Fabrication Cost Analysis and Cost-Aware Design Space Exploration for 3-D ICs

Xiangyu Dong, *Member, IEEE,* Jishen Zhao, *Member, IEEE,* and Yuan Xie, *Senior Member, IEEE*

***Abstract*—3-D integration technology is emerging as an attractive alternative to increase the transistor count for future chips. The majority of the existing 3-D integrated circuit (IC) research is focused on the performance, power, density, and heterogeneous integration benefits offered by 3-D integration. All such advantages, however, ultimately have to translate into cost evaluation when a design strategy has to be decided. Consequently, system-level cost analysis at early design stages is imperative to decide on whether 3-D integration should be adopted. This paper presents a cost estimation method for 3-D ICs at early design stages and proposes a set of cost models that include wafer cost, 3-D bonding cost, package cost, and cooling cost. The proposed 3-D IC cost estimation method can help designers analyze the cost implication for 3-D ICs during the design space exploration at the early stage, and it enables a cost-driven 3-D IC design flow that can guide the design choice toward a cost-effective direction. Based on the proposed cost estimation method, this paper demonstrates two case studies that explore the cost benefits of 3-D integration for application-specific integrated circuit designs and many-core microprocessor designs style, respectively. Finally, this paper suggests the optimum partitioning strategy for future 3-D IC designs.**

***Index Terms*—3-D integration, application-specific integrated circuit (ASIC), cost, microprocessor.**

I. Introduction

THREE-DIMENSIONAL integrated circuit (3-D IC) [1]–[5] is emerging as an attractive option for overcoming the barriers in interconnect scaling, thereby offering an opportunity to continue improving the complementary metal-oxide-semiconductor (CMOS) device performance. There are various vertical 3-D interconnect technologies that have been explored [2], including wire bonding, microbump, contactless (capacitive or inductive), monolithic multiple-layer buried structure [6], and through-silicon via (TSV) interconnects. Davis *et al.* [2] have compared these technologies in terms of vertical density and practical limits. Among these technologies, TSV-based 3-D integration is the most promising one as it has the potential to offer the greatest vertical interconnect density with the minimum impact on conventional IC fabrication approaches. Consequently, we focus our paper on TSV-based 3-D ICs in this paper.

Manuscript received February 12, 2010; revised May 7, 2010; accepted July 7, 2010. Date of current version November 19, 2010. This work was supported in part by NSF, under Grants 0702617, 0903432, and 0905365, and by SRC Grants. This paper was recommended by Associate Editor H.-H. S. Lee.

The authors are with the Department of Computer Science and Engineering, Pennsylvania State University, University Park, PA 16802 USA (e-mail: xydong@cse.psu.edu; juz138cse.psu.edu; yuanxie@cse.psu.edu).

Digital Object Identifier 10.1109/TCAD.2010.2062811

In a TSV-based 3-D IC, multiple dies are first fabricated with traditional 2-D processes separately and later stacked together with TSVs. Fig. 1 shows a conceptual view of a 2-layer TSV-based 3-D IC. In general, the benefits of TSV-based 3-D integration are listed as follows:

1) shorter global interconnects [4], [5];
2) higher packing density and smaller footprint due to the addition of the third dimension;
3) higher performance due to the reduced average interconnect length and the higher memory bandwidth [1];
4) lower interconnect power consumption due to the reduced total interconnect length [4];
5) supports for the heterogenous chips with heterogenous technologies [7], [8].

The majority of the 3-D IC research so far is focused on how to take advantage of the performance, power, smaller form factor, and heterogeneous integration benefits that are offered by 3-D integration [1]–[5], [7], [8]. However, when it comes to the discussion on the adoption of such an emerging technology as a mainstream design approach, it all comes down to the question of the 3-D integration cost. *All the advantages of 3-D ICs ultimately have to be translated into cost savings when a design strategy has to be decided [9].* For example, designers may ask themselves questions like the following.

1) *Do all the benefits of 3-D IC design come with a much higher cost?* For example, 3-D bonding that incurs extra process cost and TSV forming that increases the total silicon die area both have a negative impact on the cost; However, the smaller silicon die size of each layer results in a higher die yield than that of a larger 2-D die and potential reduces the cost.
2) *How to implement 3-D integration in a cost-effective way?* For example, re-designing a relatively small chip might not gain the cost benefit as its die yield would not increase too much from 2-D to 3-D. On the other hand, if a relatively large design is to be implemented in 3-D, it is not clear how many layers of 3-D integration would be the most cost-effective or whether wafer-to-wafer or die-to-wafer stacking [4] is more cost-effective.
3) *Are there any design options to compensate the extra cost caused by 3-D integration?* For example, in a 3-D IC, since some global interconnects are now implemented by TSVs, it may be feasible to use fewer metal layers when fabricating each 2-D die. In addition, the heterogeneous integration capability of 3-D IC can

0278-0070/$26.00 © 2010 IEEE

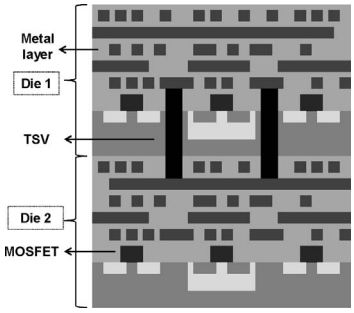

Fig. 1. Conceptual view of a 2-layer TSV-based 3-D IC: two dies are stacked together with TSVs.

potentially reduce the cost when multiple dies fabricated by incompatible processed are to be integrated into a single chip.

Considering 3-D integration has both positive and negative effects on the manufacturing cost, there is no clear answer yet to whether 3-D integration is cost-effective or how to achieve 3-D integration in a cost-effective way. In order to maximize the positive effect and compensate the negative effect on the manufacturing cost, it becomes critical to estimate the 3-D IC cost at early design stages. This early stage cost estimation can help the chip designers make decision on whether 3-D integration should be used or which 3-D partitioning strategy should be adopted. *Cost-effective design is the key for the future wide adoption of the emerging 3-D IC design, and 3-D IC cost analysis needs close coupling between the 3-D IC design and the 3-D IC process*.[1]

In this paper, we first propose a 3-D IC cost estimation method with a complete set of cost models that include wafer cost, 3-D bonding cost, package cost, and cooling cost. Using this cost estimation method along with the existing performance, power, area estimation tools, such as McPAT [10], we estimate the 3-D IC cost in two cases: one is for fully customized application-specific integrated circuit (ASIC), and the other is for many-core microprocessor designs. In these case studies, we also explore some guidelines showing what a cost-effective 3-D IC fabrication option should be in the future. Note that 3-D integration is not yet a mature technology with very well-developed and tested cost models, the optimal condition concluded by this paper is subject to parameter changes. However, the major contribution of this paper is to provide a cost estimation method for 3-D ICs. To our best knowledge, this is the first effort to model the 3-D IC cost with package and cooling cost included, while the majority of the existing 3-D IC research activities mainly focused on the circuit performance and power consumption.

The rest of this paper is organized as follows. After reviewing the related work in Section II, we introduce the background knowledge of early design stage estimation and extend the existing 2-D methodology to the 3-D domain in Section III. On the basis of the early design stage estimation method, a set of cost models is proposed. The four major components (wafer cost model, 3-D bonding cost model, package cost model, and cooling cost model) are discussed in detail in Section IV. By using these models, in Section V, we first perform a case study for fully customized ASICs and conclude some rules of thumb for cost-effective 3-D IC manufacturing. After that, we answer the question of what is the optimal partitioning strategy for a 3-D-stacked many-core microprocessor design in Section VI. Finally, we present our conclusion in Section VII.

[1]IC cost analysis needs a close interaction between designers and foundries. We work closely with our industrial partners to perform this 3-D IC 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.

II. Related Work

There are abundant research efforts put into 3-D integration research since it has been widely recognized as a promising technology and an alternative approach to preserve the chip scaling trend predicted by Moore's law. However, most of the previous work on 3-D integration just focused on its architectural benefits [3], [11]–[13] or its associated design automation tools [14]–[17]. Only a few of them considered the cost issue of 3-D ICs.

Coskun *et al.* [18] and Liu *et al.* [19] investigated the cost model for 3-D systems-on-chips (SoCs). Based on their studies, the key factor that affects the wafer cost is considered to be the die area. However, both of their cost models are based on simplified die area estimation methodologies. Coskun *et al.* [18] only considered the area of cores and memories, and they assumed the core area and the memory area are equal for the experiment convenience purpose. In addition, they assumed the number of TSVs per chip is a fixed value. Liu *et al.* [19] ignored the area overhead caused by TSVs in their study.

A yield and cost model was developed for 3-D ICs with particular emphasis on the stacking yield and how the wafer yield is affected by vertical interconnects [20]. However, there is no further analysis on the cooling cost, which is a nonnegligible part of the final system-level cost. Weerasekera *et al.* [21] also described a quantitative cost model for 3-D ICs. Their cost model features a detailed analysis of the cost per production step including the package cost. However, the package cost presented in this paper is based on plastic package, which is not an appropriate package type for either SoCs or microprocessors.

In this paper, we extended our previous studies [22], [42], and propose a complete set of 3-D cost models, which not only include wafer cost and 3-D bonding cost, but also include another two critical cost components, package cost and cooling cost. Both of them should not be ignored because while 3-D integration can potentially reduce the package cost by having a smaller chip footprint, the multiple 3-D-stacked dies might increase the cooling cost due to the increased power density. By using the proposed cost model set, we conduct two case studies on fully customized ASICs and many-core microprocessor designs, respectively. The experimental results show that our 3-D IC cost model makes it feasible to estimate the final system-level cost of the target chip design at early design stages. More importantly, it gives some guidelines to determine what a cost-effective 3-D IC manufacturing option should be.

III. Early Design Estimation for 3-D ICs

It is estimated that design choices made within the first 20% of the total design cycle time will ultimately result up to 80% of the final product cost [21]. Hence, to facilitate the design decision of using 3-D integration from a cost perspective, it is necessary to perform cost analysis at early design stages. While it is straightforward to sum up all the incurred cost after production, predicting the final chip cost at early design stages is a big challenge because most of the detailed design information is not available at that stage. Taking two different IC design styles (*the ASIC design style and the many-core microprocessor design style*) as examples.

