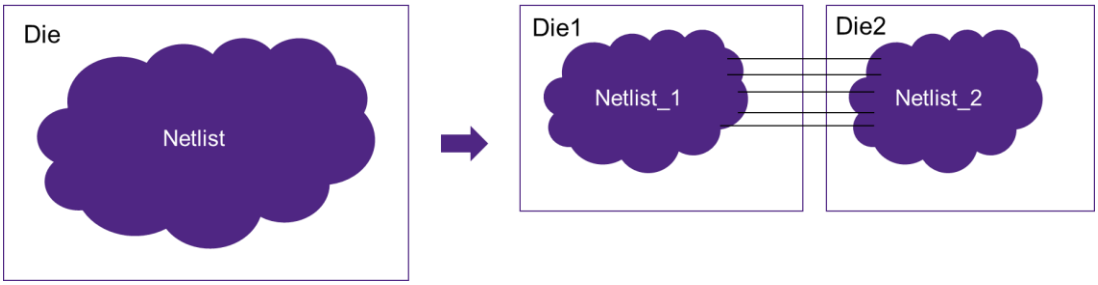



Fig. 1, netlist partitioning

In this contest, we will focus on 2-die face-to-face vertically stacked configuration with std cells only designs. The 2 dies will have the same die size. As it is face-to-face connected, the inter-die connection terminals will be on the top-most layer for both dies. Thus, the inter-die connections would not occupy any placement resource on both dies. In the fabrication, the inter-die connections can be realized with hybrid bonding technology.

Fig. 2 showed an example of 2-die face-to-face vertically connected with hybrid bonding technology. The top die will be flipped to be face down connected to the bottom die in the fabrication process.

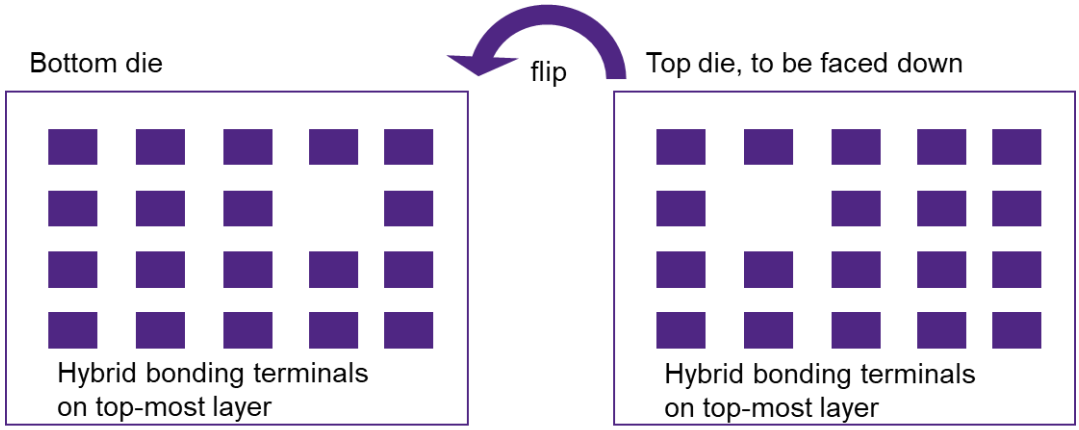



Fig. 2, face-to-face vertically connected with hybrid bonding terminals

As the pitch of hybrid bonding terminals is much larger than the std cell size, the total number of hybrid bonding terminals is limited. That is because all the hybrid bonding terminals need to be placed on the top-most layer of the die with required terminal size and terminal spacing requirement.

The 2 or more dies can be implemented with different technologies to achieve cost reduction. With different technologies, the cell characteristic, cell height, cell size, cell pin locations would be different.

In this contest, we will focus on 2-die face-to-face configuration with either the same or different technology process on 2 dies to achieve following:

1. Partition the given netlist into 2 dies with the given placement utilization constraints of 2 dies.
2. The inter-die connections need to have 1 and only 1 hybrid bonding terminal on the top-most layer of the 2 dies that satisfied the given terminal size and spacing constraint.
3. After partitioning the given netlist into 2 dies, contestants need to place and legalize both the top die and bottom die. The technology process of the 2 dies will be given. If the 2 dies are given with different technologies, both given technologies would have the same logical library cells by library cell name matching. However, the cell size, cell height, cell pin location of the same logical library cell would be different in each technology.
4. Goal is to optimize the total Half-Perimeter Wirelength (HPWL) of the 2 placed dies. The center point of the hybrid bonding terminals needs to be included in the HPWL calculation for each die.

