System-Level Cost Analysis and Design Exploration for Three-Dimensional Integrated Circuits (3D ICs)

Xiangyu Dong and Yuan Xie
Computer Science and Engineering Department
Pennsylvania State University
University Park, PA 16802, USA
e-mail: {xydong,yuanxie}@cse.psu.edu

Abstract— Three-dimensional integrated circuit (3D IC) is emerging as an attractive option for overcoming the barriers in interconnect scaling. The majority of the existing 3D IC research is focused on how to take advantage of the performance, power, smaller form-factor, and heterogeneous integration benefits that offered by 3D integration. However, all such advantages ultimately have to translate into cost savings when a design strategy has to be decided: *Is 3D integration a cost effective technology for a particular IC design?* Consequently, system-level cost analysis at the early design stage is imperative to help the decision making on whether 3D integration should be adopted. In this paper, we study the design estimation method for 3D ICs at the early design stage, and propose a cost analysis model to study the cost implication for 3D ICs, and address the following cost-related problems related to 3D IC design: *(1) Do all the benefits of 3D IC design come with a much higher cost? (2) How can 3D integration be achieved in a cost-effective way? (3) Are there any design options to compensate the extra 3D bonding cost?* A cost-driven 3D IC design flow is also proposed to guide the design space exploration for 3D ICs toward a cost-effective direction.

I. Introduction

Three-dimensional integrated circuit (3D IC) [1–6] is emerging as an attractive option for overcoming the barriers in interconnect scaling, thereby offering an opportunity to continue performance improvements using CMOS technology. In a 3D IC, multiple device layers are stacked together with direct vertical interconnects through them (Figure 1 shows a conceptual 2-layer 3D IC). The direct vertical interconnects are called *Through-Silicon Vias (TSVs)*. Consequently, one of the most important benefits of a 3D chip over a traditional two-dimensional (2D) design is the reduction in global interconnects [5, 6]. Other benefits of 3D ICs include: (i) higher packing density and smaller footprint due to the addition of a third dimension; (ii) higher performance due to reduced average interconnect length and higher memory bandwidth [1]; (iii) lower interconnect power consumption due to the reduction in total wiring length [5]; and (iv) support for the realization of mixed-technology chips [7, 8].

The majority of the 3D IC research so far is focused on how to take advantage of the performance, power, smaller form-factor, and heterogeneous integration benefits offered by 3D integration [1, 2, 4–11]. However, when deciding to adopt this emerging technology as a mainstream design approach, the cost of 3D integration must be considered . *All the advantages of 3D ICs ultimately have to be translated into cost savings when a design strategy has to be decided [12].* For example, designers may ask themselves questions like:

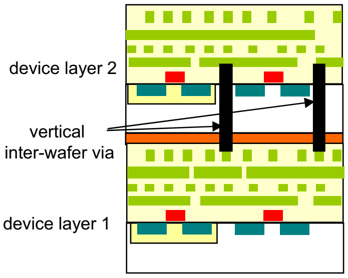

Fig. 1. A conceptual 3D IC: two device layers are stacked together with through-silicon-via connection [11].

- *Do all the benefits of 3D IC design come with a much higher cost?* For example, 3D bonding incurs extra process cost, and the Through-Silicon Vias (TSVs) may increase the total die area, which has a negative impact on the cost; However, smaller die sizes in 3D ICs may result in higher yield than that of a larger 2D die, and reduce the cost.
- *How can 3D integration be achieved in a cost-effective way?* For example, to re-design a small chip may not gain the cost benefits of improved yield resulted from 3D integration. In addition, if a chip is to be implemented in 3D, how many layers of 3D integration would be cost effective? and should one use wafer-to-wafer or die-to-wafer stacking [5]?
- *Are there any design options to compensate the extra 3D bonding cost?* For example, in a 3D IC, since some global interconnects are now implemented by TSVs, it may be feasible to use fewer number of metal layers for each 2D die. In addition, heterogeneous integration via 3D could also help cost reduction.

Cost analysis for 3D ICs at the early design stage is critical to answer these questions, and help the decision making on whether 3D integration should be used, and what design options should be adopted (such as the number of layers and the bonding approaches).

Cost-efficient design is the key for the future wide adoption of the emerging 3D IC design, and 3D IC cost analysis needs close coupling between 3D IC design and 3D IC process[1]. In this paper, we first study the design estimation method for 3D ICs at the early design stage (Section II), and propose a cost analysis model to study the cost implication for 3D ICs (Section III). Using the design estimation methods and the 3D cost

[1] IC Cost analysis needs a close interaction between designers and foundry. We work closely with our industrial partners to perform cost analysis. However, we cannot disclose absolute numbers for the cost, and therefore in this paper, we either use arbitrary units (a.u.) or normalized value to present the data.

978-1-4244-2749-9/09/$25.00 ©2009 IEEE

analysis model, we compare the estimated cost between 2D and 3D designs, and investigate the impact of various factors on the cost, as well as possible ways that 3D integration offers to reduce the cost. A cost-driven 3D IC design flow is also proposed to guide the design space exploration for 3D ICs toward a cost-effective direction (Section IV). Finally, we conclude the paper in Section VI.

II. Early Design Estimation for 3D ICs

To facilitate the design decision of using 3D integration from a cost perspective, it is necessary to perform cost analysis at the early design stage when detailed design information is not available. The cost of an IC chip is closely related to the die area. In 3D ICs, the Through-Silicon Vias (TSVs) may incur extra area overhead. However, it is possible to use fewer number of metal layers for routing in 3D ICs, which helps in cost reduction.

In this section, we describe how to estimate the die area, the metal layers for feasible routing, and the impact of TSVs on die area, at the very early design stage, when only limited design information (such as the estimation of the gate counts in the design) is available. Such an early estimation will facilitate the 3D IC cost analysis discussed in Section III.