1) *ASIC design style:* at early design stages, probably the only available information is the system-level (such as behavioral level or RTL level) design specification and the approximate gate count that roughly estimated from a quick pass of high-level/logic synthesis tools or simply from past design experience.
2) *Microprocessor design style:* the information available at the early stages of many-core microprocessor designs is also very limited. The design specification may only include information as follows: 1) the number of microprocessor cores; 2) the type of microprocessor cores (in-order or out-of-order); and 3) the number of cache levels and the cache capacity of each level. All these specifications are on the architectural level. Referring to previous design experience, it is feasible to achieve a rough estimation on the gate count for logic-intensive cores and the cell count for memory-intensive caches, respectively.

Consequently, it is very likely that at early design stages, the cost estimation is simply based on the design style and a rough gate/cell count for the design as the initial starting point. In this section, we describe how to translate the logic gate count (or the memory cell count) into higher level estimations, such as the die area, the metal layer count, and the TSV count, given a specified fabrication process node. The die area estimator and the metal layer estimator are two key components in our cost model: 1) larger die area usually causes lower die yield, and thus leads to higher chip cost, and 2) the fewer the metal layers are required, the less the number of fabrication steps (and fabrication masks) are needed, which reduces the chip cost. It is also important to note that the 3-D partitioning strategy can directly affect these two numbers: 1) 3-D integration can partition the original 2-D design into several smaller dies, and 2) TSVs can potentially shorten the total interconnect length and thus reduce the number of metal layers.

A. Die Area Estimation

At early design stages, the relationship between the die area and the gate count can be roughly described as

$$A_{\text{die}} = N_{\text{gate}} A_{\text{gate}} \tag{1}$$

where N_{gate} is the gate count and A_{gate} is an empirical parameter that shows the proportional relationship between area and gate counts. Based on empirical data from many industrial designs, in this paper, we assume that A_{gate} is equal to $3125\lambda^2$, in which λ is half of the feature size for a specific technology node. Although this area estimation methodology is straightforward and highly simplified, it accords with the real-world measurement quite well.[2]

B. Wire Length Estimation

Besides estimating the die area, it is necessary to estimate the design complexity, which can be represented by the total length of wire interconnects, as well. *Rent's rule* [23] is a well-known and powerful tool that reveals the relationship between the number of signal terminals and the number of internal gates. *Rent's rule* can be expressed as follows:

$$T = kN_{\text{gate}}^{p} \tag{2}$$

where the parameters k and p are Rent's coefficient and exponent and T is the number of signal terminals. Although *Rent's rule* is an empirical result based on the observations of previous designs and it is not proper to use it for nontraditional designs, it does provide a useful framework to compare similar architectures and it is plausible to use it as a part of our 3-D IC cost model.

Based on *Rent's rule*, Donath [24] discovered that it can be used to estimate the average wire length, and Davis *et al.* [25] found it can be used to estimate the wire length distribution in VLSI chips. As a result, in this paper, we use the derivation of *Rent's rule* to predict the wire length distribution function $i(l)$ [25], which has the following forms:

Region I: $1 \le l \le \sqrt{N_{\text{gate}}}$

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

Region II: $\sqrt{N_{\text{gate}}} \le l < 2\sqrt{N_{\text{gate}}}$

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

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 (*f.o.*)of a gate as follows:

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

and Γ is calculated as

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

C. Metal Layer Estimation

After estimating the die area and the wire length distribution, it becomes applicable to further predict the number of metal layers that are required to route all the interconnects within the die area constraint. The number of required metal

[2]Note that it is necessary to update A_{gate} along with the technology advance because the gate length is not strictly linearly scaled.

layers for routing depends on the complexity of the interconnects. A simplified metal layer estimation can be derived from the average wire length [21] as follows:

$$n_{\text{wire}} = \frac{f.o.\bar{R}_m w}{\eta}\sqrt{\frac{N_{\text{gate}}}{A_{\text{die}}}} \tag{7}$$

where $f.o.$ refers to the average gate fanout, w to the wire pitch, η to the utilization efficiency of metal layers, $\bar{R}_m$ to the average wire length, and n_{wire} to the number of required metal layers.

Such a simplified model is based on the assumptions that each metal layer has the same utilization efficiency and the same wire width [21]. However, such assumptions may not be valid in real designs [26]. To improve the estimation of the number of metal layers needed for feasible routing, a more sophisticated metal layer estimation method is derived from the wire length distribution rather than the simplified average wire length estimation. The basic flow of this method is explained as follows.

1) *Estimate the available routing area of each metal layer with the expression*

$$K_i = \frac{A_{\text{die}}\eta_i - 2Av_i\left(N_{\text{gate}} f.o. - I(l_i)\right)}{w_i} \tag{8}$$

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

2) *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 expressed as

$$\chi L(l_i) - \chi L(l_{i-1}) \le K_i \tag{9}$$

where $\chi = 4/(f.o.+3)$ is a factor accounting for the sharing of wires between interconnects in the same net [25], [27]. The function $L(l)$ is the first-order moment of $i(l)$.

3) *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 early design stages. Table I lists the values of all the related parameters [26], [28]. Fig. 2 shows examples of estimating the area and the number of metal layers of 65 nm designs with different scale of gates.

Fig. 2 also shows an important implication for 3-D IC cost reduction. *When a large 2-D chip is partitioned into multiple smaller dies with 3-D integration, each smaller die requires fewer metal layers to satisfy the interconnect routability requirements*. Such metal layer reduction opportunity can potentially offset the cost of extra steps incurred by 3-D integration such as 3-D bonding and testing.

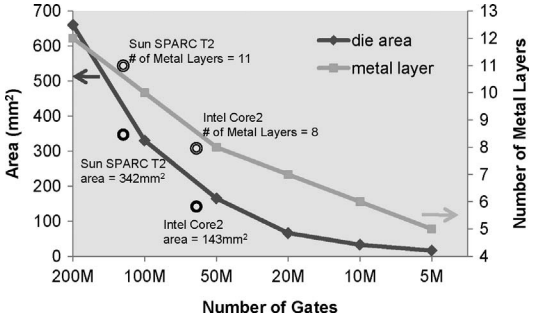

Fig. 2. Examples of the early design estimation of the die area and the metal layer under 65 nm process. This estimation is well correlated with the state-of-the-art microprocessor designs. For example, Sun SPARC T2 [29] contains about 500 million transistors (roughly equivalent to 125 million gates), with an area of 342 mm^2 and 11 metal layers).

TABLE I
PARAMETERS USED IN THE METAL LAYER ESTIMATION MODEL

Rent's exponent for logic, p_{logic}	0.63
Rent's coefficient for logic, k_{logic}	0.14
Rent's exponent for memory, p_{memory}	0.12
Rent's coefficient for memory, k_{memory}	6.00
Average gate fanout, $f.o.$	4
Utilization efficiency of layer 1, η_1	9%
Utilization efficiency of layer 2, η_2	25%
Utilization efficiency of layer 3, η_3	54%
Utilization efficiency of layer 4, η_4	38%
Utilization efficiency of other odd layers, $\eta_{5,7,9,11,\ldots}$	60%
Utilization efficiency of other even layers, $\eta_{6,8,10,12,\ldots}$	42%
Wire pitch of lower metal layers, $w_{i(i<9)}$	$(i+1)F$
Wire pitch of higher metal layers, $w_{i(i\ge 9)}$	$10F$
Blockage area per via on layer i, Av_i	$(w_i+3F)^2$

D. TSV Estimation

The existence of TSVs in the 3-D IC has its effects on the wafer cost as well. The effects are twofold.

1) In 3-D ICs, some global interconnects are now implemented by TSVs, going between stacked dies. This could lead to the reduction of the total wire length and provide opportunities for metal layer reduction in each smaller die.
2) On the other hand, 3-D integration with TSVs may increase the total die area, since the silicon area where TSVs punch through cannot be utilized for building devices or 2-D metal layer connections.[3]

While the first effect is already explained by Fig. 2, showing the existence of TSVs leading to a potential metal layer reduction on the horizontal direction, modeling the second effort needs extra efforts by adding the estimation of the TSV count, and the extra silicon area occupied by TSVs.

To predict the number of required TSVs for a certain partition pattern, a derivation of *Rent's rule* [24] describing the relationship between the interconnect count (X) and the gate count (N_{gate}) can be used. This relationship can be formulated as the following equation:

$$X = \alpha k N_{\text{gate}}\left(1 - N_{\text{gate}}^{p-1}\right) \tag{10}$$

where α is calculated by (5).

[3]Based on current TSV fabrication technologies, the diameter of TSVs ranges from 0.2 μm to 10 μm [3]. In this paper, we use the TSV diameter of 10 μm and assume that the keepout area diameter is 2.5X of TSV diameter, which is 25 μm.

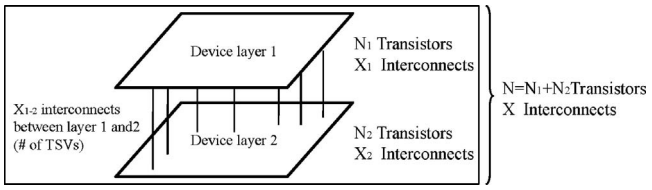

Fig. 3. Basic idea of how to estimate the number of TSVs.