## 2. Contest Objective

To simplify the problem, this contest would focus on 2-die face-to-face configuration with std cells only.

The contestants need to develop a 3D placer engine to place and optimize the 2 dies std cell placement and the hybrid bonding terminal placement to minimize the total HPWL of the 2 dies. The center point of the hybrid bonding terminals needs to be included in the HPWL calculation. All the cells need to be placed on row and cannot have overlap with each other. All the cells would be single row height of the corresponding technology of the die. Also, all the given constraints need to be satisfied.

## 3. Problem formulation, Input/Output Format

Given:

1. Netlist
2. Die size for both bottom die and top die
   - It would be the same size for both dies
3. Max placement utilization ratio for top die
4. Max placement utilization ratio for bottom die
5. Placement rows of top die
6. Placement rows of bottom die
7. Std cell library for each die. If the 2 dies are with different technologies, the given std cell library for each die would have equivalent logic
8. Hybrid bonding terminal size & the required spacing between 2 terminals and between terminal and die boundary

Output:

1. Top die placement result
2. Bottom die placement result
3. Hybrid bonding terminal placement result with net information

   All the output, including placement result and hybrid bonding terminal locations are with the resolution of integer.

For the output, please use the same coordinate system for both top die and bottom die. That means the bottom-left point of top die and bottom die are both at (0, 0). In the output file, the same (x, y) coordinate on 2 dies would mean the same physical location on the surface. So, the same (x, y) coordinate of the hybrid bonding terminal on top die and bottom die would means the hybrid bonding terminal would touch each other to complete the physical connection.

The fabrication process can be illustrated as in Fig 3. We firstly use the same coordinate system to generate the output. Then, the layout of the top die would be mirrored by Y-axis for getting the finalized layout for fabrication. But, in this contest, we do NOT need to consider the Y-mirrored process for top die. What needed to be outputted in this contest is (3.A).