A. Rent's Rule

Our early design estimation is based on the well-known *Rent's Rule* [13]. Rent's Rule reveals the trend between the number of signal terminals and the number of internal gates. It is an empirical result based on the observations of existing designs, and can be expressed as:

$$T = kN_g^p \tag{1}$$

where the parameters k and p are Rent's coefficient and exponent, N_g is the gate counts, and T is the number of signal terminals.

Using Rent's Rule, it becomes possible to further estimate the average wire length [14] and the wire length distribution [15]. The average wire length can be given by:

$$\bar{R_m} = \frac{2}{9}\frac{1-4^{p-1}}{1-N_g^{p-1}}\left(7\frac{N_g^{p-0.5}-1}{4^{p-0.5}-1} - \frac{1-N_g^{p-1.5}}{1-4^{p-1.5}}\right); \tag{2}$$

When $p = 0.5$, the expression can be calculated using *L'Hospital Law* [14].

According to Rent's rule, the wire length distribution function $i(l)$ has the forms as follows:

Region I: $1 \leq l \leq \sqrt{N_g}$

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

Region II: $\sqrt{N_g} \leq l < 2\sqrt{N_g}$

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

where l is the interconnect length in units of gate pitches, α is the fraction of the on-chip terminals that are sink terminals and is related to the average fanout of a gate ($f.o.$) as follows:

$$\alpha = \frac{f.o.}{f.o. + 1} \tag{5}$$

and Γ is given by

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

B. Die Area and Metal Layer Estimator

At the early design stage, the die area can be estimated as a function of the gate counts:

$$A_{die} = N_g A_g \tag{7}$$

where N_g is the number of gates, and A_g is an empirical parameter that shows the proportional relationship between area and gate counts. Based on empirical data from our industrial designs, in this work, we assume that $A_g = 3125\lambda^2$, in which λ is half of the feature size for a specific technology node.

The number of required metal layers for routing depends on the complexity of the interconnections. A simple metal layer estimation can be derived from the average wire length [16]:

$$n_w = \frac{f.o.\bar{R_m}p_w}{e_w}\sqrt{\frac{N_g}{A_{die}}} \tag{8}$$

where $f.o.$ refers to the average gate fanout, p_w to wire pitch, e_w to the utilization efficiency of metal layers, $\bar{R_m}$ to the average wire length, which is formulated by Equation 2, and n_w to the number of metal layers. Such simplified model is based on the assumptions that each metal layer has the same utilization efficiency and the same wire width [16]. However, such assumptions may not be valid in real design [17]. Moreover, the model in [16] does not include the impact of TSV area overhead, which will be a considerable penalty when the complexity of 3D ICs increase.

To improve the estimation of the number of metal layers needed for feasible routing, we propose a new 3D routability model, which is based on the wire length distribution rather than a simple estimation of average wire length. The basic idea of this model is explained as follows:

- *Estimate the available routing area of each metal layer with the expression:*

  $$K_i = \frac{A_{die}\eta_i - 2A_v\left(N_g f.o. - I(l_i)\right)}{w_i} \tag{9}$$

  where i is the metal layer, η_i is the layer's utilization efficiency, w_i is the layer's wire pitch, A_v is the blockage area of each via, and function $I(l)$ is the cumulative integral of the wire length distribution function $i(l)$, which is expressed in Equation 4.

- *Assume that shorter interconnects are routed on lower metal layers. Starting from Metal 1, we route as many interconnects as possible on the current metal layer until the available routing area is used up.* The interconnects routed on each metal layer can be express as:

  $$\chi L(l_i) - \chi L(l_{i-1}) \leq K_i \tag{10}$$

  where $\chi = 4/(f.o. + 3)$ is a factor accounting for the sharing of wires between interconnects on the same net [15] [18]. The function $L(l)$ is the first-order moment of $i(l)$.

- *Repeat the same calculations for each metal layer in a bottom-up manner until all the interconnects are routed properly.*

By applying the estimation methodology introduced above, we can predict the die area and the number of metal layers at the early design stage where we only have the number of gates as the input. Fig. 2 shows an example which estimates the area and the number of metal layers of 65nm designs with different scale of gates.

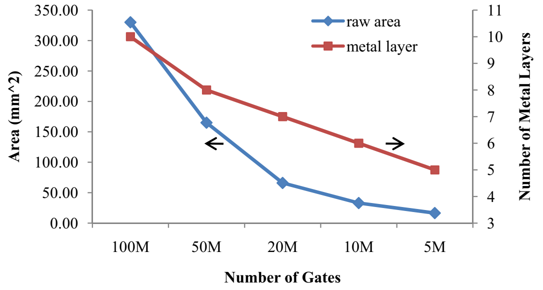

Fig. 2. Early Design Estimation of Die Area and Metal Layer (65nm process) (The estimation is well correlated with the state-of-the-art microprocessor designs. For example, Sun SPARC T2 [19] contains about $500M$ transistors (equivalent to $125M$ gates), with an area of $342mm^2$ and 11 metal layers)

Fig. 2 shows an important implication for 3D IC cost reduction: *When a large 2D chip is partitioned into multiple smaller dies with 3D stacking, each smaller die requires fewer number of metal layers to satisfy the interconnect routability requirements*. Such metal layer reduction could offset the extra cost resulting from 3D stacking.

C. The Impact of TSVs

The impact of Through-Silicon Vias (TSVs) used in 3D stacking on the cost analysis are two folds:

- In 3D ICs, some global interconnects are now implemented by TSVs, going between stacked ides. This could lead to the reduction of the total wire length, and provides opportunities for metal layer reduction for each smaller die;
- On the other hand, 3D stacking with Through-Silicon Vias (TSVs) may increase the total die area, since the silicon area where TSVs punch through may not be utilized for building devices or 2D metal layer connections (Based on current TSVs technologies, the diameter of TSVs ranges from $0.2\mu m$ to $10\mu m$ [4]).

