Cost-driven 3D Design Optimization with Metal Layer Reduction Technique

Qiaosha Zou *, Jing Xie *, Yuan Xie *†
* The Pennsylvania State University, USA
† AMD Research, Advanced Micro Devices, Inc., USA
Email: *{qszou, jzx105}@cse.psu.edu, † yuan.xie@amd.com

Abstract—Three-dimensional integrated circuit (3D IC) is a promising solution to continue the performance scaling. However, the fabrication cost for 3D ICs can be a major concern for the adoption of this emerging technology. In this paper, we study the cost implication for both TSV-based and interposer-based 3D ICs, with the observation that many long metal interconnects in 2D designs are replaced by TSVs in 3D designs, and therefore the number of metal layers to satisfy routing requirements can be reduced, resulting in cost saving in 3D ICs. Based on our cost model, we propose a cost-driven 3D design space optimization flow that balances the design area and metal layer requirement, by optimizing the cost tradeoffs between silicon area and the number of metal layers. With the cost-driven design optimization flow, we can achieve cost saving up to 19% for TSV-based 3D designs, and 26% for interposer-based 3D designs, respectively, compared to the baseline designs.

I. INTRODUCTION

Three-dimensional integrated circuit (3D IC) is a promising technology that benefits from both integration density and performance. By integrating a high-bandwidth low-latency vertical interconnects Through-Silicon Vias (TSVs), 3D integrations can provide: (i) smaller footprint and higher package density; (ii) higher performance and lower interconnect power consumption; (iii) heterogeneous chip stacking [1]–[3].

The high fabrication cost of 3D stacking is a major concern for adopting this emerging technology. Different from 2D designs, 3D designs need wafer thinning and wafer bonding, implying higher process complexity and lower fabrication yield. Previous study shows that, a 2 layer 3D design is more cost efficient than the corresponding 2D design only when the design gate count is more than 143 million for 45nm process technology, making 3D integration not suitable for small designs [4]. Providing a cost-efficient design is an urgent requirement for 3D technology.

Mask cost is one important component in fabrication cost. Contemporary complex circuits require more metal layers, making the mask cost continuously increases. For example, at 32nm technology node, the number of process steps is 462 with mask cost of 2.974M/set for 9 metal layers; when the number of metal layers increases to 11, the process steps and mask cost increase to 498 and 3.212M/set, respectively [5]. Besides mask cost for metal layers, chip area is another determinant of fabrication cost due to its direct impact on yield. Reducing metal layers can cut down the process steps and mask cost; however, it will cause dramatic increment in chip size in 2D designs. By replacing long interconnects with TSVs in 3D designs, metal layers can be reduced while small chip size is maintained as shown in [4].

In this work, we implemented a block granularity 3D design optimization flow to generate cost-efficient designs by balancing the chip size (placement density) and routability (metal layers). At physical design stage, CAD tools require designers to input the area utilization for placement density. Thus, the design quality highly depends on the designers’ experience and circuit complexity. In this flow, optimal placement density and metal layer reduction are considered concurrently for cost-efficient designs. Both TSV-based 3D and interposer-based 3D designs are included in our work.

The rest of the paper is organized as follows: the background of TSV and interposer-based 3D ICs are introduced in Section II; Section III illustrates the interconnect model for routability and the 3D design cost model used in this work; the cost-driven 3D design optimization flow is demonstrated in Section IV; the experimental setup and results are shown in Section V following by conclusion in Section VI.

II. BACKGROUND

With continuously technology scaling, the interconnect performance degrades due to the increasing fraction of interconnect delay and power consumption. 3D stacking provides a promising solution to tackle the interconnect bottleneck by providing vertical connections and smaller chip footprint. In terms of stacking components, 3D ICs can be categorized into TSV-based and interposer-based designs.

