# Cost-Aware Three-Dimensional (3D) Many-Core Multiprocessor Design

Jishen Zhao, Xiangyu Dong, Yuan Xie
Pennsylvania State University
Computer Science and Engineering Department
University Park, PA, 16802, USA
{juz138, xydong, yuanxie}@cse.psu.edu

## ABSTRACT

The emerging three-dimensional integrated circuit (3D IC) is beneficial for various applications from both area and performance perspectives. While the general trend in processor design has been shifting from multi-core to many-core, questions such as whether 3D integration should be adopted, and how to choose among various design options must be addressed at the early design stage. In order to guide the final design towards a cost-effective direction, system-level cost evaluation is one of the most critical issues to be considered. In this paper, we propose a 3D many-core multiprocessor cost model, which includes wafer, bonding, package, and cooling cost analysis. Using the proposed cost model, we evaluate the optimal partitioning strategies for 16-, 32- and 64-core multiprocessors from the cost point of view.[1]

## Categories and Subject Descriptors

B.7.1 [**Types and Design Styles**]: Microprocessors and microcomputers

## General Terms

Design, Economics

## Keywords

Cost Modeling, Many-core processor design, 3D IC Design

## 1. INTRODUCTION

As an emerging technology, three-dimensional integrated circuit (3D IC) benefits from both integration density and performance improvements. A general trend in processor designs has been from multi-core to many-core processors, which often have tens of cores. 3D many-core multiprocessor design is very attractive, since it potentially combines the benefits from both technologies. However, cost is often the dominant factor for making decisions of whether this combination should be adopted, and how to choose among various design options. Ideally, cost issues should be considered before any design decision is made. As is identified in [21], design choices made within the first 20% of the total design cycle time will ultimately result up to 80% of the final product cost. Consequently, cost analysis at the early design stage is critical throughout the 3D many-core multiprocessor design and fabrication.

While cost can be easily measured after production, it is probably the hardest to predict at the early design stage. The reason is that costs always depend on various factors, such as yield, testing, the impact of design reuse, time-to-market, etc. Due to the limited information at the early design stage, most of the factors are hard to evaluate ahead of time. Thus, rather than generating accurate cost estimation, the proposed cost models target at helping designers make decisions at the early design stage with a rough cost evaluation.

[1]This work was supported in part by NSF grants 0903432, 0905365 and SRC grants.


*DAC 2010,* June 13-18, 2010, Anaheim, California, USA.


## 2. RELATED WORK

Most of the prior research work on 3D IC design has focused on architectures [10, 15, 16, 20], design-automation tools [5, 7, 23, 24], and yield enhancement [18]. There are very few existing works which have considered the cost issues.

Coskun *et al.* [1] and Liu *et al.* [14] presented cost models for 3D Systems-on-Chip (SoCs). Both of their cost models made some assumptions to simplify the die area estimation. The number of through silicon vias (TSVs) per chip was assumed to be a fixed value [1] or was not considered [14]. In addition, the processor core and memory areas were assumed to be equal in [1].

A yield and cost model [17] was developed for 3D-stacked chips with particular emphasis on stacking yield and how wafer yield is affected by vertical interconnections. The authors have mentioned that the cooling cost should be considered for the final system-level cost. However, the paper did not give any further analysis on this issue.

Weerasekera *et al.* [21] described a yield and quantitative cost model for 3D ICs. The cost model featured a detailed analysis of the cost per production step. Package cost was evaluated in this work. A similar detailed package cost model was also described in their later work [22]. However, the package cost model presented in both papers targeted at plastic package, which is not an appropriate package type for future many-core processors.

A system-level cost model for 3D ICs is presented in [3]

with respect to wafer cost and 3D bonding cost. The testing cost throughout the fabrication life cycle was also considered. However, the package cost and the cooling cost, were missing in this model. Both costs are critical to the overall cost: 3D can potentially reduce the package cost by achieving a smaller die area, but increase the cooling cost by bringing in thermal issues. Consequently, it is important to consider both issues in the cost analysis.

In this work, we propose a 3D many-core multiprocessor cost model, which includes wafer, 3D bonding, package, and chip cooling cost analysis. In addition, the overall cost is calculated cumulatively with the consideration of yield and testing cost. The following contributions have been made in this work.