Consequently, it is important to estimate the number of TSVs and the impact on the die area increase.

To predict the number of required TSVs for a certain partition pattern, a derivation of Rent's Rule describing the relationship between the interconnections (X) and gates (N_g) can be used [14]:

$$X = \alpha k N_g \left(1 - N_g^{p-1}\right) \qquad (11)$$

As illustrated in Fig 3, the number of TSVs can be estimated by:

$$\begin{aligned} X_{TSV} = {} & \alpha k_{1,2}(N_1 + N_2)\left(1 - (N_1 + N_2)^{p_{1,2}-1}\right) \\ & -\alpha k_1 N_1 \left(1 - N_1^{p_1-1}\right) - \alpha k_2 N_2 \left(1 - N_2^{p_2-1}\right) \end{aligned} \qquad (12)$$

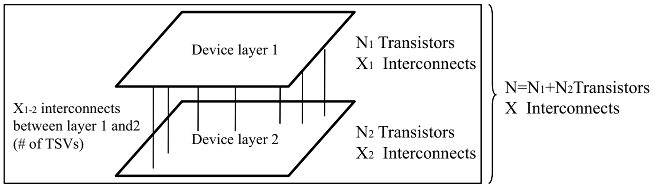

Fig. 3. The basic idea of how to estimate the number of TSVs

where $k_{1,2}$ and $p_{1,2}$ are the equivalent Rent's coefficient and exponent.

The area overhead caused by TSVs can be modeled as follows:

$$A_{3D} = A_{die} + N_{TSV/die} A_{TSV} \qquad (13)$$

where A_{die} is calculated by die area estimator, $N_{TSV/die}$ is the equivalent number of TSVs on each die, A_{TSV} is the size of TSVs, and A_{3D} is the final 3D component die area.

III. 3D Cost Model

3D integration involves stacking multiple dies through traditional fabrication processes. There are several different ways to stack separate dies together [3], and the TSV-based approach is the most promising approach. In addition to conventional 2D processes, 3D integration needs extra fabrication steps such as forming TSVs via laser drilling or etching, wafer thinning, and wafer bonding.

We model the cost brought by each step during the 3D fabrication and the cost analysis can be divided into the die cost model and the 3D bonding cost model as shown in Fig. 4.

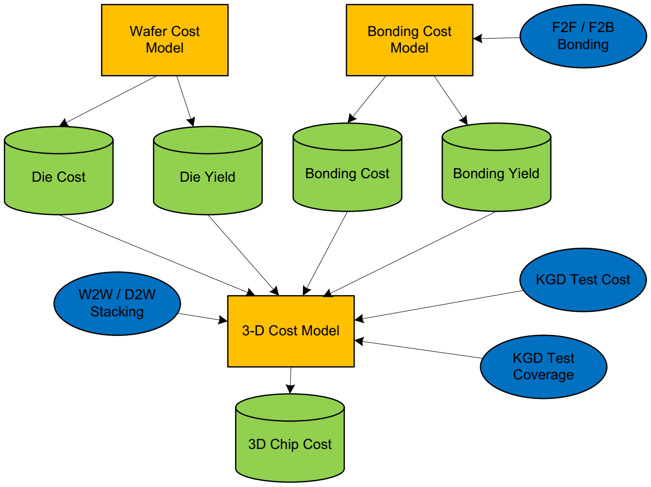

Fig. 4. The overview of the proposed 3D cost model

- *Wafer Cost Model.* The key factor of the die cost model is the die area. If we assume that the wafer cost, the wafer yield, and the defect density are constant for a specific foundry using a specific technology node, the impact of die areas can be formulated by two expressions [20] as following:

$$N_{die} = \frac{\pi \times (\phi_{wafer}/2)^2}{A_{die}} - \frac{\pi \times \phi_{wafer}}{\sqrt{2 \times A_{die}}} \qquad (14)$$

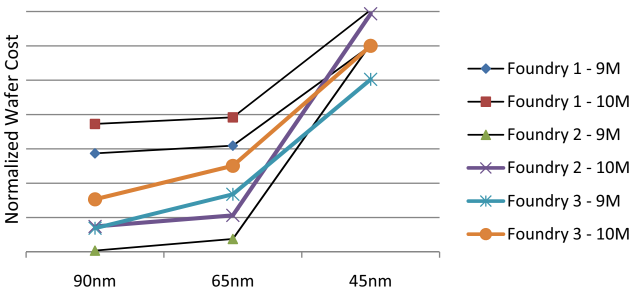

Fig. 5. A batch of data calculated by the wafer cost model. The wafer cost varies from different processes, different number of metal layers, different foundries, and some other factors.

$$Y_{die} = Y_{wafer} \times \frac{\left(1 - e^{-2A_{die}D_0}\right)}{2A_{die}D_0} \tag{15}$$

where N_{die} is the number of dies per wafer, ϕ_{wafer} is the diameter of the wafer, Y_{die} and Y_{wafer} are the yields of dies and wafers respectively, and D_0 is the defect density of the wafer.

Our wafer cost model obtained from different foundries includes material cost, labor cost, foundry margin, number of reticles, cost per reticle, and other miscellaneous cost [21]. Fig. 5 shows the predicted wafer cost of $90nm$, $65nm$, and $45nm$ processes, with 9 or 10 layers of metal, for three different foundries, respectively.