A. TSV-based 3D ICs

With the capability of vertical connections, multiple dies can stack in TSV-based 3D design, as shown in Figure 1. The stacking method between two dies can be either face-to-back or face-to-face. Figure 1(a) shows an example of face-to-back stacking, where TSVs are built in the silicon substrate and carry the signal from top tier to bottom tier. In face-to-face stacking as shown in Figure 1(b), micro-bumps directly connect the top metal layers of two tiers without TSVs going through the substrate. By influencing the overall stacking yield and adding bonding cost, TSV-based 3D designs have great impact on the fabrication cost.
we describe how to estimate the die area and minimum metal cost analysis. In this section, the related interconnect model models for TSV and interposer-based 3D ICs are necessary for requirement that can guarantee the routability given the design in 3D designs, we need to estimate the minimum metal layer in our proposed framework. In order to reduce metal layer are silicon device with pre-built TSVs: (a) is stacking chips on one side of the interposer, forming two tiers stacking with one interposer, as shown in Figure 2(a). The TSVs act as interconnects for dies on the same tier and inter-tier communication. In this case, single interposer size is bounded by the maximum tier size.

In our cost-efficient design flow, we only consider the two sides stacking scenario with one interposer layer. This stacking method has higher interposer utilization rate and the interconnect behavior between tiers is similar to TSV-based 3D designs.

III. INTERCONNECT AND COST MODELS

Metal layers can be reduced to achieve cost-efficient design in our proposed framework. In order to reduce metal layer in 3D designs, we need to estimate the minimum metal layer requirement that can guarantee the routability given the design gate count. Besides the interconnect model, 3D system cost models for TSV and interposer-based 3D ICs are necessary for cost analysis. In this section, the related interconnect model and 3D cost model are introduced. In the interconnect model, we describe how to estimate the die area and minimum metal layers for feasible routing, when only limited information are available. The second part of this section introduces a comprehensive 3D cost model that considers the area overhead and yield impact of TSV bonding, and the fabrication overhead of interposer stacking. These models facilitate the cost-driven 3D design flow in Section IV.

A. Interconnect Model

The die area, which is defined as the area occupied by the transistors and the interconnects, is closely related to the gate count in designs [11]. However, in physical circuit designs, gates are not placed tightly with other, which introduces an area utilization rate to indicate the placement density. So the die area can be estimated as a function of the gate counts and area utilization rate:

\[
A_{\text{die}} = \frac{N_g A_g}{A_{\text{util}}} \tag{1}
\]

where \(N_g\) is the number of gates in the design, \(A_g\) is an empirical parameter that represents the relation between single gate area and feature size, \(A_{\text{util}}\) is the area utilization rate. In our work, \(A_g\) is assumed to be 3125\(\lambda^2\), and \(\lambda\) is half of the feature size.

The required metal layers for feasible routing highly depends on the placement and routing methodology, however, the detailed information are unavailable at the early design stage. In this model, we mainly analyze the relation between routing demand and routing resource to determine the required metal layers. A wire length distribution model based on Rent’s rule is used to perform the routing demand estimation [12]. The wire length distribution equations with respect to the interconnect segment length \(l\) is as follows:

**Region I:** \(1 \leq l \leq \sqrt{N_g}\)

\[
i(l) = \frac{\alpha k}{2} \Gamma \left( \frac{3}{2} - 2\sqrt{N_g}l^2 + 2N_g l \right) l^{2p-4} \tag{2}
\]

**Region II:** \(\sqrt{N_g} \leq l < 2\sqrt{N_g}\)

\[
i(l) = \frac{\alpha k}{6} \Gamma \left( 2\sqrt{N_g} - l \right)^3 l^{2p-4} \tag{3}
\]

where \(k\) is Rent’s coefficient, represents the average number of pins per block and \(p\) is Rent’s exponent [13] within the range of 0.4 to 0.9 [14]. \(\alpha = \frac{f_o}{0.1}\) is related to the average fanout of a gate to represent the proportion of on-chip sink terminals. \(\Gamma\) is given as follows:

\[
\Gamma = \frac{2N_g \left( 1 - N_g^{-p-1} \right)}{\left( N_g^{p+1} \frac{1+2p-2p^{-1}}{p(2p-1)(2p-3)} - \frac{1}{6p} + 2\sqrt{\frac{N_g}{2p-1}} - \frac{N_g}{p-1} \right)} \tag{4}
\]