- **A Package Cost Model:** An empirical relationship between the 3D many-core processor package cost and the combination of die area and pin count is presented. In addition, packaging yield and class test cost are also considered in this model.
- **A Cooling Cost Model:** In the proposed cost model, the peak steady state temperature of a target processor determines the selection of a cooling solution, and further determines the cooling cost.
- **Wafer Area Estimation:** We extend the wafer cost model in [3], targeting to the many-core processor case. The major modification is the die area estimation, which is a key issue to the wafer cost estimation.
- **Cost evaluation of 3D many-core processors:** We use the proposed cost model to investigate the cost of 16-, 32-, and 64-core multiprocessors. We also explore the costs of homogeneous and heterogeneous partitioning strategies, and identify the optimal partitioning strategies in terms of cost.

# 3. COST ANALYSIS METHODOLOGY

In this section, we describe the proposed 3D many-core multiprocessor cost model. Our cost model consists of four parts: the wafer cost, the 3D bonding cost, the package cost, and the cooling cost. An overview of the cost model is shown in Figure 1.

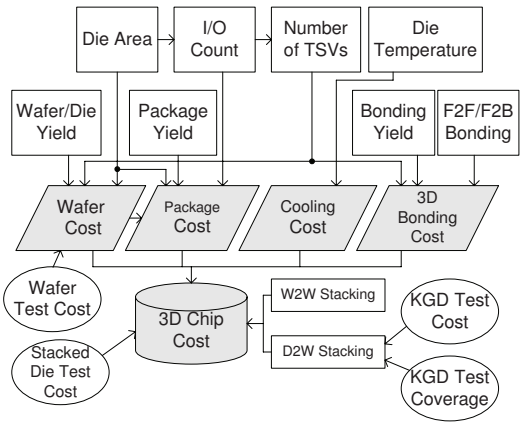


**Figure 1: Overview of the proposed 3D many-core processor cost model.**

## 3.1 Wafer Cost Model

The wafer cost of a many-core processor is calculated using Equation 1.

$$C_{wafer} = N_{layer} C_{layer} \quad (1)$$

where $N_{layer}$ is the number of 3D layers of a many-core processor, $C_{layer}$ is calculated based on the die area of each layer. We extend the wafer cost model described in [3] to many-core processor case. The major difference is the methodology for die area estimation, which is a key issue to the wafer cost analysis. In the 2D case, Sandborn *et al.* [19] estimated the die area as a function of processor core area and bond pad area. In this work, we evaluate the die area by extending Sandborn's work to 3D case.

$$A_{3D} = max(N_p A_p + A_{proc3D}, (P_p \lceil \sqrt{N_p} \rceil)^2) + A_{TSV} \quad (2)$$

In this equation, $A_{proc3D}$ is the area occupied by the processor function units partitioned to each 3D layer, $N_p$ is the number of bond pads on the die (including signal I/Os and power/ground pads), $P_p$ is the minimum center-to-center pitch of bond pads, $A_p$ is the area of a bond pad, $A_{TSV}$ is the TSV area on each die, and $A_{3D}$ is the die area. Each component of the die area is estimated in following steps.

- **Areas of processor function units:** We use McPAT [12, 13] to estimate the total area of processor function units. These function units include the processor core, the L1 instruction and data caches, the L2 cache, and the router for network-on-chip(NOC).
- **Number of signal I/Os and power/ground pads:** We add the processor function units area together, and obtain the gate count $N_g$ using Equation 3.

$$N_g = \frac{A_{processor}}{A_g} \quad (3)$$

  in which $A_{processor}$ is the total area occupied by the processor function units. We assume $A_g = 3125\lambda^2$ [3]. $\lambda$ is the half of the feature size for a specific technology node. The number of signal I/Os can then be determined using Rent's rule [11]. For many-core processors, which often have a large number of gates, we estimate the I/O count based on region II of the Rent's rule, *i.e.*:

$$N_{I/O} = KN_g - K_b N_g^{r_b} - K_s N_g^{r_s} \quad (4)$$

  where $N_g$ is the gate count of the processor, $K$, $K_b$, $K_s$, $r_b$, and $r_s$ are Rent's coefficients and exponents, and $N_{I/O}$ is the number of signal I/Os. The second and the third parts of the equation represent the I/Os saved by buried and shared gates respectively. We also assume that the ratio of the signal I/O count to the power/ground pin count $N_{P/G}$ is 6.0 [19]. The total number of pads on the die can be estimated as:

$$N_p = N_{I/O} + N_{P/G} \quad (5)$$

- **TSV area overhead:** We evaluate the impact of TSVs area overhead to the die area by extending the methodology described in [3]. Since a large number of interconnections can be shared within processor function units, we use region II of the Rent's rule (Equation 4) when calculating the TSV count.