As illustrated in Fig. 3, if a planar 2-D design is partitioned into two separate dies, whose gate counts are N_1 and N_2, respectively, by using (10), the interconnect count on each layer, X_1 and X_2, is given by

$$X_1 = \alpha k_1 N_1 \left(1 - N_1^{p_1 - 1}\right) \tag{11}$$

$$X_2 = \alpha k_2 N_2 \left(1 - N_2^{p_2 - 1}\right). \tag{12}$$

Irrespective of the partition pattern, the total interconnect count of a certain design always keeps constant. Thus, the number of TSVs can be calculated as

$$\begin{aligned} X_{TSV} &= \alpha k_{1,2}(N_1 + N_2)\left(1 - (N_1 + N_2)^{p_{1,2}-1}\right) \\ &\quad - \alpha k_1 N_1 \left(1 - N_1^{p_1 - 1}\right) - \alpha k_2 N_2 \left(1 - N_2^{p_2 - 1}\right). \end{aligned} \tag{13}$$

Here $k_{1,2}$ and $p_{1,2}$ are the equivalent Rent's coefficient and exponent, which are derived as follows [30]:

$$p_{1,2} = \frac{p_1 N_1 + p_2 N_2}{N_1 + N_2} \tag{14}$$

$$k_{1,2} = \left(k_1^{N_1} k_2^{N_2}\right)^{1/(N_1 + N_2)} \tag{15}$$

where N_i is the gate count on layer i.

Therefore, after adding the TSV area overhead, we formulate the new TSV-included area estimation, $A_{3\text{-D}}$, as follows:

$$A_{3\text{-D}} = A_{\text{die}} + N_{\text{TSV/die}} A_{\text{TSV}} \tag{16}$$

where A_{die} is calculated by die area estimator, $N_{\text{TSV/die}}$ is the equivalent number of TSVs on each die, and A_{TSV} is the size of TSVs (including the keepout area). In practice, the final TSV-included die area, $A_{3\text{-D}}$, is used in the wafer cost estimation later, while A_{die} is used in (8) for metal layer estimation since that is the actually die area available for routing.

IV. 3-D Cost Model

Based on the estimation of the die area of each layer, the number of required metal layers, and the number of TSVs across planar dies in the previous section, a complete set of cost models are proposed in this section. Our 3-D IC cost estimation is composed of four separate parts: wafer cost model, 3-D bonding cost model, package cost model, and cooling cost model. An overview of the proposed cost model set is illustrated in Fig. 4.

A. Wafer Cost Model

The wafer cost model estimates the cost of each separate planar die. This cost estimation includes all the cost incurred before the 3-D bonding process. Before estimating the die cost, we first model the wafer cost.

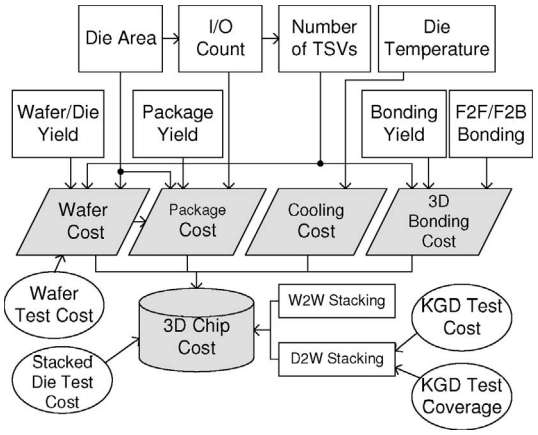

Fig. 4. Overview of the proposed 3-D cost model, which includes four components: wafer cost model, 3-D bonding cost model, package cost model, and cooling cost model.

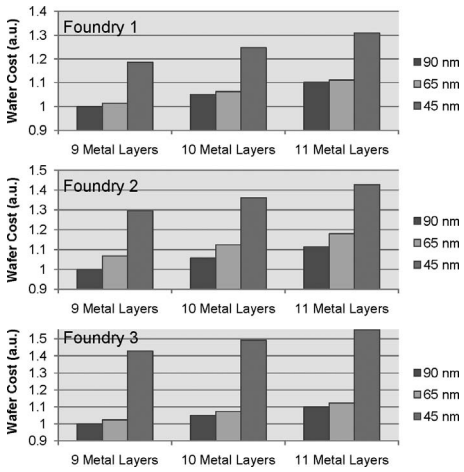

Fig. 5. Batch of data calculated by the wafer cost model. The wafer cost varies from different processes, different number of metal layers, different foundries, and some other factors.

The wafer cost is determined by several parameters, such as fabrication process type (i.e., CMOS logic or DRAM memory), process node (from 180 nm down to 22 nm), wafer diameter (200 mm or 300 mm), and the number of metal layers (polysilicon, aluminum, or copper depending on the fabrication process type). We model the wafer cost by dividing it into two parts: a fixed part that is determined by process type, process node, wafer diameter, and the actual fabrication vendor; a variable part that is mainly dependent on the number of metal layers. We call the fixed part as *silicon cost* and the variable part as *metal cost*. Therefore, the wafer cost is given by

$$C_{\text{wafer}} = C_{\text{silicon}} + N_{\text{metal}} \times C_{\text{metal}} \tag{17}$$

where C_{wafer} is the wafer cost, C_{silicon} is the silicon cost (the fixed part), C_{metal} is the metal cost per layer, and N_{metal} is the number of metal layers. All the parameters used in this model are collected from *IC Knowledge LLC* [31], which considers cost-related factors including material cost, labor cost, foundry margin, number of reticles, cost per reticle, and other miscellaneous cost. Fig. 5 shows the predicted wafer cost of 90 nm, 65 nm, and 45 nm processes, with 9 or 10 layers of metal, for three different foundries.

Besides the number of metal layers affecting the cost as indicated by (17), the die area is another key factor that affects the bare die cost, since smaller die area implies more dies per wafer. The number of dies per wafer is formulated by [32]

$$N_{\text{die}} = \frac{\pi \times (\phi_{\text{wafer}}/2)^2}{A_{\text{die}}} - \frac{\pi \times \phi_{\text{wafer}}}{\sqrt{2 \times A_{\text{die}}}} \tag{18}$$

where N_{die} is the number of dies per wafer, ϕ_{wafer} is the diameter of the wafer, and A_{die} is the die area. After having estimated the wafer cost by (17) and the number of dies per wafer by (18), the bare die cost can be expressed as

$$C_{\text{die}} = \frac{C_{\text{wafer}}}{N_{\text{die}}}. \tag{19}$$

In addition, the die area is also related to the die yield, which later affects the net die cost. By assuming rectangular defect density distribution, the relationship between the die area and the die yield can be given by the following equation [33]:

$$Y_{\text{die}} = Y_{\text{wafer}} \times \frac{\left(1 - e^{-2A_{\text{die}}D_0}\right)}{2A_{\text{die}}D_0} \tag{20}$$

where Y_{die} and Y_{wafer} are the die yield and the wafer yield, respectively, and D_0 is the wafer defect density. Therefore, the net die cost is $C_{\text{die}}/Y_{\text{die}}$.

B. 3-D Bonding Cost Model

The 3-D bonding cost model estimated the cost incurred during the process that integrates several planar 2-D dies together using TSVs. The extra fabrication steps required by 3-D integration consist of TSV forming, wafer/die thinning, and wafer/die bonding.

There are two ways to build 3-D TSVs: *laser drilling* and *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. Furthermore, there are two approaches for TSV etching: 1) *TSV-first approach:* TSVs can be formed during the 2-D 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), and 2) *TSV-last approach:* TSVs can also be formed after the completion of 2-D fabrications, after the BEOL processes. Such an approach is called *TSV-later* approach, and is shown in Fig. 6(b). Either approach has advantages and disadvantages. *TSV-first* approach builds TSVs before building metal layers, thus there is no TSVs punching through metal layers and hence no TSV area overhead for routing; *TSV-last* approach has TSV area overhead for routing, but it isolates the die fabrication from 3-D bonding, which does not need to change the traditional fabrication process. In order to separate the wafer cost model and the 3-D bonding cost model, we assume *TSV-last* approach is used in 3-D IC fabrication. The TSV area overhead caused by *TSV-last* approach is modeled by (16). The data of the 3-D bonding cost including the cost of TSV forming, wafer thinning, and wafer (or die) bonding are obtained from our industry partner, with the assumption that the yield of each 3-D process step is 99%.

Combined with the wafer cost model, the 3-D bonding cost model can be used to estimate the cost of a 3-D-stacked chip with multiple dies. In addition, the entire 3-D-stacked chip cost depends on some design options. For instance, it depends on whether die-to-wafer (D2W) or wafer-to-wafer (W2W) bonding is used, and it also depends on whether face-to-face (F2F) or face-to-back (F2B) bonding is used. If D2W bonding is selected, cost of known-good-die (KGD) test should also be included [9].

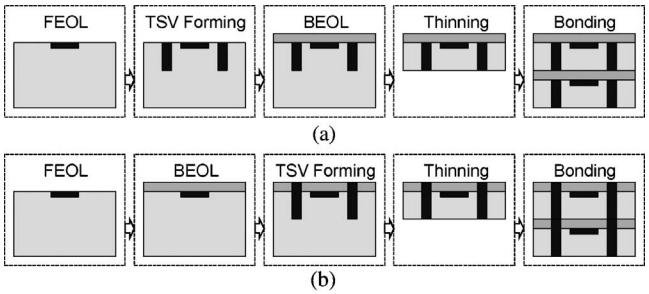

Fig. 6. Fabrication steps for 3-D ICs. (a) TSVs are formed before BEOL process, thus TSVs only punch through the silicon substrate but not the metal layers. (b) TSVs are formed after BEOL process, thus TSVs punch through not only the silicon substrate but the metal layers as well.

For D2W bonding, the cost of a bare N-layer 3-D-stacked chip before packaging is calculated as follows:

$$C_{\text{D2W}} = \frac{\sum_{i=1}^{N}\left(C_{\text{die}_i} + C_{\text{KGDtest}}\right)/Y_{\text{die}_i} + (N-1)C_{\text{bonding}}}{Y_{\text{bonding}}^{N-1}} \tag{21}$$

where C_{KGDtest} is the KGD test cost, which we model as $C_{\text{wafersort}}/N_{\text{die}}$, and the wafer sort cost, $C_{\text{wafersort}}$, is a constant value for a specific process in a specific foundry in our model. Note that the testing cost for 3-D IC itself is a complicated problem. Another study of us has demonstrated a more detailed test cost analysis model with the study of design-for-test (DFT) circuitry and various testing strategies, showing the variants of testing cost estimations [34]. For example, adding extra DFT circuitry to improve the yield of each die in D2W stacking can help the cost reduction but the increased area may increase the cost. In this paper, we adopt the above testing cost assumptions to simplify the total cost estimation, due to space limitation.