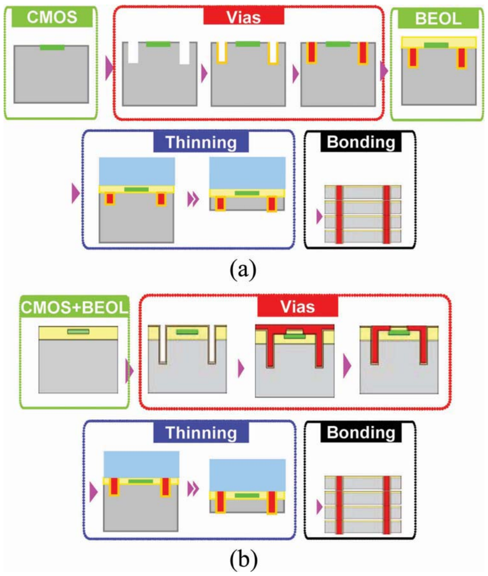

Fig. 6. Fabrication steps for 3D ICs: (a) TSVs are formed before BEOL process, thus TSVs only punch through the silicon substrate but not the metal layers; (b) TSVs are formed after BEOL process, thus TSVs punch through not only the silicon substrate but the metal layers as well.

- *3D Bonding Cost.* The extra fabrication steps required by 3D integrations consist of TSV forming, thinning, and bonding. There are two ways to build 3D TSVs: *laser drilling or etching*. Laser drilling is only suitable for a small number of TSVs (hundreds to thousands) while etching is suitable for a large number of TSVs. TSV etching process is similar to building conventional vias between metal layers, but as its name implies, TSV is "through-silicon". There are two approaches for TSV etching: (1) *TSV-first approach:* TSVs can be formed during the 2D die fabrication process, before the Back-End-of-Line (BEOL) processes. Such an approach is called *TSV-first* approach, and is shown in Fig. 6(a); (2) *TSV-later approach:* TSVs can also be formed after the completion of 2D fabrications, after the BEOL processes. Such an approach is called *TSV-later* approach, and is shown in Fig. 6(b). Our 3D bonding cost model is based on the 3D process from our industry partners, with the assumption that the yield of each 3D process step is 99%.

- *Overall 3D Cost Model.* In addition to the wafer cost model and the bonding cost model, the entire 3D cost model also depends on some design options, such as Die-to-Wafer/Wafer-to-Wafer bonding, Face-to-Face/Face-to-Back bonding, and Known-Good-Die cost [12].

  For D2W bonding, the bare chip cost before package is calculated by:

$$C_{D2W} = \frac{\sum_{i=1}^{N}\left(C_{die_i + C_{KGDtest}}\right)/Y_{die_i} + (N-1)C_{bonding}}{Y_{bonding}^{N-1}} \tag{16}$$

  For W2W bonding, the calculation becomes:

$$C_{W2W} = \frac{\sum_{i=1}^{N} C_{die_i} + (N-1)C_{bonding}}{\left(\Pi_{i=1}^{N} Y_{die_i}\right) Y_{bonding}^{N-1}} \tag{17}$$

  In order to support multiple-layer bonding, the default bonding mode is Face-to-Back. If Face-to-Face mode is used, there is one more component die that doesn't need the thinning process, and the thinning cost of this die is subtracted from the total cost.

IV. System level 3D IC Design Exploration

Based on the early design estimation methods and the 3D cost analysis model described in the previous sections, we use the IBM Common Platform foundry cost model as an example to perform a series of design analysis at the system level, investigating the impact of different design options on the 3D IC cost, and inducing a few rules of thumb as 3D IC design guidelines from cost perspectives.

A. Evaluation of the TSV's Impact on Die Area

As mentioned in Section II, building TSVs in 3D ICs not only incurs additional process cost but also causes area overhead. The area overhead affects the die yield and the wafer utilization. Based on the TSV estimation equation (Equ.12), we set the exponent parameter $p = 0.63$ and the coefficient parameter $k = 1.4$, according to *Bakoglu's* research [22]. We further assume that the number of 3D layers is N, and all the gates are uniformly partitioned into N layers. We choose the pitch size of TSVs to be $8\mu m$. Using the early design estimation, the predicted TSV impacts under 65nm process are shown in Fig. 7.

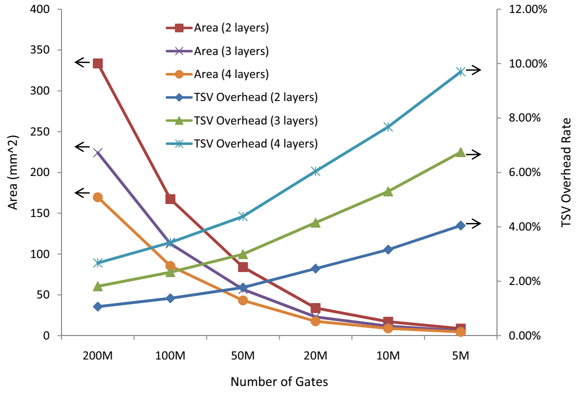

Fig. 7. The total area and the percentage of area occupied by TSVs: For small designs, the TSV area overhead is near to 10%, but for large designs, the TSV area overhead is less than 4%.

Consistent with what Equation 13, the TSV area overhead increases with the increase of 3D layers. The overhead can reach as high as 10% for a small design (5M gates) using 4 layers. However, when the design is sufficiently large (200M gates) and 3D integration only stacks two dies together, the TSV area overhead is usually below 2%. To summarize, *for large design, the area overhead due to TSVs is acceptable*.

B. The Potential of Metal Layer Reduction in 3D ICs

Theoretically, when the number of gates is distributed evenly to multiple dies in 3D stacking, the total wire length on each smaller die equals the total wire length for a large 2D chip divided by the number of dies. In addition, as discussed in Section 2, the total wire length decreases as the number of 3D layers increases, due to the existence of TSVs. Comparing these two factors together, the routing complexity on each 3D component die is much less than that of the 2D baseline design. As a result, it becomes possible to remove one or two metal layers on each smaller die in 3D stacking.