For any given interconnect length, the accumulative number of interconnects \(I(l)\), which means the number of interconnects with length smaller or equal to \(l\), can be derived from the cumulative integral of the wire length distribution function \(i(l)\). The accumulative routed wire length \(L(l)\), which denotes the routing demand, is given as the first-order moment of \(i(l)\).
To this end, one of the two components in interconnect model, the routing resource, has been analyzed. The routing resource is related to the available routing area and wire pitches. The available signal routing resource is significantly smaller than the total track length on all metal layers because of the routing efficiency, the impact of the vias, and the resource occupied by power, ground, and clock distribution [15]. The available signal routing resource for each metal layer is given as follows:

\[
K_i = \frac{A_{\text{die}} \eta_i - 2A_v (I(l_{\text{max}}) - I(l_i))}{\omega_i}
\]  

(5)

where \( A_{\text{die}} \) is the die area given in equation 1, \( \eta_i \) is the metal layer utilization, \( A_v \) and \( \omega_i \) are the via area and wire pitches on metal layer \( i \), \( l_{\text{max}} \) and \( l_i \) are the maximum single wire length on the whole chip and one metal layer.

With this interconnect model, we are assuming the shorter interconnects are routed first on lower metal layers. The routing process starts from bottom metal layer, and moves up the higher metal layers only when the bottom layers are fully utilized. Furthermore, for each metal layer, the total routed wire length cannot exceed the available routing resource. The required metal layers are derived from repeating this process until all the wires are properly assigned to metal layers.

### B. 3D Cost Model

The cost models are different in TSV-based and interposer-based designs. The cost of 3D TSV-based ICs contains two parts: wafer cost and bonding cost. Wafer cost captures the wafer cost with related die cost and die yield, similar to the cost in traditional 2D designs. Bonding cost models the cost overhead of TSV bonding, including wafer thinning, TSV forming, and die bonding, which is unique in 3D designs. In our work, the interposer is treated as regular silicon device, therefore, only the wafer cost is considered.

**Wafer Cost Model.** In the wafer cost part, the most important factor is the die area. The die area estimations of TSV-based and interposer-based structure are different. TSV-based structure introduces area overhead, because the silicon area where TSVs are built cannot be utilized. Normally, the diameter of TSVs is large compared to the transistor feature size and it can range from 1\( \mu \)m to 10\( \mu \)m [16]. The area overhead can be estimated through the equation \( A_{3D} = A_{\text{die}} + N_{TSV/die} \times A_{TSV} \) [4]. \( N_{TSV/die} \) refers to the number of TSVs on each die, \( A_{TSV} \) is the area of TSV and it can be calculated from the TSV pitch, and \( A_{3D} \) is the final die area of one 3D stacking die.

For interposer-based structure, the vertical interconnects have no impact on chip area because the TSVs are fabricated inside the interposer. So the die area can be estimated directly from Equation 1. Given the die area and wafer diameter, the wafer utilization in terms of number of dies per wafer can be calculated.

Besides the wafer utilization, the die yield also influences the cost, which is formulated as follows:

\[
Y_{\text{die}} = Y_{\text{wafer}} \times \frac{1 - e^{-2A_{\text{die}}D_0}}{2A_{\text{die}}D_0}
\]

(6)

where \( D_0 \) is the defect density of the wafer, \( Y_{\text{die}} \) and \( Y_{\text{wafer}} \) are the die yield and wafer yield, respectively.

Since 3D integration enables heterogeneous stacking, the chip size of stacking dies can be different. The overall die yield can be calculated by multiplying the individual yield of each die in the stacking.