For W2W bonding, the cost is calculated as

$$C_{\text{W2W}} = \frac{\sum_{i=1}^{N} C_{\text{die}_i} + (N-1)C_{\text{bonding}}}{\left(\prod_{i=1}^{N} Y_{\text{die}_i}\right) Y_{\text{bonding}}^{N-1}}. \tag{22}$$

In order to support multiple-layer bonding, the default bonding mode is F2B. If F2F mode is used, there is one more component die that does not need the thinning process, and the thinning cost of this die is subtracted from the total cost.

C. Package Cost Model

The package cost model estimates the cost incurred during the packaging process. Package cost can be determined by three factors: package type, package area, and pin count.

As an example, we select the package type to be flip-chip land grid array (fcLGA) in this paper. fcLGA is the common package type of multiprocessors, while other types of packages, such as fcPGA, PGA, pPGA, and so on are also available in our model. Besides package type, package area

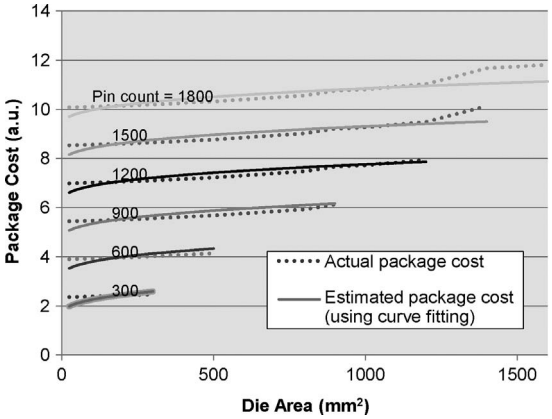

Fig. 7. Package cost depends on both pin count and die area.

and pin count are the other two key factors that determine the package cost. By using the early stage estimation method described in Section III, the package area can be estimated from the die area, $A_{3-\mathrm{D}}$ in (16), and the pin count can be estimated by *Rent's rule* in (2). We analyze the actual package cost data and find that the number of pins becomes the dominant factor to the package cost when the die area is much smaller than the total area of pin pads. It is easy to understand, since there is a base material and production cost per pin which will not be reduced with die area shrinking.

Fig. 7 shows the sample data of package cost obtained from *IC Knowledge LLC* [31]. Based on this set of data, we use curve fitting to derive the following package cost formula with the parameters of the die area and the pin count:

$$C_{\text{package}} = \mu_1 N_p + \mu_2 A_{3-\mathrm{D}}^{\alpha} \tag{23}$$

where N_p is the pin count, $A_{3-\mathrm{D}}$ is the die area. In addition, μ_1, μ_2, and α are the coefficients and exponents, which are 0.00515, 0.141, and 0.35. By observing (23), we can also find that pin count will dominate the package cost if the die area is sufficiently small, since the order of pin count is higher than that of area. Fig. 7 also shows the curve fitting result, which is well aligned to the raw data.

D. Cooling Cost Model

Although the combination of the aforementioned wafer cost model, 3-D bonding cost model, and package cost model can already offer the cost of a 3-D-stacked chip after packaging, we present the cooling cost model, since it has been widely noticed that 3-D-stacked chip has higher working temperature than its 2-D counterpart and it might need more powerful cooling solution that causes extra cost.

Gunther *et al.* [35] noticed that the cooling cost is related to the power dissipation of the system. Furthermore, the system power dissipation is highly related to the chip working temperature [36]. Our cooling cost estimation is based on the peak steady-state temperature of the targeted 3-D-stacked multiprocessor.

There are many types of cooling solutions, ranging from a simple extruded aluminum heat sink to an elaborate vapor phase refrigeration. Depending on the cooling mechanisms, cooling solutions used today can be defined as either convection cooling, phase-change cooling, thermoelectric cooling (TEC), or liquid cooling [37]. Typical convection cooling solutions are heat sinks and fans, which are widely adopted for the microprocessor chips in today's desktop computers. Phase-change cooling solutions, such as heatpipes, might be used in laptop computers. In addition, thermoelectric cooling and liquid cooling are used in some high-end computers. We collect the cost of all the cooling solutions from the commercial market by searching the data from *Digikey* [38] and *Heatsink Factory* [39]. We find that more powerful types of cooling solutions often lead to higher costs as expected. Table II is a list of typical prices of these cooling solutions.

TABLE II
COST OF VARIOUS COOLING SOLUTIONS

Cooling Solution	Unit Price ($)
Heat sink	3–6
Fan	10–40
TEC	15–20
Heatpipe	30–70
Liquid cooling	Above 75

TABLE III
VALUES OF K_c AND c IN (24), WHICH ARE RELATED TO THE CHIP TEMPERATURE

Chip Temperature (°C)	K_c	c
< 60	0.2	−6
60–90	0.4	−16
90–120	0.2	2
120–150	1.6	−170
150–180	2	−230

In our model, we further assume that cooling cost increases linearly with the rise of chip temperature, if the same type of cooling solution is adopted. Based on this assumption and the data listed in Table II, the cooling cost is therefore given by

$$C_{\text{cooling}} = K_c T + c \tag{24}$$

where T is the temperature from which the cooling solution can bring down, K_c and c are the cooling cost parameters, which can be determined by Table III.

Fig. 8 shows the cost of these five types of cooling solutions. It can be observed that the chips with higher steady-state temperatures will require more powerful cooling solutions, which lead to higher costs. It is also illustrated in Fig. 8 that the cooling cost is not a global linear function of the temperature, whereas there are several regions with linear increase of the cost. Each of the regions is correspondent to one type of the cooling solutions.

V. COST EVALUATION FOR FULLY CUSTOMIZED ASICS

The ideal 3-D fabrication gives the freedom that transistors can be built on their optimal layers. In this case, we first use our early design estimation methodology and the proposed cost model to demonstrate how the cost of fully customized ASICs can be reduced by 3-D fabrication. In this section, we use the IBM common platform foundry cost model as an example to perform a series of analysis for logic circuitry.

A. TSV Area Overhead

As mentioned in Section III, building TSVs not only incurs additional process cost but also causes area overhead. To esti-

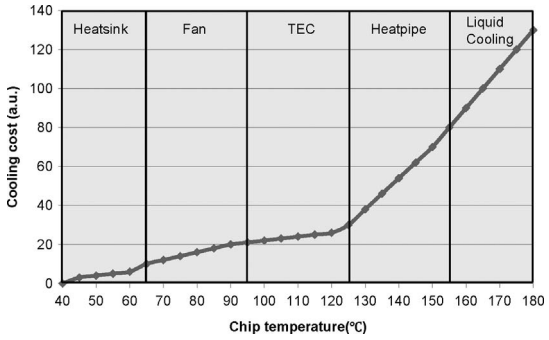

Fig. 8. Plot for the cooling cost model: the cooling cost increases linearly with the chip temperature if the same cooling solution is used; more powerful cooling solutions result in higher costs.

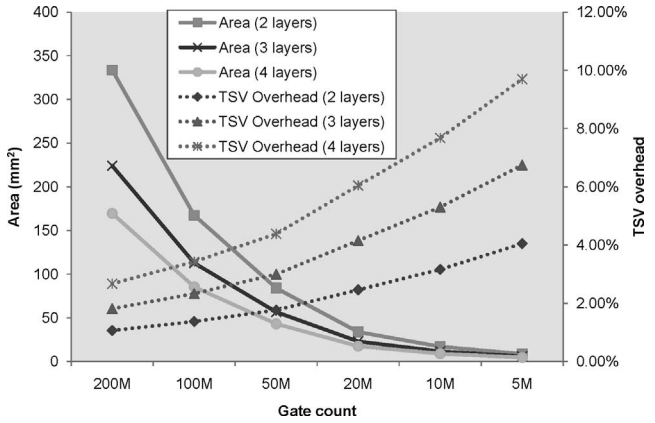

Fig. 9. Total area and the percentage of area occupied by TSVs. For small designs, the TSV area overhead is near to 10%, but for large designs, the TSV area overhead is less than 4%.

mate the TSV area overhead, we set the exponent parameter p and the coefficient parameter k for logic designs [in (2)] to 0.63 and 0.14, respectively, according to Bakoglu's research [28]. We further assume that all the gates are uniformly partitioned into N layers. We choose the TSV size to be $8\,\mu\text{m} \times 8\,\mu\text{m}$. Using the early design estimation methodology described in Section III, the predicted TSV area overhead under 65 nm process is shown in Fig. 9.

Consistent with (16), the TSV area overhead increases as the number of 3-D layers increases. The overhead can reach as high as 10% for a small design (with five million gates) using four layers. However, when the design is sufficiently large (with 200 million gates) and 3-D integration only stacks two dies together, the TSV area overhead can be as low as 2%. To summarize, *for large design, the area overhead due to TSVs is acceptable*.

B. Potential Metal Layer Reduction

As discussed in Section III, the total wire length decreases as the number of 3-D layers increases, since the existence of TSVs removes the necessity of global interconnects. After subtracting the global interconnects implemented by TSVs, we further assume the remaining interconnects are evenly partitioned into N dies. Therefore, the routing complexity on each die is much less than that of the original 2-D design. As a result, it becomes possible to remove one or more metal layers on each smaller die after 3-D partitioning.