Estimated by the 3D routability model discussed in Section II, we can obtain the prediction of the metal layer reduction effect and the result is listed in Table. I.

TABLE I
THE NUMBER OF REQUIRED METAL LAYERS PER DIE (65NM MICROPROCESSOR)

Gate Counts	1-layer 2D	2-layer 3D	3-layer 3D	4-layer 3D
5M	5	5	5	4
10M	6	5	5	5
20M	7	6	5	5
50M	8	7	7	6
100M	10	8	7	7
200M	12	10	9	8

Although the result shows that there is little opportunity to reduce metal layers in a relatively small design (such as the 5M-gate design), the metal layer reduction becomes more and more obvious with the growth of design complexities. For instance, the number of required metal layers can be reduced by 2, 3, and 4 when a large 2D design (i.e. 200M gates) is uniformly partitioned into 2, 3, and 4 separate dies, respectively.

To summarize, *in 3D IC design, it is possible to use fewer number of metal layers for each smaller die, compared to the baseline 2D design. Such metal layer reduction could offset the extra bonding cost in 3D integration.*

C. Bonding Techniques: D2W or W2W

Die-to-Wafer (D2W) and Wafer-to-Wafer(W2W) are two different ways to bond multiple dies together in 3D integration [3]. Section III discusses the modeling for these two methods. D2W bonding can achieve a higher yield by introducing Known-Good-Die(KGD) test, while W2W bonding does not need any test before bonding and it is easy for die alignments with higher throughput, at the expense of yield loss [3]. Since both of D2W and W2W have their pros and cons, we use our 3D cost model to find out which one is more suitable for 3D integration technologies.

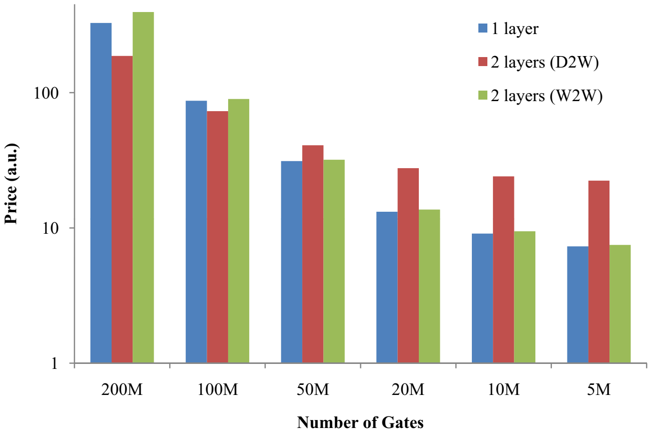

Fig. 8. The cost comparison among 2D, 2-layer D2W, and 2-layer W2W under 65nm process. (With the consideration of TSV area overheads and metal layer reductions

Fig. 8 shows the cost comparison among the conventional 2D process, the 2-layer D2W bonding, and the 2-layer W2W bonding. It can be observed that, although the cost of W2W is lower than that of D2W in the cases of small design, the cost of W2W is always higher than that of 2D processes. This phenomena can be explained by the relationship between areas and yields, which is expressed by Equation 15. For the W2W bonding, the yield of each component die does increase due to the area reduction, but when all the dies are stacked together without pre-stacking test, the chip yield equals to the product of the yield of each component die. Thus the overall yield of W2W-bonded 3D chips becomes as low as the one of 2D chips. After the extra bonding cost is included, it becomes reasonable to understand why W2W is always more expensive than conventional 2D.

To summarize, *from yield perspective, Die-to-Wafer (D2W) stacking has cost advantage over Wafer-to-Wafer (W2W) stacking*, based on our wafer cost model and 3D bonding cost model.

D. Cost vs Number of 3D Layers

Based on the early design estimation methods that predict the 3D IC die area, the 3D TSV impact, and the metal layer reduction effect, we can further use these design-related parameters as the inputs of the 3D IC cost model proposed in Section III, and estimate the cost of each 3D design option.

First of all, we select the IBM Common Platform 65nm model and compare the cost of the 2D baseline design with its 3D counterparts, which have 2, 3, and 4 dies stacked together. Fig. 9 shows the cost estimations based on the assumption that the 2D baseline design is partitioned uniformly.

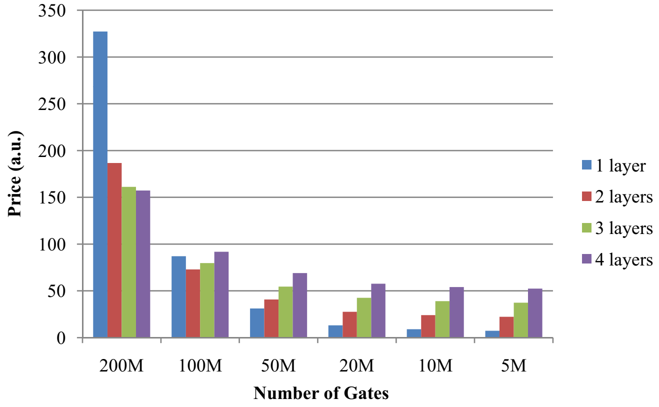

Fig. 9. The cost of 3D ICs with different number of layers under 65nm process (With the consideration of TSV area overheads and metal layer reductions).

It can be observed from Fig. 9 that, the cost increases with the growth of the chip size dramatically due to the exponential relationship between die sizes and die yields. Because the yield becomes more sensitive to the die area when the area is large, splitting a large 2D design into multiple dies is more likely to reduce the total cost than splitting a small 2D design.