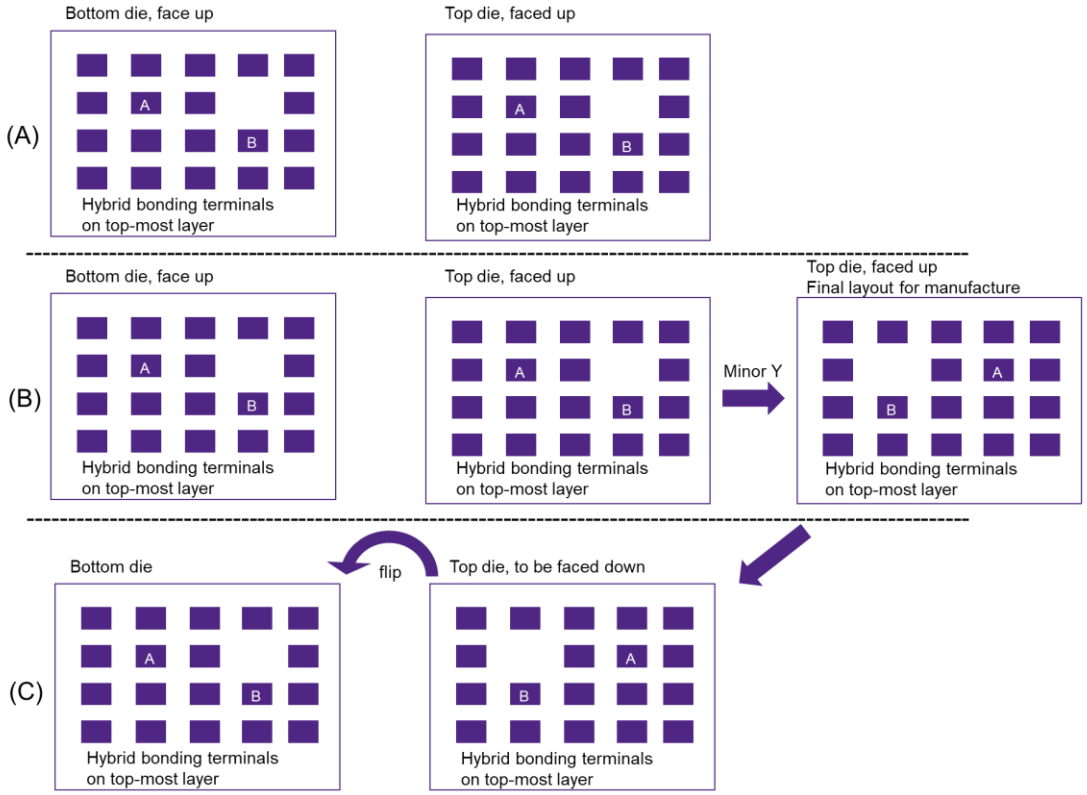



Fig. 3, three steps of preparing the final layout for fabrication. (3.A) the output for this contest. (3.B) the layout of top die would be mirrored by Y-axis. (3.C) would ship the top die Y-mirrored layout for fabrication.

### 3.1 Format of Input data

Netlist

*Syntax*

```
NumInstances <instanceCount>
Inst <instName> <libCellName>
NumNets <netCount>
Net <netName> <numPins>
Pin <instName>/<libPinName>
```

*Example*

```
NumInstances 2
Inst C1 MC1
Inst C2 MC3
NumNets 1
Net N1 2
Pin C1/P2
Pin C2/P1
```

Die size for both the bottom die and top die

*Syntax*

```
DieSize <lowerLeftX> <lowerLeftY> <upperRightX> <upperRightY>
```

*Example*

```
DieSize 0 0 500 450
```

Max placement utilization ratio for top die & the ratio for bottom die. The value is the max utilization percentage.

*Syntax*

```
TopDieMaxUtil <util>
BottomDieMaxUtil <util>
```

*Example*

```
TopDieMaxUtil 75
BottomDieMaxUtil 80
```

The formulation of utilization ratio of a die =

$$\frac{\sum(area\ of\ each\ cells\ on\ the\ die)}{Area\ of\ the\ die}$$

Placement rows of top die & of bottom die. The given rows would start from (0, 0) and cover the entire die.

*Syntax*

```
TopDieRows <startX> <startY> <rowLength> <rowHeight> <repeatCount>
BottomDieRows <startX> <startY> <rowLength> <rowHeight> <repeatCount>
```

*Example*

```
TopDieRows 0 0 500 10 45
BottomDieRows 0 0 500 15 30
```

Technology and corresponding std cell library. All the cells would be single row-height of the corresponding technology.

*Syntax*

```
NumTechnologies <technologyCount>
Tech <techName> <libCellCount>
LibCell <libCellName> <libCellSizeX> <libCellSizeY> <pinCount>
Pin <pinName> <pinLocationX> <pinLocationY>
```

*Example*

```
NumTechnologies 2
Tech TA 2
LibCell MC1 5 10 1
Pin P1 2 7
LibCell MC2 7 10 2
Pin P1 5 3
Pin P2 3 6
Tech TB 2
LibCell MC1 5 15 1
Pin P1 2 11
LibCell MC2 7 15 2
Pin P1 2 12
Pin P2 3 3
```

The technology for top die and bottom die.

*Syntax*

```
TopDieTech <TechName>
BottomDieTech <TechName>
```

*Example*

```
TopDieTech TA
BottomDieTech TB
```

Hybrid bonding terminal size & required spacing between 2 terminals and between terminal and die boundary

*Syntax*

```
TerminalSize <sizeX> <sizeY>
TerminalSpacing <spacing>
```

*Example*

```
TerminalSize 20 20
TerminalSpacing 15
```

Input file would always be with this following keyword sequence.

```
NumTechnologies
...
DieSize

TopDieMaxUtil
BottomDieMaxUtil

TopDieRows
BottomDieRows

TopDieTech
BottomDieTech

TerminalSize
TerminalSpacing

NumInstances
...
NumNets
...
```

### 3.2 format of Output data

Top die placement result. The outputted coordinate is lower-left coordinate of the cell. All the cells can NOT be flipped nor mirrored.