## 3.2 Package Cost Model

The package cost is determined by three factors: the package type, the pin count, and the package area. Ball Grid Array (BGA) and Land Grid Array (LGA) are commonly used in microprocessors. In this study, we only focus on the BGA package type. The die area will also influence the package cost, since it determines the package area. According to our study, the number of pins on a package becomes to dominate the package cost, when the die area is much smaller than the total area of signal I/O and power/ground pads. It is easy to understand, since the base material and production cost per pin will not be reduced despite the reduction of the die area. Figure 2 shows the data for package costs obtained from [9]. Based on the data, we derive an empirical relationship between the package cost and the combination of the die area and the number of pins.

$$C_p = \mu_1 N_p + \mu_2 A_{3D}^{\alpha} \tag{6}$$

In this equation, $\mu_1$, $\mu_2$, and $\alpha$ are the coefficients and exponents, which can be adjusted based on the following constraints: $\alpha$ is between 2 and 3, and $\mu_2$ is much smaller than $\mu_1$. For example, we set $\mu_1$, $\mu_2$ and $\alpha$ to 0.072, 0.0000000398 and 2.7, respectively, in the experiments described in Section 4. By observing Equation 6, we can also find out that the pin count will dominate the package cost, when the die area becomes small enough.

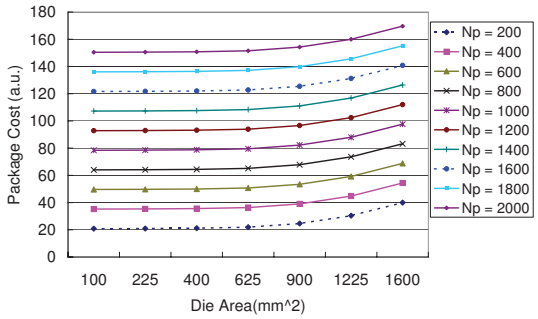


**Figure 2: The package cost depends on both pin count and die area.**

## 3.3 Cumulative Cost Model

We use the methodology described in [3] to estimate the 3D bonding cost. We also obtain additional costs throughout the 3D fabrication life cycle, including material cost, labor cost, foundry margin, wafer testing cost, number of reticles per mask, cost per reticle, and other miscellaneous costs [9]. The total cost is evaluated cumulatively with wafer, package, 3D bonding, and other additional costs listed above [19]. We define $F_{pass}$ as the fraction of dies that passes each testing step, *i.e.*:

$$F_{pass} = Y_d^{Cov_t} \tag{7}$$

in which $Y_d$ is the die yield on the wafer, and $Cov_t$ is the test coverage. The cumulative cost can be calculated recursively by the following equations.

$$C_{cumulative} = \frac{C_{preCumulative} + C_{thisStep}}{F_{pass}} \tag{8}$$

$$C_{preCumulative} = C_{cumulative} \tag{9}$$

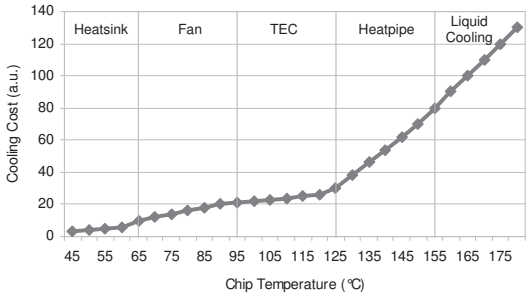


**Figure 3: A plot for the proposed cooling cost model.**

where $C_{preCumulative}$ is the accumulated cost up to the previous step, $C_{thisStep}$ is the cost of the present step, and $C_{cumulative}$ is the resulting cumulative cost in each step.

## 3.4 Cooling Cost Model

Cooling cost is determined by the cooling solutions used for a processor. It is related to the thermal dissipation of the system [6]. The cooling cost model proposed in this work is based on the peak steady state temperature estimation. By using McPAT [12, 13], we can obtain power data of each function units respectively. We then feed the power data into Hotspot 5.0 [8] to estimate the temperature of the 3D many-core processor.