**3D Bonding Cost.** In order to build 3D stacking, extra fabrication steps are needed, including wafer thinning, TSV forming, and die bonding. These process steps have impact on bonding cost: wafer thinning decreases the wafer yield; TSV forming increases the process cost; and die bonding influences both the process cost and stacking yield. The total stacking yield is a function of the TSV number \( (N_{TSV}) \) and the single TSV yield \( (Y_{TSV}) \), and it can be calculated by

\[
Y_{S} = Y_{\text{bonding}} \times Y_{TSV}^N \quad [17].
\]

The TSV number can be estimated as follows at the early design stage:

\[
N_{TSV} = \alpha k_1(N_1 + N_2)(1 - (N_1 + N_2)^{p_{1,2}} - 1)
\]

(7)

\[
= -\alpha k_1(N_1 + N_2)(1 - N_1^{p_{1,2}}) - \alpha k_2(N_1 + N_2)(1 - N_2^{p_{2,1}})
\]

where \( N_1 \) and \( N_2 \) are the number of blocks in two tiers, \( k_{1,2} \) and \( p_{1,2} \) are the equivalent Rent’s coefficient and exponent.

**Overall 3D Cost Model.** The TSV-based cost combines the wafer cost and the bonding cost. In our work, we only consider die-to-wafer stacking, the TSV-based 3D cost \( (C_{T3D}) \) can be calculated from:

\[
C_{T3D} = \sum_{i=1}^{N} \left( \frac{C_{\text{die},i} + C_{KGDtest}}{Y_{\text{die},i}} \right) + (N - 1)C_{\text{bonding}}
\]

(8)

where \( C_{\text{bonding}} \) is the bonding cost, which captures the wafer thinning, TSV forming, and bonding cost.

The interposer-based structure cost only contains the wafer cost model. And we only consider the case shown in Figure 2(b) where two known good dies stack on both sides of an interposer. In this scenario, the overall cost of one interposer stacking \( (C_{I3D}) \) is given as follows:

\[
C_{I3D} = \sum_{i=1}^{N} (C_{\text{die},i} + C_{KGDtest}) + \sum_{i=1}^{N/2} C_{\text{interposer}}
\]

(9)

where \( N \) is the number of tiers in the 3D design, \( C_{\text{die},i} \), \( C_{\text{interposer}} \) are the cost of each die in the stacking and interposer wafer cost, respectively, \( C_{KGDtest} \) is the testing cost for each die.

### IV. COST-DRIVEN 3D DESIGN OPTIMIZATION FLOW

Based on the interconnect model and 3D cost model, cost-driven design optimization flow is implemented in block granularity to reduce system cost. In this section, we introduce the proposed design flow with metal layer reduction in detail.

The flow takes the gate counts from synthesized design as input and outputs the cost-efficient 2D designs, TSV-based,
and interposer-based 3D designs. Figure 3 shows the proposed cost-driven design optimization flow. In this work, the 3D partitioning is in block granularity and each block contains an arbitrary number of gates according to the block functionality.

![Diagram](image)

Fig. 3. Cost-driven design optimization flow in 2D designs, TSV-based and interposer-based 3D designs with metal layer reduction technique.

The 2D designs, TSV-based and interposer-based 3D designs after partitioning are generated with default area utilization. The minimum required metal layers for feasible routing are calculated using the interconnect model introduced in Section III. The design costs are calculated based on the estimated chip area and metal layers.

In this flow, the important steps are area utilization estimation and corresponding metal layers estimation. The interconnect model in Section III is used for metal layers estimation. However, the $l_i$, which is the maximum interconnect segment length that can be routed on metal layer $i$, is constrained by the routing resource. The routing resource on the other hand is bounded by the via area, which depends on $l_i$. For example, on metal layer 1, if the maximum wire length that can be routed is $l_1$, then the wires that are longer than $l_1$ need to be put on higher metal layers, resulting in $I_{max} - I_1$ number of vias. The bigger $l_1$ value is, the less area occupied by vias. But the bigger $l_1$ needs more routing resource determined by available chip area. Finding the balanced $l_i$ for each metal layer is the key step in metal layer estimation. Searching all the possible combinations in sequence is extremely time consuming. In this work, we are using binary search algorithm shown in Algorithm 1. The algorithm takes the chip area and wire pitches as input and gives the metal layers as output.