*Syntax*

```
TopDiePlacement <InstCount>
Inst <instName> <locationX> <locationY>
```

*Example*

```
TopDiePlacement 2
Inst C1 0 10
Inst C4 15 20
```

Bottom die placement result. The outputted coordinate is lower-left coordinate of the cell. All the cells can NOT be flipped nor mirrored.

*Syntax*

```
BottomDiePlacement <InstCount>
Inst <instName> <locationX> <locationY>
```

*Example*

```
BottomDiePlacement 3
Inst C2 20 15
Inst C3 23 30
Inst C5 50 15
```

Hybrid bonding terminal placement result with net information. The outputted terminal (x, y) location is the center of the hybrid bonding terminal.

*Syntax*

```
NumTerminals <TerminalCount>
Terminal <netName> <locationX> <locationY>
```

*Example*

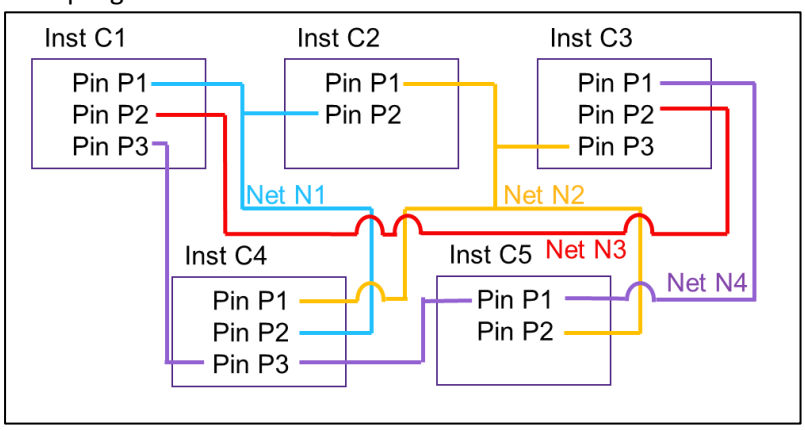
```
NumTerminals 2
Terminal N1 100 200
Terminal N3 180 180
```

Output file needs to be with this following keyword sequence.

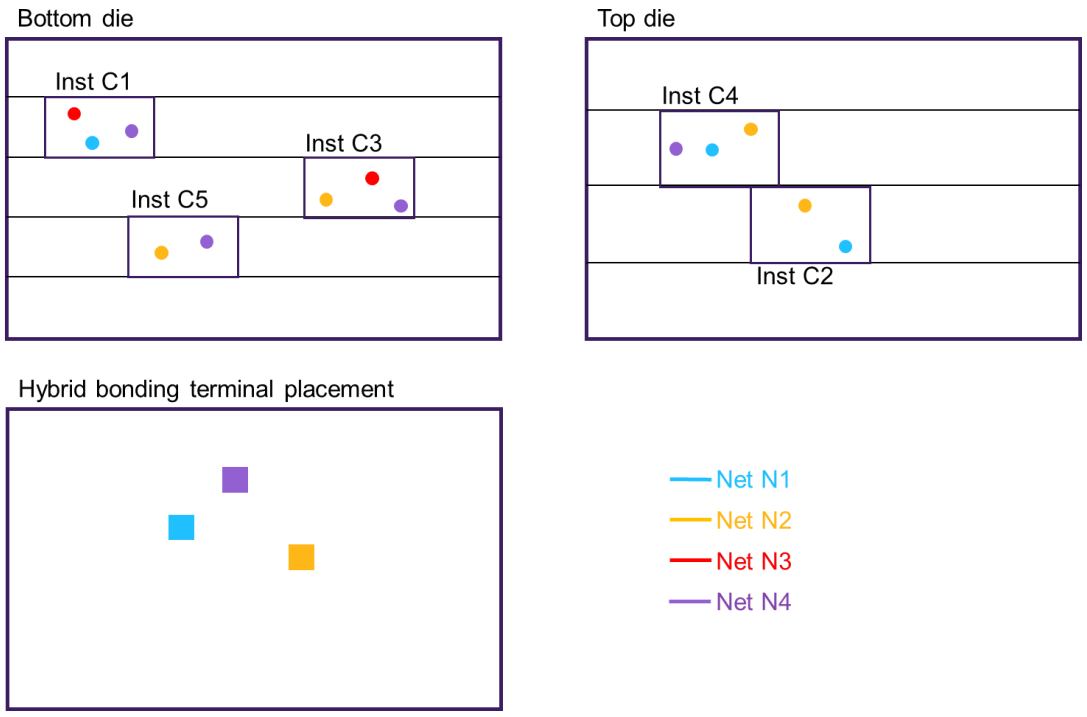
```
TopDiePlacement
...
BottomDiePlacement
...
NumTerminals
...
```