TABLE IV
NUMBER OF REQUIRED METAL LAYERS PER DIE (65 NM PROCESS)

Gate Counts (million)	2-D	2-layer	3-layer	4-layer
5	5	5	5	4
10	6	5	5	5
20	7	6	5	5
50	8	7	7	6
100	10	8	7	7
200	12	10	9	8

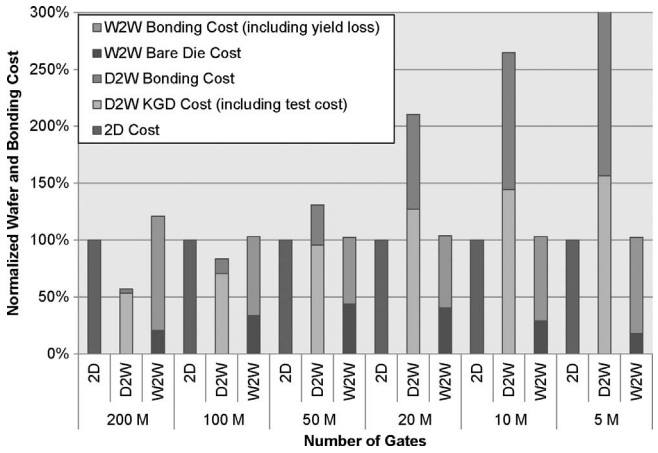

Fig. 10. Cost comparison among 2-D, 2-layer D2W, and 2-layer W2W under 65 nm process with the consideration of TSV area overheads and metal layer reductions.

Using the 3-D routability estimation methodology discussed in Section III, we project the metal layer reduction effect for various design scales ranging from five million gates to 200 million gates when they are partitioned into $1, 2, 3,$ and 4 layers. The estimation of the metal layer reduction is listed in Table IV.

Although the result shows that there are few opportunities to reduce the number of required metal layers in a relatively small design (i.e., with five million gates), the effect of the metal layer reduction becomes more and more obvious with the growth of the design complexity. For instance, the number of required metal layers can be reduced by $2, 3,$ and 4 when a large 2-D design (i.e., with 200 million gates) is uniformly partitioned into $2, 3,$ and 4 separate dies.

To summarize, *in large-scale 3-D IC design, it is possible to use fewer metal layers for each smaller die, compared to the baseline 2-D design.*

C. Bonding Methods: D2W or W2W

D2W and W2W are two different ways to bond multiple dies together in 3-D integration [2]. In Section IV, how to model these two bonding methods is discussed. D2W bonding can achieve a higher yield by introducing KGD test, while W2W bonding waives the necessity of test before bonding at the expense of yield loss [2]. Since both D2W and W2W have their pros and cons, we use our 3-D cost model to find out which one is more cost-effective.

Fig. 10 shows the cost comparison among conventional 2-D processes, 2-layer D2W bonding, and 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 designs, the cost of W2W is always higher than that of 2-D processes in our

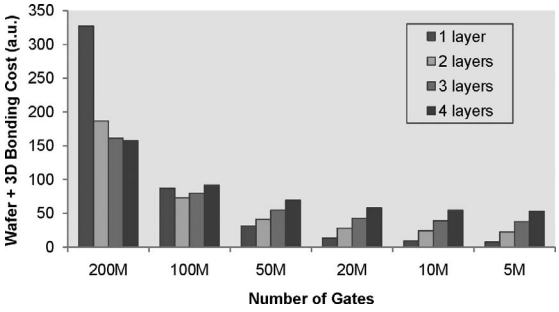

Fig. 11. Cost of 3-D ICs with different number of layers using 65 nm process and D2W bonding with the consideration of TSV area overheads and metal layer reductions.

TABLE V
ENABLING POINT OF 3-D FABRICATION (DATA FROM IBM COMMON PLATFORM IS USED IN THIS EXAMPLE)

Process Node (nm)	Gate Count (million)	Die Area (mm^2)
130	21	277.3
90	40	253.1
65	76	250.9
45	143	226.2

results. This phenomenon can be explained by the relationship between die area and die yield as expressed by (20). 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 is approximate to the product of the yield of each planar die. Thus, the overall yield of W2W-bonded 3-D chips becomes similar to that of 2-D chips. Since the extra bonding cost is inevitable for W2W bonding, but not for 2-D processes, W2W-bonded chips are then more expensive than their 2-D counterparts.

However, while D2W bonding has to pay extra cost for KGD test, it can ensure only good dies are integrated together. Therefore, the yield benefit achieved from partitioning design into multiple smaller dies is reserved. Although it might not offset the KGD test cost for small-size dies whose yields are already high, the yield improvement can potentially offset the bonding cost and the KGD test cost when the original design scale is too large to implement using 2-D processes with high yields. Based on our evaluation result, the D2W bonding starts to show its cost efficiency when the design scale is larger than 100 million gates per chip.

To summarize, *from yield perspective, D2W stacking has cost advantage over W2W stacking, and the cost of D2W-bonded chips can be lower than that of 2-D chips when the design scale is sufficiently large.*

D. Enabling Point of 3-D

It is critical to recognize what is the enabling point beyond which 3-D ICs would be cheaper than their 2-D counterparts. For this purpose, we select the IBM Common Platform 65 nm fabrication technology as our experiment target and compare the cost of the 2-D baseline design with its 3-D counterparts, which have two, three, and four layers. We calculate the cost of bare chips before packaging by using our proposed wafer cost model and 3-D bonding cost model. Fig. 11 shows the cost estimations based on the assumption that the 2-D baseline design is partitioned uniformly.

It can be observed from Fig. 11 that the cost dramatically increases with the growth of the chip size due to the exponential relationship between die sizes and die yields. Because the die yield becomes more sensitive to large die areas, splitting a large-scale 2-D design into multiple dies is more likely to reduce the total cost than splitting a small 2-D design.

Another observation is that the optimal number of 3-D layers in terms of cost might vary depending on how large the design scale is. For example, the most cost-effective way for a large-scale design with 200 million gates is to partition it into four layers. Whereas, for a smaller design with 100 million gates, the most cost-effective option is to use 2-layer partitioning. Finally, when the design scale is only 50 million gates per chip, the conventional 2-D fabrication is always the most cost-effective one, since the 3-D bonding cost is the dominant cost in this case and the yield improvement brought by 3-D partitioning is too small.

Running the same experiment using different fabrication processes ranging from 130 nm down to 45 nm, we achieve the estimation of the boundary that indicates where the 2-layer 3-D stacking starts to be more cost-effective than the baseline 2-D design. The data are listed in Table V. If we convert the number of gates into chip area, the enabling point of 2-layer 3-D process is about 250 mm^2. This is mainly because the yield of fabricating a die larger than 250 mm^2 starts to drop quickly. If the design scale keeps increasing, which means the chip area keeps growing, more 3-D layers are required to achieve the highest cost efficiency.

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

E. Cost-Driven 3-D Design Flow

The 3-D 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 3-D IC design flow becomes possible. Fig. 12 shows a proposed cost-driven 3-D IC design flow. The integration of 3-D IC cost models into design flows guides the designer to optimize their 3-D IC design and eventually to manufacture low-cost product.

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

VI. COST EVALUATION FOR MANY-CORE MICROPROCESSOR DESIGNS

In the previous section, we evaluate the cost benefit of 3-D fabrication for fully customized ASIC designs. However, the

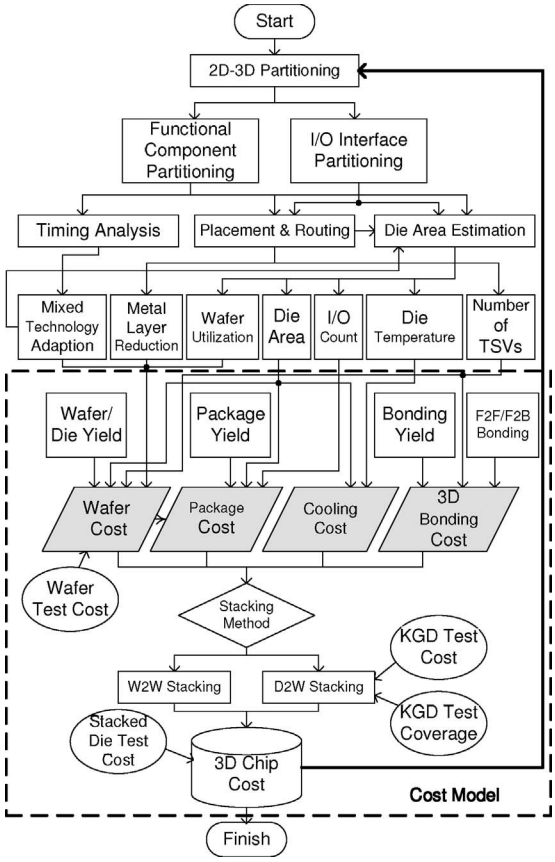

Fig. 12. Scheme of a cost-driven 3-D IC design flow.

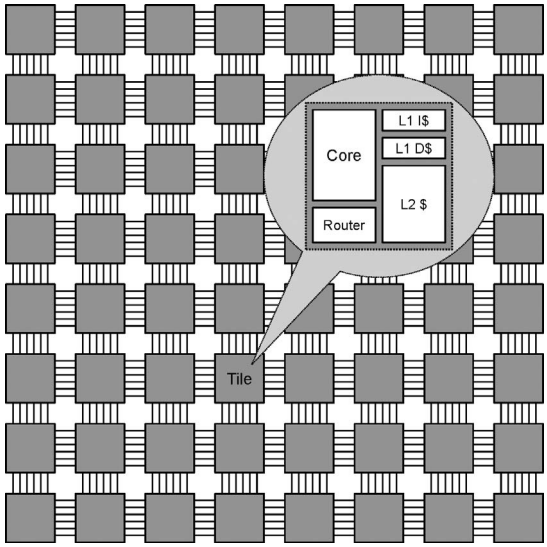

Fig. 13. Structure of the baseline 2-D multiprocessor, which consists of 64 tiles.

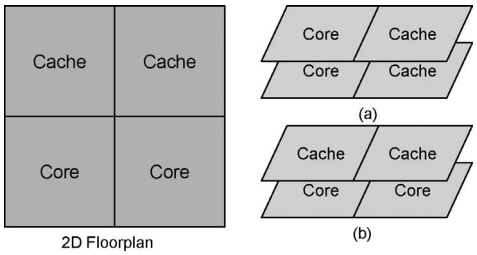

Fig. 14. Three ways to implement a 3-D dual-core microprocessor.

gate-level fine-granularity 3-D partitioning is not feasible in the near future since it requires all the basic modules (such as adders, multipliers, and so on) to be redesigned in a 3-D-partitioned way. Hence, as a near-term goal, the module-level coarse-granularity 3-D partitioning is regarded as the more realistic way for 3-D IC implementation, such as SoC style or many-core microprocessor design style, where each IP block (such each core/memory IP) remains to be a 2-D block and the 3-D partitioning is performed at the block level. In such design style, the Rent's rule-based TSV estimation is not valid, and the TSV estimation would have to be carried out based on system-level specification (such as the signal connections between cores and memory). In this section, we demonstrate how to use the remaining parts of the proposed 3-D cost model to estimate the 3-D many-core microprocessor cost. We use 45 nm IBM common platform cost model in this case study.