Area utilization is the major factor that influences the fabrication cost. First, die yield is exponentially proportional to the die area which is calculated from area utilization. It means larger chip area results in significant higher cost. Second, the required metal layers are determined by the area utilization which determines the routing resource. Increased area utilization results in denser placement and more metal layers, implying higher mask cost. In this work, we consider the maximum area utilization that one design can achieve without sacrificing routability. Optimal area utilization denotes the maximum utilization with default metal layers and maximum area utilization is the maximum area utilization after one metal layer reduction. The design exploration flow finds the cost-efficient design through calculating the optimal and maximum area utilization. At physical design stage, CAD tools usually require designers to input the area utilization and perform placement and routing accordingly. However, sometimes the utilization is pessimistically estimated resulting in large unused chip area. So finding the best area utilization can help reduce the die area and design cost with feasible routing.

The same binary search algorithm is used for exploring the area utilization. In order to keep the design practical and reduce computation complexity, we set two utilization thresholds: the maximum and minimum thresholds, to control the searching range. The optimal area utilization is searched between the default utilization rate and maximum threshold. The maximum utilization rate without causing additional metal layers is the output. The maximum utilization rate after metal layer reduction should be between the minimum threshold and default utilization.

The metal layer and area utilization estimations are suitable for generic logic designs. However, the estimation is inaccurate for regular pattern designs, such as cache and memory. In these regular designs, the routing complexity is not related to the number of gates and the metal layers are fixed once design pattern is known.

**Algorithm 1 Outline of the binary search algorithm for the estimation of metal layer requirement**

```plaintext
Metal Layer Estimation (chip area, wire pitch)
{initialize layer count, $l_{max}$, $L(l_{max})$ and $I(l_{max})$}
layer count = 0, i = 0;
while $l_i \leq l_{max}$ do
  $l_i = (upbound - lowbound)/2$;
  repeat
    Calculate $k_i$ from $l_i$;
    Calculate $L(l_i)$ from $l_i$;
    if $L(l_i) - L(l_{i-1}) < k_i$ then
      Update the lowbound;
    else
      Update the upbound;
    end if
  until lowbound > upbound
  Finish searching on metal layer $i$; i++;
  Record $L(l_i)$ and $l_i$;
  {find the balanced $l_i$}
end while
layer count = i;
return Layer count;
```

V. Experiment Results

In this section, the results from cost-driven design optimization are shown. We implement the framework in C++ and use 45nm technology node. The gate size and wire pitches information are extracted from NanGate FreePDK45 Generic Open Cell Library [18]. The parameters used in the experiments are listed in Table I. During the design optimization process, the area utilization variation is within the range of 20% to keep the
The cost related parameters in this experiment are from IC cost model [5]. We use 300mm TSMC dual gate CMOS logic process technology for device layer and 300mm UMC interposer process technology for interposer layer. TSV-based 3D bonding method is face-to-back bonding and die-to-wafer stacking.

### A. Cost Analysis without Optimization

The costs of 2D designs, TSV-based and interposer-based designs with default area utilization and minimum required metal layers are calculated. All the 3D designs are partitioned into 2 layers, although the framework is capable to support 3D designs with more tiers. The cost comparison without optimization are shown in Figure 4.

The fabrication cost rises with increased gate count as expected. When the gate count is smaller than 100M, the 2D implementation has lower cost than corresponding 3D cases. However, as the gate count continuously increased, 3D designs show the advantages of smaller footprint and less routing complexity over 2D designs. In the proposed flow, we only consider one interposer layer design, and the interposer area is determined by the maximum tier size. When the gate count increased, the cost of interposer is higher than 3D TSV bonding due to the lower die yield and higher wafer cost of larger interposer area. In previous work [4], the gate count of 3D enabling point is less than 100M, which is smaller than our results because the previous process technology is 65nm. As transistor size continuously scaling, the chip size is reduced, and therefore amortized cost per chip is lower, making 3D stacking designs more cost-efficient for larger designs.

Besides 2 tiers partitioning, 3D TSV-based design costs with more tiers are examined. The costs are shown in Figure 5. In 3D TSV-based designs, more tiers do not necessarily lead to higher fabrication cost. From the results, for larger designs with gate count more than 150M, we can gain cost saving by partitioning the design into more tiers. It is because when the design is large, the TSV bonding cost overhead is compensated by smaller footprint since the die yield is exponentially related to the chip size.