## 4. Example

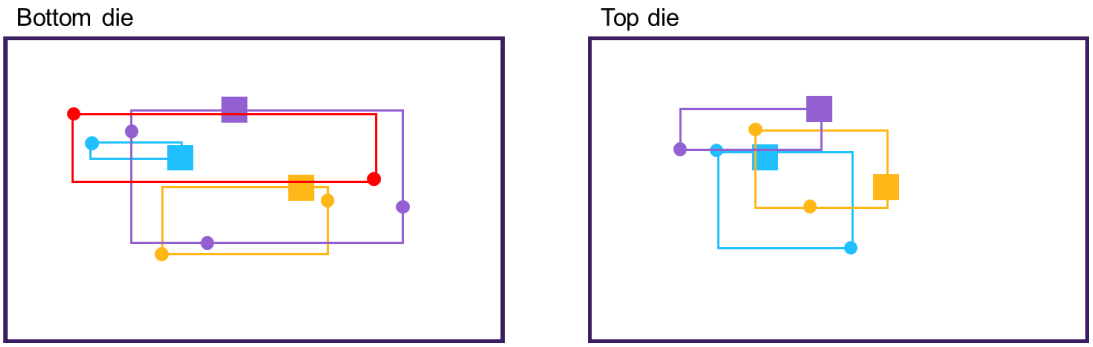
Example given netlist:



Example output:




HPWL result of the output example:




## 5. Evaluation

- Max placement utilization constraint must be satisfied
- All the given instances must be placed on either top die or bottom die
- All the instances must be on row without overlap
- Hybrid bonding terminal spacing constraint must be satisfied
- Crossing-die nets must have 1 and only 1 hybrid bonding terminal
- The minimum resolution of all the coordinate values is integer
- Runtime limit is 1hr for each case in the evaluation machine. The hidden cases will be in the same scale as public cases.

If the program and the output data violate any of these above bullets, you will get 0 score for the corresponding test case.

Evaluation score = HPWL of top die + HPWL of bottom die

### 5.1 Program requirements

Your program should be able execute like following:

```
./$binary_name <input.txt> <output.txt>
```

The number of CPU cores available for your program is 8 cores in the evaluation.

### 5.2 Evaluator

There will be an Evaluator provided in the contest website. Contestants can use this provided Evaluator to validate file format of the outputted file and also the correctness of the result.

```
./evaluator <input.txt> <output.txt>
```

## 6. References


T. Thorolfsson, G. Luo, J. Cong and P. D. Franzon, "Logic-on-logic 3D integration and placement," 2010 IEEE International 3D Systems Integration Conference (3DIC), 2010, pp. 1-4, doi: 10.1109/3DIC.2010.5751451.

J. Kim et al., "RTL-to-GDS Design Tools for Monolithic 3D ICs," 2020 IEEE/ACM International Conference On Computer Aided Design (ICCAD), 2020, pp. 1-8.

S. S. Kiran Pentapati, K. Chang, V. Gerousis, R. Sengupta and S. K. Lim, "Pin-3D: A Physical Synthesis and Post-Layout Optimization Flow for Heterogeneous Monolithic 3D ICs," 2020 IEEE/ACM International Conference On Computer Aided Design (ICCAD), 2020, pp. 1-9.

B. W. Ku, K. Chang and S. K. Lim, "Compact-2D: A Physical Design Methodology to Build Two-Tier Gate-Level 3-D ICs," in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 39, no. 6, pp. 1151-1164, June 2020, doi: 10.1109/TCAD.2019.2952542.