Another observation is that the optimal number of 3D layers (in terms of cost) may vary, depending on how large the design it is. For example, the most cost-effective way for a 200M-gate design is to do a 3D stacking with 4-layer 3D partitioning; but for a 100M-gate design, the most cost-effective option is to use 2-layer 3D integration; finally, when the original 2D design is relatively small (less than 50M gates), the conventional 2D fabrication is always the cheapest one, because the 3D bonding cost starts to dominate and the yield improvement due to 3D stacking is small.

Repeating the experiment using different technology nodes with IBM Common Platform technology models, we estimate a set of boundaries that indicate where the 2-layer 3D stacking starts to be more cost-effective than the traditional 2D process. The data are listed in Table II. If we convert the number of gates into chip size, the enabling point of 2-layer 3D process is about $250mm^2$. The enabling point of more than 2 layer of stacking can be even larger.

To summarize, *3D integration is cost-effective for large designs, but not for small designs; The optimal number of 3D layers (from a cost perspective) increases as the gate count increases.*

E. Heterogenous Stacking

All the discussions above are focused on homogeneous stacking. However, one of the biggest advantage of 3D integration is that it supports heterogeneous stacking because different types of components can be fabricated separately.

TABLE II
THE ENABLING POINT OF 3D FABRICATIONS

Process (IBM Common Platform)	45nm	65nm	90nm	130nm
Enabling Point (number of gates)	$143M$	$76M$	$40M$	$21M$

Using today's high-performance microprocessors as an example, a large portion of the silicon area is occupied by on-chip SRAM or DRAM, and non-volatile memory can also be integrated as on-chip memory [7]. However, the fabrication processes for these different modules are different. For instance, while the underlying conventional CMOS logic circuits require 1-poly-9-copper-1-aluminum interconnect layers, the one needed by DRAM modules is 7-poly-3-copper and the one needed by Flash modules is 4-poly-1-tungsten-2-aluminum. As a result, heterogeneous integration will dramatically increase the cost. As an example, Intel shows that heterogenous integration for large 2D SoC could boost the chip cost by 3X [23].

Separating the fabrication of heterogenous technology and stacking them with 3D integration could be a cost effective way for such systems. Here, we take the OpenSPARC T2 [19] as a case study. The original 2D OpenSPARC T2 chip has the area of $342mm^2$ and fabricated with TI 65nm process with 11 metal layers. About half of the die area is attributed to on-chip SRAM cache. One way of using 3D integration for such microprocessor is to partition all SRAM modules on one die and all the rest of modules on the other die, similar to the recent Intel 80-core Tera-scale chip [24]. Applying the early design estimation method in Section II and choosing the Rent's parameters for SRAM as $p = 0.12$, $k = 6$, we estimate that the number of metal layers for the SRAM modules can be reduced to 5. And we further estimate the total chip total by using our 3D IC cost model. The comparison is shown in Fig. 10.

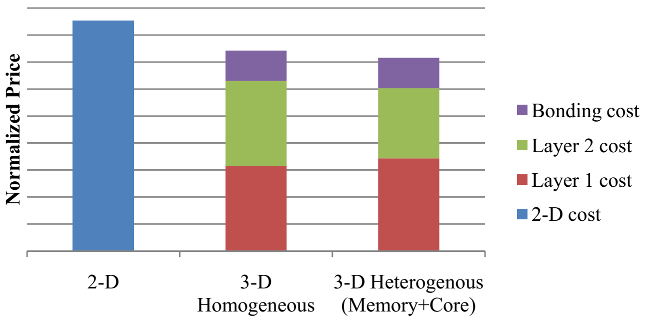

Fig. 10. The estimated cost of OpenSPARC T2 by using conventional 2D, homogeneous 3D partitioning, and heterogeneous 3D partitioning: Fabricating the memory and the core part separately can further reduce the cost.

To summarize, *the ability to enable heterogenous integration offers extra opportunities to reduce the total cost in 3D IC designs.*

V. COST-DRIVEN 3D DESIGN FLOW

The 3D IC cost analysis discussed above is conducted before the real design, and all the inputs of the cost model are predicted from early design estimation. However, if the same cost analysis methodology is applied during design time, using the real design data, such as die area, TSV interconnects, and metal interconnects, as the inputs of the cost model, then a cost-driven 3D IC design flow becomes possible. Fig. 11 shows a proposed cost-driven 3D IC design flow. The integration of 3D IC cost models into design flows guides the designer to optimize their 3D IC design and eventually to manufacture low-cost product.