A. Baseline 2-D Configuration

In order to have a scalable architecture for many-core microprocessors, the baseline 2-D design used in our paper is an NoC-based architecture, in which the processing tile is the basic building block. Each processing tile is composed of one in-order SPARC-like microprocessor core [29], one L1 cache, one L2 cache, and one router. Fig. 13 shows the example of such architecture with 64 tiles connected by a 8-by-8 mesh structure. Table VI lists the details of cores, L1, L2 caches, and routers, respectively. The gate counts of the core and the router modules are obtained from empirical data of existing designs [29], and the cell count of the cache module by only considering the SRAM cells and ignoring the cost of peripheral circuits. In this case study, we simply assume logic gates and SRAM cells (six-transistor model) occupy the same die area. We consider the resulting estimation error is tolerable since it is an early design stage estimation. Note that we set the memory cell count (from L1 and L2 caches) to be close to the logic gate count (from cores and routers) by tuning cache capacities. This manual tuning is just for the convenience of our later heterogeneous stacking. Generally, there is no constraints on the ratio between the logic and the memory modules.

Fig. 14 illustrates two ways to implement a 3-D dual-core microprocessor, and it can be conceptually extended to the many-core microprocessor case. Generally, the two types of partitioning strategies are: *homogeneous* and *heterogeneous*. As shown in Fig. 14(a), all the layers after *homogeneous* partitioning are identical, while *heterogenous* partitioning leads to core (logic) layers and cache (memory) layers as shown in Fig. 14(b).

B. Cost Evaluation with Homogeneous Partitioning

We first explore the *homogeneous* partitioning strategies, in which all the processing tiles are distributed into $1, 2, 4, 8,$ and 16 layers, respectively, without breaking tiles into finer granularity.

TABLE VI
CONFIGURATION OF THE SPARC-LIKE CORE IN A TILE OF THE BASELINE 2-D MULTIPROCESSOR

Processor Cores	
Clock frequency	1.2 GHz
Architecture type	In-order
INT pipeline	Six stages
FP pipeline	Six stages
ALU count	1
FPU count	1
Empirical gate count	1.7 million
Routers	
Type	4 ports
Empirical gate count	1.0 million
Caches	
Cache line size	64 B
L1 I-cache capacity	32 KB
L1 D-cache capacity	32 KB
L2 cache capacity	256 KB
Empirical cell count	2.6 million

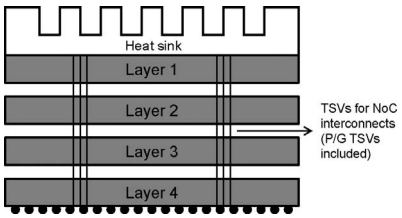

Fig. 15. Conceptual view of a 4-layer homogeneous stacking composed of four identical layers.

Fig. 15 illustrates the conceptual view of a 4-layer homogeneous stacking. The role of TSVs in the homogeneous 3-D-stacked chip is to provide the interlayer NoC connection and to deliver the power from I/O pins to inner layers. Instead of using (13), the TSV count is predicted on the basis of NoC bus width. We assume 3-D mesh is used and the NoC flit size is 128 bits. Therefore, the number of TSVs between two layers, N_{TSV}, is estimated as

$$N_{\mathrm{TSV}} = 2 \times 128 \times N_{\mathrm{tiles/layer}}/(1-\gamma) \tag{25}$$

where $N_{\mathrm{tiles/layer}}$ is the number of tiles per layer, 2 means the bidirectional channel between two tiles, and γ is the proportion of the power/ground pins. For simplicity, we assume γ is 50% in this paper. After calculation, the TSV area overhead is around 3.8%.

In the next step, we use the cost model set to estimate the cost of wafer, 3-D bonding, package, and cooling, respectively. We analyze microprocessor designs of different scales. Figs. 16–19 show the estimated price breakdown of 16-core, 32-core, 64-core, and 128-core microprocessors when they are fabricated by 1-layer, 2-layer, 4-layer, 8-layer, and 16-layer processes, respectively. In these figures, all the prices are divided into five parts (the first bar, which represents 2-D planar process, only has three parts as the 2-D process only has one die and does not need the bonding process). From bottom up, they are the cost of one die, the cost of remaining dies, the bonding cost, the package cost, and the cooling cost.

As the figures illustrate, the cost of one die is descending when the number of layers is ascending. This descending trend is mainly due to the yield improvement brought by the smaller

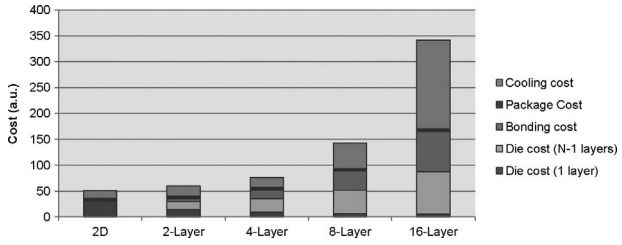

Fig. 16. Cost breakdown of a 16-core microprocessor design with different homogeneous partitioning.

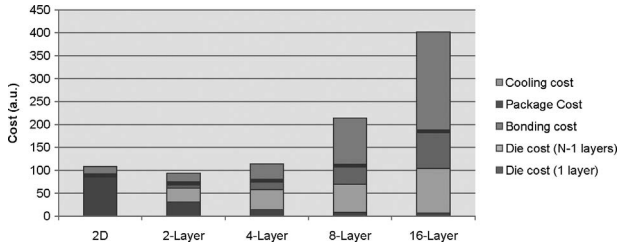

Fig. 17. Cost breakdown of a 32-core microprocessor design with different homogeneous partitioning.

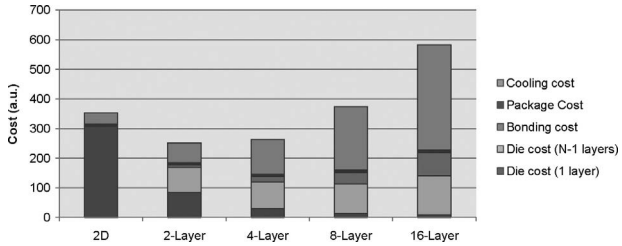

Fig. 18. Cost breakdown of a 64-core microprocessor design with different homogeneous partitioning.

die size. However, this trend is flattened at the end since the yield improvement is not significant any more when the die size is sufficiently small. As a result, the cumulative cost of all the dies does not follow a descending trend. Combined with the 3-D bonding cost, which is an ascending function to the layer count, the wafer cost usually hits its minimum at some middle points. For example, if only considering the wafer cost and the 3-D bonding cost, the optimal number of layers for 16-core, 32-core, 64-core, and 128-core designs are 2, 4, 4, 8, respectively, which is consistent to our rules of thumb discussed in Section V.

However, the cost before packaging is not the final cost. System-wise, the final cost needs to include the package cost and the cooling cost. As mentioned, the package cost is mainly determined by the pin count. Thus, the package cost almost remains constant in all the cases since the pin count does not change from one partitioning to another partitioning. The cooling cost mainly depends on the chip temperature. As more layers are vertically stacked, the higher the chip temperature is, the cooling cost grows with the increase of 3-D layers, and this growth is quite fast because a 3-D-stacked chip with more than 8 layers can easily reach the peak temperature of more than 150 °C. Actually, the extreme cases in this experiment (such as 16-layer stacking) is not practical indeed as the underlying chip is severely overheated. In this experiment, we first obtain

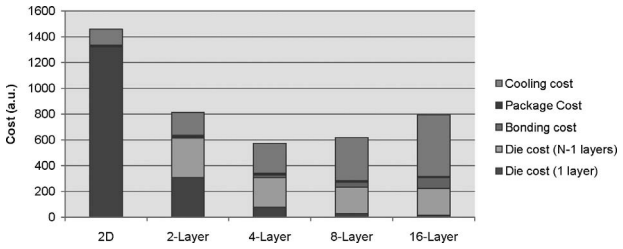

Fig. 19. Cost breakdown of a 128-core microprocessor design with different homogeneous partitioning.

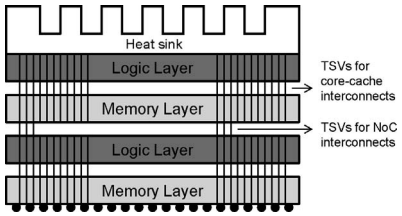

Fig. 20. Conceptual view of a 4-layer heterogeneous stacking composed of 2 logic layers and 2 memory layers.

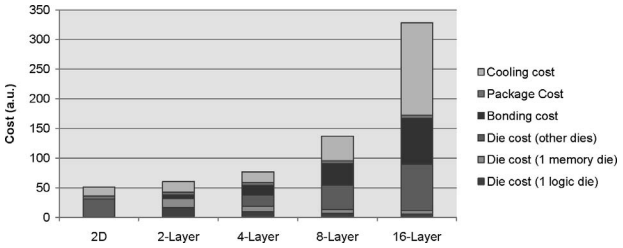

Fig. 21. Cost breakdown of a 16-core microprocessor design with different heterogeneous partitioning.

the power data of each component (core, router, L1 cache, and L2 cache) from a power estimation tool, McPAT [10]. Combined with the area data achieved from early design stage estimation, we get the power density data, then feed them into HotSpot [37], which is a 3-D-aware thermal estimation tool, and finally get the estimated chip temperature and its corresponding cost of cooling solutions. Fig. 16 to Fig. 19 also show the cost of packaging and cooling. As we can find, the cooling cost is just a small portion for few layer stacking (such as 2-layer), but it starts to dominate the total cost for aggressive stacking (such as 8-layer and 16-layer). Therefore, the optimal partitioning solution in terms of the minimum total cost differs from the one in terms of the cost only before packaging. Observed from the result, the optimal number of layers for 16-core, 32-core, 64-core, and 128-core become 1, 2, 2, and 4, respectively instead. This result also reveals our motivation why the package cost and the cooling cost are included in the decision of optimal 3-D stacking layers.