There are many types of cooling solutions, ranging from a simple extruded aluminum heatsink to an elaborate vapor phase refrigeration. We study the average costs of commonly used five types of cooling solutions: heatsink, fan, thermoelectric cooling (TEC), heatpiple and liquid cooling [2, 4]. We find out that more powerful types of cooling solutions often lead to higher costs. In our cooling cost model, we assume that cooling cost increases linearly with the rise of peak steady state temperature of a processor, when the same type of cooling solution is applied. The cooling cost is therefore estimated by the following equation.

$$C_{cool} = K_c t + c \tag{10}$$

in which $K_c$ and $c$ are the cooling cost parameters. They are derived from the average cost for each type of cooling solutions, and can be determined by Table 1.

**Table 1: The values of $K_c$ and $c$ in Equation 10. They are related to the peak steady state temperature achieved by a processor.**

| 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 | $-230$ |

Figure 3 shows the cooling costs of the five types of cooling solutions according to the proposed cooling cost model. It is illustrated in the figure that the chips with higher steady state temperatures will require more powerful cooling solutions, which lead to higher costs. It is also illustrated

that the overall cooling cost is not a linear function of the temperature, whereas there are several regions with linearly increasing cooling cost. Each of the region is correspondent to a type of cooling solutions.

We sum up all the costs evaluated above, and calculate the final cost to fabricate a 3D many-core processor. The final cost estimation is given by Equation 11.

$$C_{final} = C_{cumulative} + C_{cool} \quad (11)$$

in which $C_{cumulative}$ and $C_{cool}$ have been estimated previously, and $C_{final}$ is the resulting final cost.

## 3.5 Cost-Driven 3D EDA Flow

The proposed cost model can be integrated into 3D EDA flows to help designers make decisions at the early design stage. Such decisions could be (1) whether the design will go for 3D, (2) what kind of 3D partition strategies will be used and (3) how many layers there will be for the 3D design.

We propose to integrate the wafer, bonding, package, and cooling cost models into a 3D EDA flow similar to [3]. Overview of such a 3D EDA flow is shown in Figure 4. Note that there is a basic settings module. We take the basic settings as an input to the cost model. Users may adjust the parameters of the cost model according to the technologies and package types, or other specific requirements for their design.

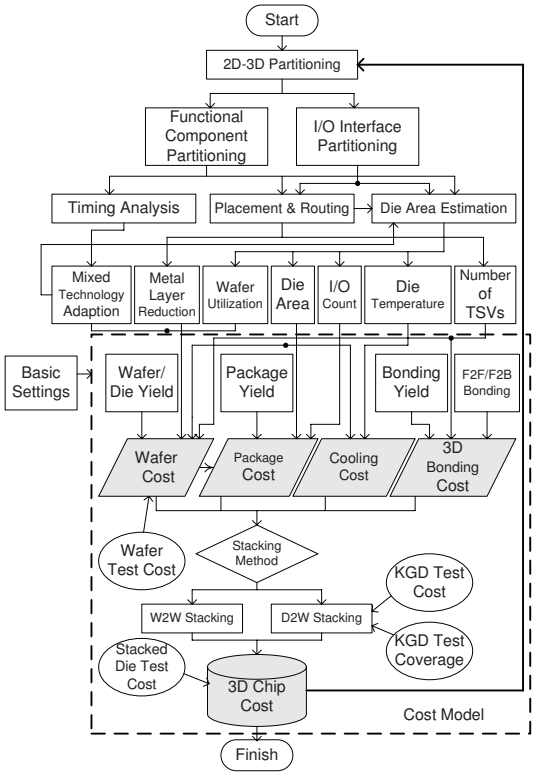


**Figure 4: Integrating the proposed cost model with 3D EDA flow for cost-aware 3D many-core multiprocessor design.**

# 4. EXPERIMENTAL RESULTS

Based on the proposed cost model, we investigate the costs of 3D many-core processors with various partitioning strategies, and compare them against a baseline 2D many-core processor.