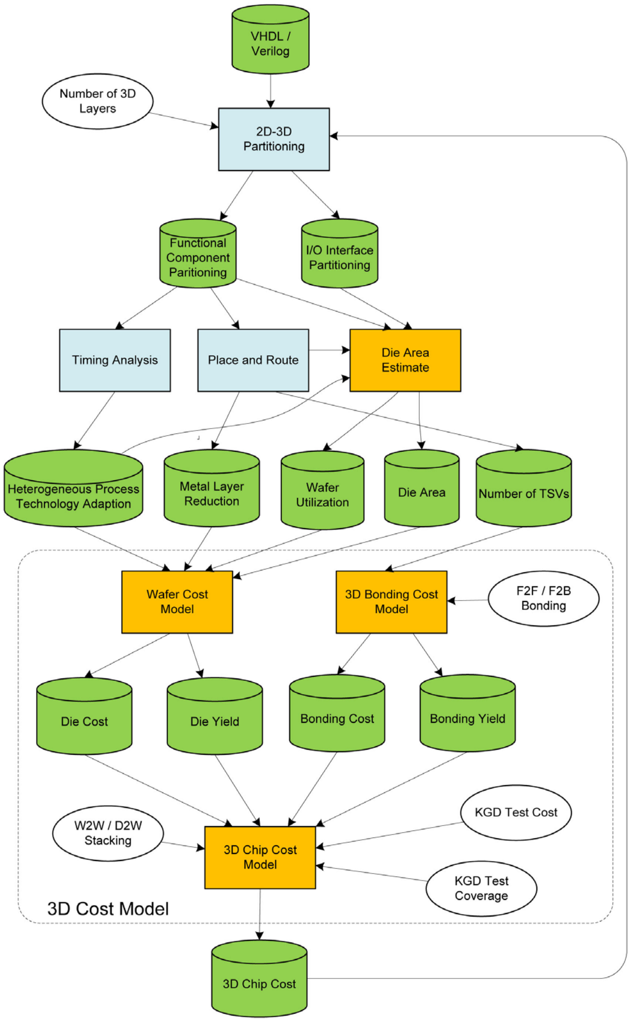

Fig. 11. The scheme of a Cost-Driven 3D IC Design Flow

Such a cost analysis/reducion EDA flow consists of three groups of operations: design-related operations, cost modeling related operations, and cost reduction operations:

- **Design-related operations** include *3D Partitioning*, *Timing Analysis*, and *Placement & Routing*, all of which are part of a typical 3D chip design flow. Such operations can affect the cost estimation. For example, different partitioning strategies may result in different components on a die with different number of I/O interfaces. Placement & Routing determines the interconnect topology of each layer, and results in different number of Through-Silicon-Vias needed for 3D integration, which impacts the bonding overhead.

- **Cost modeling operations** include *die area estimation* (which evaluates the area of the die in each layer), *wafer cost modeling* (which evaluates the cost of each stacking layer), *3D bonding cost model* (which evaluates the the cost of stacking multiple layers of dies together and the cost of fabricating 3D vias), as well as *stacking cost model*, which evaluates the cost related to different stacking options. For example, die-to-wafer (D2W) stacking requires Known-Good-Die test before stacking a die on top of other dies, and incurs additional cost for die test, but it can improve the yield of stacked-chips. These models are described in previous sections.

- **Cost reduction operations** include possible ways to to reduce the cost. For example, one approach is called *Heterogenous Process Technology Stacking*: components that are not critical can be partitioned onto a die with a slower (but cheaper) technology fabrication (such as 0.18um CMOS), while critical components are partitioned onto a die with a more advanced (faster but expensive) technology fabrication (such as 65nm CMOS). The second approach is called *Metal Layer Reduction*: when moving from 2D to 3D design, each die itself may be able to use fewer number of the metal layers to do routing, which can save the back-end process cost.

Such a unique and close integration of cost analysis with 3D EDA design flow has two advantages. First, as we discussed earlier, many design decisions (such as partitioning and placement & routing) can affect cost analysis. Closely coupling cost analysis with 3D EDA flow can result in more accurate cost estimation. Second, cost analysis result can drive 3D EDA tools to carry out a more cost-effective optimization, in addition to considering other design goals (such as performance and power).

A. Case Study: Two-Layer OpenSPARC T1 3D Processor

We use the Sun OpenSPARC T1 processor [25] [2] as a case study to demonstrate how the extra manufacture cost related to 3D technology could be offset by the cost reduction methods we have mentioned in the last section. As a result of cost reduction, the total cost of a 3D chip could be lower than that of its 2D counterpart.

As we mentioned in the previous section, the two major 3D cost reduction methods are:(1) *Metal Layer Reduction*, which reduce the number of metal layers during fabrication by taking advantage of the third routing dimension introduced by 3D technology; and (2) *Heterogenous Process Technology Stacking*, which partitions the non-critical components into a specific layer manufactured using older and cheaper process node.

There are two partitioning approaches that could help reduce the cost using 3D stacking:

- (1) Coarse-granularity partitioning. The OpenSPARC T1 processor can be split into processor cores and cache

[2] The design has open-source design verilog code and synthesis scripts on http://www.opensparc.net. The original T1 chip was fabricated at 90nm technology, with $300M$ transistors and an area of $340mm^2$.

banks; In Section IV.E, we have seen that such partitioning can help the cost reduction because separating memory from logic layer can reduce the cost.

- (2) Fine-granularity partitioning. In this approach, we partition the components at the unit-level [1], using our cost-driven design flow proposed in Fig. 11.

Following the fine-granularity partitioning approach, we divide the entire 8-core OpenSPARC T1 processor into 2-layer fine-grained partitioning, and ensure that the timing requirements are not changed. With such fine-granularity partitioning, there are two possible ways to save cost:

- (a)*90nm-90nm stacking*. In this approach, both layers are implemented using 90nm technology. Carefully partitioning these units into two layers, we can make the area of the resulting layer equals to each other, and the critical path remains unchanged. From the synthesis results using a 90nm standard cell library, we observe that the total area of a single core in the 8-core SPARC T1 is about 10.63mm^2. With 2-layer 3D partitioning, the area of a single-core is reduced to 7.18mm^2 and 7.03mm^2 respectively. Based on our cost model, the 3D implementation cost is \$125, comparing to the original 2D cost of \$146.

- (b)*90nm-130nm heterogenous process technology stacking*. In this approach, the timing analysis results are used to find out which sets of components are not on the critical paths and can be moved to a slower layer that is synthesized with 130nm standard cell library. Based on the synthesis results and the cost analysis, the cost is further reduced to \$121.

VI. Conclusion

To overcome the barriers in technology scaling, three-dimensional integrated circuit (3D IC) is emerging as an attractive option for future IC design. However, fabrication cost is one of the important considerations for the wide adoption of the 3D integration. System-level cost analysis at the early design stage to help the decision making on whether 3D integration should be used for the application is very critical.