C. Cost Evaluation with Heterogeneous Partitioning

Similar to our baseline design illustrated in Fig. 13, today's high-performance microprocessors have a large portion of the silicon area occupied by on-chip SRAM or DRAM caches. In addition, nonvolatile memory can also be integrated as on-chip memory [7]. However, different fabrication processes are not compatible. For example, the CMOS logic module

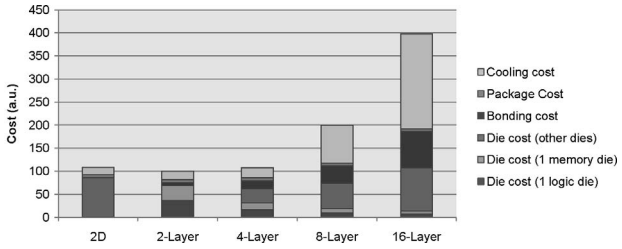

Fig. 22. Cost breakdown of a 32-core microprocessor design with different heterogeneous partitioning.

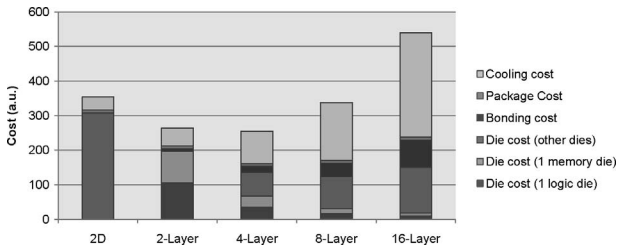

Fig. 23. Cost breakdown of a 64-core microprocessor design with different heterogeneous partitioning.

might require 1-poly-9-copper-1-aluminum interconnect layers, while DRAM memory module needs 7-poly-3-copper and flash memory module needs 4-poly-1-tungsten-2-aluminum. As a result, integrating heterogeneous modules in a planar 2-D chip would dramatically increase the chip cost. Intel shows that heterogeneous integration for large 2-D SoC would boost the chip cost by three times [40].

However, fabricating heterogeneous modules separately could be a cost-effective way for such systems. For example, if the SRAM cache modules are fabricated separately, as per our estimation, the required number of metals is only six while the logic modules (such as cores and routers) usually need more than 10 layers of metals. Hence, We reevaluate the many-core microprocessor cost by using *heterogeneous integration*, in which the tile is broken into logic parts (i.e., core and router) and memory parts (i.e., L1 and L2 caches).

Fig. 20 illustrates a conceptual view of 4-layer heterogeneous partitioning where logic modules (cores and routers) and memory modules (L1 and L2 caches) are on separate dies. As shown in Fig. 20, there are two types of TSVs in the heterogeneous stacking chip: one is for NoC mesh interconnect, which is the same as that in the homogeneous stacking case; the other is for the interconnect between cores and caches, which is caused by the separation of the logic and memory modules. The number of the extra core-to-cache TSVs is calculated as follows:

$$N_{\text{extraTSV}} = \left(\text{Data} + \sum \text{Address}_i\right) \times N_{\text{tiles/layer}} \tag{26}$$

where Data is the cache line size and Address_i is the address width of cache i. Note that the number of tiles per layer, $N_{\text{tiles/layer}}$ is doubled in the heterogeneous partitioning compared to its homogeneous counterpart. For our baseline configuration (Table VI), in which IL1 is 32 KB, DL2 is 32 KB, L2 is 256 KB, and cache line size is 64B, $\text{Data} + \sum \text{Address}_i$ is 542. As a side-effect, the extra core-to-cache TSVs cause the TSV area overhead increasing from 3.8% to 15.7%.

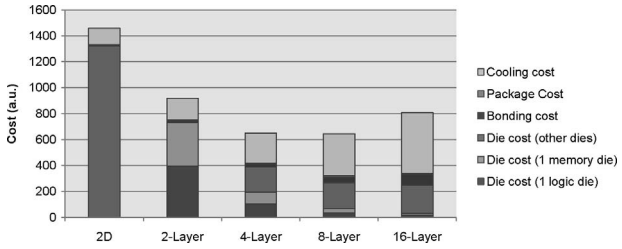

Fig. 24. Cost breakdown of a 128-core microprocessor design with different heterogeneous partitioning.

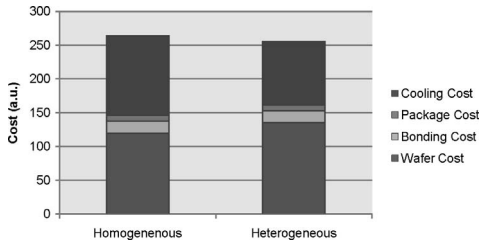

Fig. 25. Cost comparison between homogeneous and heterogeneous integrations for a 64-core microprocessor using 45 nm 4-layer 3-D process.

TABLE VII
OPTIMAL NUMBER OF 3-D LAYERS FOR MANY-CORE MICROPROCESSOR DESIGNS (45 NM)

	Homogeneous	Heterogeneous
16-core	1	1
32-core	2	2
64-core	2	4
128-core	4	8

Compared to its homogeneous counterpart, the advantages and disadvantages of partitioning an NoC-based many-core microprocessor in the heterogeneous way are:

1) fabricating logic and memory dies separately reduces the wafer cost;
2) putting logic dies closer to heat sink reduces the cooling cost;
3) the higher TSV area overhead increases the wafer cost and the package cost.

In order to know whether the heterogenous integration is more cost-effective, we repeat the cost estimation for 16-core, 32-core, 64-core, and 128-core microprocessor designs by using heterogenous partitioning. The price breakdown of heterogeneous stacking is shown in Figs. 21–24. Compared to the homogeneous integration cost as illustrated in Figs. 16–19, we can observe that the cost of heterogeneous integration is higher than that of homogenous integration in most cases, which is mainly due to the increased die area caused by higher TSV counts. However, in some cases, the cost of heterogenous integration can be cheaper compared to its homogeneous counterparts. Fig. 25 demonstrates the cost comparison between homogeneous and heterogeneous integrations for a 64-core microprocessor design using the 45 nm 4-layer 3-D process. It can be observed that although the fabrication efficiency of separated logic and memory dies cannot offset the extra cost caused by the increased die area, the cooling cost is reduced after putting logic layers, which closer to the heat sink (or other cooling solutions).

While our experiment shows heterogeneous integration has the cost advantages on cheaper memory layers and lower chip temperature, in reality, some designers might still prefer homogeneous integration, which has identical layout on each die. The reciprocal design symmetry (RDS) was proposed by Alam *et al.* [41], [42] to reuse one mask set for multiple 3-D-stacked layers. The RDS could relieve the design effort, and thus reduce the design time, which means a cost reduction on human resources. However, this personnel cost is not included in our cost model set, and it is out of the scope of this paper.

We list the optimal partitioning options for 45 nm many-core microprocessor designs in Table VII, which shows how our proposed cost estimation methodology helps decide the 3-D partitioning at the early design stage.

VII. CONCLUSION

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

To facilitate the early design stage cost analysis, we proposed a set of cost models that include wafer cost, 3-D bonding cost, package cost, and cooling cost. Based on the cost analysis, we identified the design opportunities for cost reduction in 3-D ICs, and provided a few design guidelines on cost-effective 3-D IC designs. Our research is complementary to the existing research on 3-D microprocessors that are focused on other design goals, such as performance and power consumption.

ACKNOWLEDGMENT

The authors would like to thank Dr. L. Smith from SEMATECH, Austin, TX, M. Ignatowski and K. Bernstein from IBM, Armonk, NY, Dr. S. Gu from Qualcomm, San Diego, CA, and Dr. P. Marchal from IMEC, Leuven, Belgium, for the valuable discussions and the guidance on this research.

REFERENCES

[1] K. Bernstein, "New dimension in performance: Harnessing 3-D integration technology," *EDA Forum*, vol. 3, no. 2, pp. 8–9, Feb. 2006.
[2] W. R. Davis, J. Wilson, S. Mick, J. Xu, H. Hua, C. Mineo, A. M. Sule, M. Steer, and P. D. Franzon, "Demystifying 3-D ICs: The pros and cons of going vertical," *IEEE Design Test Comput.*, vol. 22, no. 6, pp. 498–510, Jun. 2005.
[3] G. Loh, Y. Xie, and B. Black, "Processor design in 3-D die-stacking technologies," *IEEE Micro*, vol. 27, no. 3, pp. 31–48, Mar. 2007.
[4] Y. Xie, G. H. Loh, B. Black, and K. Bernstein, "Design space exploration for 3-D architectures," *ACM J. Emerging Technol. Comput. Syst.*, vol. 2, no. 2, pp. 65–103, 2006.
[5] C. Ababei, Y. Feng, B. Goplen, M. Hushrav. T. Zhang, K. Bazargan, and S. Sapatnekar, "Placement and routing in 3-D integrated circuits," *IEEE Design Test Comput.*, vol. 22, no. 6, pp. 520–531, Jun. 2005.
[6] S.-M. Jung, J.-H. Jang, W.-S. Cho, J. Moon, K. Kwak, B. Choi, B. Hwang, H. Lim, J. Jeong, J. Kim, and K. Kim, "The revolutionary and truly 3-D $25F^2$ SRAM technology with the smallest S^3 (stacked single-crystal Si) cell, $0.16\mu m^2$, and SSTFT (stacked single-crystal thin film transistor) for ultrahigh density SRAM," in *Proc. Symp. VLSI Technol.*, 2004, pp. 228–229.
[7] X. Dong, X. Wu, G. Sun, Y. Xie, H. Li, and Y. Chen, "Circuit and microarchitecture evaluation of 3-D stacking magnetic RAM (MRAM) as a universal memory replacement," in *Proc. Design Autom. Conf.*, 2008, pp. 554–559.