### B. Cost Analysis with optimization

In this section, the results after cost-efficient optimization are shown. For each design (both 2D and 3D cases), the optimal utilization and maximum utilization are obtained to calculate the corresponding design costs. The results are shown in Table II with the most cost efficient utilization.

Among all the designs, the most cost-efficient designs are obtained when the area utilization rate is higher than the default utilization, implying the significant impact of chip area on fabrication cost. In 2D designs, the optimal utilization decreases when the gate counts increases. The optimal utilization is achieved when all the metal layers are fully utilized for signal routing. For small designs, the top metal layer is usually with low utilization, only a small portion of interconnects is used. The large designs with higher gate counts require more routing resources causing high utilization even in high metal layers. It makes the large designs harder to shrink the chip area. In this results, 89% chip area utilization is the optimal utilization for 5M gate count designs, but for designs with 150M gate counts, the optimal utilization decreases to 72%.

After the design optimization, the costs of 2D and 3D designs are significantly reduced. Cost savings are shown in Table III. 2D designs have the highest saving on average except the last design when the original design before optimization is already the best one. TSV and interposer based designs have significant cost saving after the optimization. For large designs, the benefits are higher because of larger flexibility in chip area.

Although chip area reduction is the major factor for cost saving, metal layer reduction also provides cost saving compared to the baseline designs. Only one metal layer is reduced in the experiment because significant area overhead is introduced after aggressive metal layer reduction resulting in high fabrication cost. The cost saving results from metal layer reduction is shown Table IV. In order to maintain feasible routing after one metal layer reduction, chip area is increased.
to provide additional routing resources. The area increment percentage depends on different designs. For example, designs with low utilized top metal layer can have metal layer reduction with very small area overhead. But for some designs, area needs to increase about 10% to enable one metal layer reduction, thus the cost saving is negative since cost overhead for larger chip area outweighs the cost saving from metal layer reduction. For 2D designs with 200M gate counts, when the area increases only 1%, the required metal layer can be reduced. Thus the cost saving for the design is high. For 3 layer 3D TSV designs with 50M gate counts, in order to gain one metal layer reduction, the area needs to increase 20%, so no cost saving can be achieved from metal layer reduction. On average, the area increment is within 10% for designs to have one metal layer reduced in 2D and 3D cases.

<table>
<thead>
<tr>
<th>Gate Count</th>
<th>2D Design</th>
<th>TSV-based 3D</th>
<th>Interposer-based 3D</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>metal layer</td>
<td>area util (%)</td>
<td>cost</td>
</tr>
<tr>
<td>5M</td>
<td>3</td>
<td>89</td>
<td>7.17</td>
</tr>
<tr>
<td>10M</td>
<td>3</td>
<td>81</td>
<td>8.18</td>
</tr>
<tr>
<td>50M</td>
<td>5</td>
<td>79</td>
<td>18.34</td>
</tr>
<tr>
<td>100M</td>
<td>6</td>
<td>78</td>
<td>38.82</td>
</tr>
<tr>
<td>150M</td>
<td>6</td>
<td>72</td>
<td>78.37</td>
</tr>
<tr>
<td>200M</td>
<td>7</td>
<td>71</td>
<td>131.85</td>
</tr>
<tr>
<td>250M</td>
<td>8</td>
<td>70</td>
<td>204.62</td>
</tr>
</tbody>
</table>

VI. CONCLUSION

3D IC is a promising solution for performance improvement. However, the fabrication cost is the major hinder for designers to adopt 3D designs. In this paper, a 3D design cost-driven optimization flow is proposed to find the cost-efficient designs in 2D and 3D cases. Both TSV and interposer-based 3D structures are considered. The cost-efficient designs are found with balanced area utilization and metal layers. The experiment results show that for both 2D and 3D cases, metal layer reduction and optimal area utilization exploration can achieve cost saving compared to baseline designs. We can achieve cost saving up to 19% for TSV-based 3D designs, and 26% for interposer-based 3D designs, respectively, compared to the baseline designs.

REFERENCES