To facilitate the system level cost analysis, we study the design estimation method for 3D ICs at the early design stage, and propose a cost analysis model to study the cost implication for 3D ICs. Based on the cost analysis, we identify the design opportunities for cost reduction in 3D ICs, and provide a few design guidelines on cost-effective 3D IC designs. Our research is complementary to the existing research on the 3D IC benefits analysis on other design goals (such as performance and power analysis).

Acknowledgement

The authors would like to thank Dr. Larry Smith from SEMATECH, Dr. Mike Ignatowski from IBM, Dr. Sam Gu from Qualcomm, and Dr. Pol Marchal from IMEC for the valuable discussion and guidance on this research. This work was supported in part by NSF grants of CAREER 0643902, CCF 0702617, a grant from Qualcomm, and IBM Faculty Award.

References

[1] K. Bernstein, “New Dimension in Performance,” *EDA Forum*, vol. 3, no. 2, 2006.

[2] T. Vucurevich, “The Long Road to 3D Integration: Are We There Yet?” in *Keynote speech at the 3D Architecture Conference*, 2007.

[3] W. R. Davis, J. Wilson, S. Mick, J. Xu, H. Hua, C. Mineo, A. M. Sule, M. Steer, and P. D. Franzon, “Demystifying 3D ICs: the Pros and Cons of Going Vertical,” *IEEE Design and Test of Computers*, vol. 22, no. 6, pp. 498– 510, 2005.

[4] G. H. Loh, Y. Xie, and B. Black, “Processor Design in 3D Die-Stacking Technologies,” *MICRO*, 2007.

[5] Y. Xie, G. H. Loh, B. Black, and K. Bernstein, “Design Space Exploration for 3D Architectures,” *J. Emerg. Technol. Comput. Syst.*, vol. 2, no. 2, pp. 65–103, 2006.

[6] C. Ababei, Y. Feng, B. Goplen, H. Mogal, T. Zhang, K. Bazargan, and S. S. Sapatnekar, “Placement and Routing in 3D Integrated Circuits,” *IEEE Design and Test of Computers*, vol. 22, no. 6, pp. 520– 531, 2005.

[7] X. Dong, X. Wu, G. Sun, Y. Xie, H. Li, and Y. Chen, “Circuit and Microarchitecture Evaluation of 3D Stacking Magnetic RAM (MRAM) as a Universal Memory Replacement,” in *Design Automation Conference*, 2008, pp. 554–559.

[8] G. Sun, X. Dong, , Y. Xie, J. Li, and Y. Chen, “A Novel 3D Stacked MRAM Cache Architecture for CMPs,” in *International Symposium on High Performance Computer Architecture*, 2009.

[9] J. Cong, J. Wei, and Y. Zhang, “A Thermal-Driven Floorplanning Algorithm for 3D ICs,” in *International Conference on Computer Aided Design*, 2004, pp. 306–313.

[10] S. Das, A. Chandrakasan, and R. Reif, “Design Tools for 3-D Integrated Circuits,” in *ASPDAC*, 2003, pp. 53–56.

[11] Y. Deng and W. Maly, “A Feasibility Study of 2.5D System Integration,” in *Custom Integrated Circuits Conference, 2003. Proceedings of the IEEE 2003*, 2003, pp. 667–670.

[12] L. Smith, G. Smith, S. Hosali, and S. Arkalgud, “3D: It All Comes Down to Cost,” *Proceedings of RTI Conference of 3D Architecture for Semiconductors and Packaging*, 2007.

[13] B. S. Landman and R. L. Russo, “On a Pin Versus Block Relationship For Partitions of Logic Graphs,” *IEEE Trans. on Computers*, vol. C-20, no. 12, pp. 1469–1479, 1971.

[14] W. E. Donath, “Placement and Average Interconnection Lengths of Computer Logic,” *IEEE Trans. on Circuits and Systems*, vol. 26, no. 4, pp. 272–277, 1979.

[15] J. A. Davis, V. K. De, and J. D. Meindl, “A Stochastic Wire-Length Distribution for Gigascale Integration (GSI)łPart I: Derivation and Validation,” *IEEE Trans. on Electron Devices*, vol. 45, no. 3, pp. 580–589, 1998.

[16] R. Weerasekera, L.-R. Zheng, D. Pamunuwa, and H. Tenhunen, “Extending Systems-on-Chip to the Third Dimension: Performance, Cost and Technological Tradeoffs,” in *ICCAD*, 2007, pp. 212–219.

[17] A. B. Kahng, S. Mantik, and D. Stroobandt, “Toward Accurate Models of Achievable Routing,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 20, no. 5, pp. 648–659, 2001.

[18] P. Chong and R. K. Brayton, “Estimating and Optimizing Routing Utilization in DSM Design,” in *Workshop System-Level Interconnect Prediction*, 1999.

[19] S. C. Marc Tremblay, “A Third-Generation 65nm 16-Core 32-Thread Plus 32-Scout-Thread CMT SPARC(R) Processor,” in *International Solid State Circuit Conference*, 2008, pp. 82–83.

[20] J. Rabaey, A. Chandrakasan, and B. Nikolic, “Digital Integrated Circuits,” *Prentice-Hall*, 2003.

[21] “IC Cost Model, 2008 revision 0808a,” in *IC Knowledge LLC*, 2008.

[22] H. B. Bakoglu, *Circuits, Interconnections, and Packaging for VLSI.* Addison-Wesley, 1990.

[23] S. Borkar, “3D-Technology: A System Perspective,” in *International 3D-System Integration Conference*, 2008.

[24] “http://techresearch.intel.com/articles/Tera-Scale/1421.htm,” 2007.

[25] “http://www.opensparc.net/,” 2008.