[8] G. Sun, X. Dong, Y. Xie, J. Li, and Y. Chen, "A novel architecture of the 3-D stacked MRAM L2 cache for CMPs," in *Proc. IEEE Int. Symp. High Perform. Comput. Arch.*, 2009, pp. 239–249.
[9] L. Smith, G. Smith, S. Hosali, and S. Arkalgud, "3-D: It all comes down to cost," in *Proc. RTI Conf. 3-D Arch. Semicond. Pack.*, 2007.
[10] S. Li, J. H. Ahn, R. D. Strong, J. B. Brockman, D. M. Tullsen, and N. P. Jouppi, "McPAT: An integrated power, area, and timing modeling framework for multicore and manycore architectures," in *Proc. Int. Symp. Microarch.*, 2009, pp. 469–480.
[11] T. Kgil, S. D'Souza, A. Saidi, N. Binkert, R. Dreslinski, T. Mudge, S. Reinhardt, and K. Flautner, "PicoServer: Using 3-D stacking technology to enable a compact energy efficient chip multiprocessor," in *Proc. Int. Conf. Arch. Support Program. Lang. Oper. Syst.*, 2006, pp. 117–128.
[12] B. Vaidyanathan, W.-L. Hung, F. Wang, Y. Xie, V. Narayanan, and M. J. Irwin, "Architecting microprocessor components in 3-D design space," in *Proc. Int. Conf. VLSI Design*, 2007, pp. 103–108.
[13] G. L. Loi, B. Agrawal, N. Srivastava, S.-C. Lin, T. Sherwood, and K. Banerjee, "A thermally-aware performance analysis of vertically integrated (3-D) processor-memory hierarchy," in *Proc. Design Autom. Conf.*, 2006, pp. 991–996.
[14] E. Wong and S. K. Lim, "3-D floorplanning with thermal vias," in *Proc. Design, Autom. Test Eur.*, 2006, pp. 878–883.
[15] T. Zhang, Y. Zhan, and S. S. Sapatnekar, "Temperature-aware routing in 3-D ICs," in *Proc. Asia South Pacific Design Autom. Conf.*, 2006, pp. 309–314.
[16] H. Hua, C. Mineo, K. Schoenfliess, A. Sule, S. Melamed, R. Jenkal, and W. R. Davis, "Exploring compromises among timing, power and temperature in 3-D integrated circuits," in *Proc. Des. Autom. Conf.*, 2006, pp. 997–1002.
[17] B. Goplen and S. Sapatnekar, "Placement of thermal vias in 3-D ICs using various thermal objectives," *IEEE Trans. Computer-Aided Design Integr. Circuits Syst.*, vol. 25, no. 4, pp. 692–709, Apr. 2006.
[18] A. Coskun, A. Kahng, and T. S. Rosing, "Temperature and cost-aware design of 3-D multiprocessor architectures," in *Proc. Euromicro Conf. Dig. Syst. Des.*, 2009, pp. 183–190.
[19] C. C. Liu, J.-H. Chen, R. Manohar, and S. Tiwari, "Mapping system-on-chip designs from 2-D to 3-D ICs," in *Proc. Int. Symp. Circuits Syst.*, 2005, pp. 2939–2942.
[20] P. Mercier, S. Singh, K. Iniewski, B. Moore, and P. O'Shea, "Yield and cost modeling for 3-D chip stack technologies," in *Proc. Custom Integr. Circuits Conf.*, 2006, pp. 357–360.
[21] R. Weerasekera, L.-R. Zheng, D. Pamunuwa, and H. Tenhunen, "Extending systems-on-chip to the third dimension: Performance, cost and technological tradeoffs," in *Proc. Int. Conf. Comput.-Aided Des.*, 2007, pp. 212–219.
[22] X. Dong and Y. Xie, "System-level cost analysis and design exploration for three-dimensional integrated circuits (3-D ICs)," in *Proc. Asia South Pac. Des. Autom. Conf.*, 2009, pp. 234–241.
[23] B. S. Landman and R. L. Russo, "On a pin vs. block relationship for partitions of logic graphs," *IEEE Trans. Comput.*, vol. 20, no. 12, pp. 1469–1479, Dec. 1971.
[24] W. E. Donath, "Placement and average interconnection lengths of computer logic," *IEEE Trans. Circuits Syst.*, vol. 26, no. 4, pp. 272–277, Apr. 1979.
[25] J. A. Davis, V. K. De, and J. D. Meindl, "A stochastic wire-length distribution for gigascale integration (GSI). Part I: Derivation and validation," *IEEE Trans. Electron Devices*, vol. 45, no. 3, pp. 580–589, Mar. 1998.
[26] A. B. Kahng, S. Mantik, and D. Stroobandt, "Toward accurate models of achievable routing," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 20, no. 5, pp. 648–659, May 2001.
[27] P. Chong and R. K. Brayton, "Estimating and optimizing routing utilization in DSM design," in *Proc. Workshop Syst.-Level Interconnect Prediction*, 1999, pp. 97–102.
[28] H. B. Bakoglu, *Circuits, Interconnections, and Packaging for VLSI*. Reading, MA: Addison-Wesley, 1990.
[29] M. Tremblay and S. Chaudhry, "A third-generation 65 nm 16-core 32-thread llus 32-scout-thread CMT SPARC(R) processor," in *Proc. Int. Solid State Circuit Conf.*, 2008, pp. 82–83.
[30] P. Zarkesh-Ha, J. Davis, W. Loh, and J. Meindl, "On a pin vs. gate relationship for heterogeneous systems: Heterogeneous Rent's rule," in *Proc. IEEE Custom Integ. Circuits Conf.*, 1998, pp. 93–96.
[31] *IC Cost Model, 2009 Revision 0906.* (2009) [Online]. IC Knowledge LLC, Georgetown, MA. Available: http://www.icknowledge.com/our_products/cost_model.html
[32] J. Rabaey, A. Chandrakasan, and B. Nikolic, *Digital Integrated Circuits*. Englewood Cliffs, NJ: Prentice-Hall, 2003.
[33] B. Murphy, "Cost-size optima of monolithic integrated circuits," *Proc. IEEE*, vol. 52, no. 12, pp. 1537–1545, Dec. 1964.
[34] Y. Chen, D. Niu, X. Dong, Y. Xie, and K. Chakrabarty, "Testing cost analysis in three-dimensional (3-D) integration technology," CSE Dept., Pennsylvania State University, University Park, Tech. Rep., 2010 [Online]. Available: http://www.cse.psu.edu/∼yuanxie/3d.html
[35] S. Gunther, F. Binns, D. M. Carmean, and J. C. Hall, "Managing the impact of increasing microprocessor power consumption," *Intel Technol. J.*, vol. 5, no. 1, pp. 1–9, 2001.
[36] K. Skadron, M. R. Stan, K. Sankaranarayanan, W. Huang, S. Velusamy, and D. Tarjan, "Temperature-aware microarchitecture: Modeling and implementation," *ACM Trans. Arch. Code Optim.*, vol. 1, no. 1, pp. 94–125, 2004.
[37] W. Huang, K. Skadron, S. Gurumurthi, R. J. Ribando, and M. R. Stan, "Differentiating the roles of IR measurement and simulation for power and temperature-aware design," in *Proc. Int. Symp. Perform. Anal. Syst. Softw.*, 2009, pp. 1–10.
[38] Digikey. (2009) [Online]. Available: www.digikey.com
[39] Heatsink Factory. (2009) [Online]. Available: www.heatsinkfactory.com
[40] S. Borkar, "3-D technology: A system perspective," in *Proc. Int. 3D-Syst. Integr. Conf.*, 2008, pp. 1–14.
[41] S. M. Alam, R. E. Jones, S. Pozder, and A. Jain, "Die/wafer stacking with reciprocal design symmetry (RDS) for mask reuse in three-dimensional (3-D) integration technology," in *Proc. Int. Symp. Quality Electron. Design*, 2009, pp. 569–575.
[42] J. Zhao, X. Dong, and Y. Xie, "Cost-aware three-dimensional (3-D) many-core multiprocessor design," in *Proc. Design Autom. Conf.*, 2010, pp. 126–131.

Xiangyu Dong (S'09) received the B.S. degree in electronic engineering from Shanghai Jiao Tong University, Shanghai, China, in 2006. Since 2007, he has been pursuing the Ph.D. degree from the Department of Computer Science and Engineering, Pennsylvania State University, University Park.

He is a Part-Time Research Associate with the Exascale Computing Laboratory, Hewlett-Packard Laboratory, Palo Alto, CA. His current research interests include computer architecture, emerging nonvolatile memory, and 3-D integration technology.

Jishen Zhao received the B.E. and M.E. degrees in information technology and instrumentation from Zhejiang University, Hangzhou, China, in 2003 and 2008, respectively.

She joined the Department of Computer Science and Engineering, Pennsylvania State University, University Park, in 2009. Her current research interests include very large scale integration design, computer architecture, and electronic design automation tools for 3-D integrated circuit design.

Yuan Xie (SM'07) received the B.S. degree in electronic engineering from Tsinghua University, Beijing, China, in 1997, and the M.S. and Ph.D. degrees in electrical engineering from Princeton University, Princeton, NJ, in 1999 and 2002, respectively.

He is currently an Associate Professor with the Department of Computer Science and Engineering, Pennsylvania State University (Penn State), University Park. Before joining Penn State in 2003, he was with the IBM Microelectronic Divisions, Worldwide Design Center, Essex Junction, VT. His current research interests include VLSI design, computer architecture, embedded systems design, and electronic design automation.

Dr. Xie is an Associate Editor for the IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS and has served as a TPC member in many conferences. He is a senior member of the ACM. He was a recipient of the SRC Inventor Recognition Award in 2002 and the U.S. National Science Foundation Faculty Early Career Development (CAREER) Award in 2006.