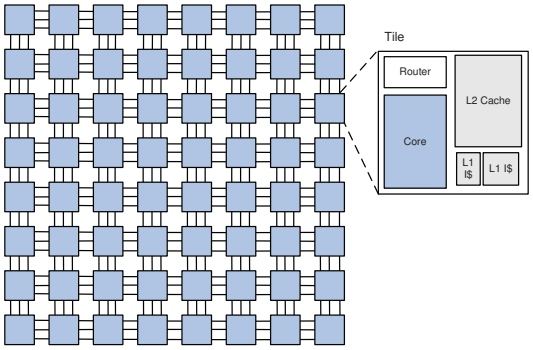


**Figure 5: Configuration of a baseline 64-core multiprocessor.**

**Table 2: The configuration of the SPARC-like core and the caches in a tile of a baseline 2D many-core processor.**

| | |
|---|---|
| **Clock Rate** | $1.4GHz$ |
| **Type** | In-order |
| **Integer Pipeline** | $12 stages$ |
| **Floating Point Pipeline** | $12 stages$ |
| **ALUs per Core** | 3 |
| **FPUs per Core** | 1 |
| **L1 Instruction Cache Size** | $8KB$ |
| **L1 Data Cache Size** | $16KB$ |
| **L2 Cache Size** | $64KB$ |

## 4.1 Baseline 2D Configuration

In our experiments, we adopt 16-, 32-, and 64-core multiprocessors with IBM Common Platform 65nm model as baseline. Figure 5 illustrates the configuration of such a many-core processor.

A baseline processor is composed of multiple tiles. Each tile consists of a SPARC-like core with L1 caches, a 64KB L2 cache, and a 4-port router. The configuration of the Sparc-like core and the caches are described in Table 2. There are two different strategies to partition the baseline processor into multiple layers. We evaluate the costs of the two strategies by partitioning the baseline processors into 1, 2, 4, 8, and 16 3D layers.

## 4.2 Homogeneous Partitioning

With homogeneous partitioning, a tile is kept as a unit. Each 3D layer consists of the same number of tiles. Figure 6 shows the cost break-down of a 32-core processors. It is important to observe that 3D partitioning can always reduce the package costs, since the die area is reduced. However, 3D partitioning will not always reduce the wafer cost and bonding cost, as a result of the significantly increased 3D bonding cost. 3D partitioning leads to higher cooling cost. In fact, when there are more than 8 layers, the peak steady state temperature is over 180 °C, and the cooling cost becomes extremely high. Experiments with 16- and 64-core processors also support these observations. Another interesting observation from our experiment is that TSV area

overhead seems not to be a big concern with such a large number of cores. This can be shown in Figure 7. Although TSVs already occupy more than 10% of the die area when the processors are partitioned to 8 layers, the die areas still shrink significantly.

Note that we have not consider the testing cost and yield to this state. Taking the cumulative cost into account, the final costs are illustrated in Figure 8. According to the estimation, we can define the optimal number of layers with minimum cost for 16-, 32- and 64-core processors. The results are shown in Table 3.

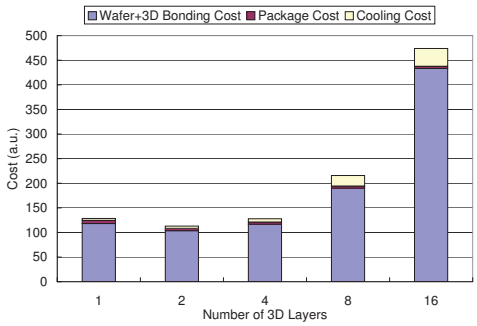


**Figure 6: Cost break-down of a 32-core processor with homogeneous partitioning strategy.**

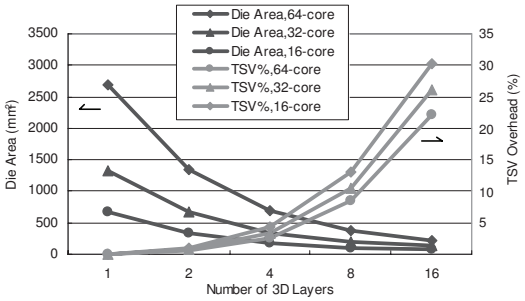


**Figure 7: Analyzing the impact of TSV area overhead with homogeneous partitioning strategy.**

## 4.3 Heterogeneous Partitioning

Heterogeneous partitioning is implemented by breaking a tile into a core, L1 caches, an L2 cache, and a router, and then put them into different layers. For example, in a 4-layer processor, we place all the cores, L1 caches, and routers in one layer, and all the L2 caches in the other three layers. We further divide each layer into 2 layers, and this results in an 8-layer heterogeneous partitioning.

The cost break-down of the 4-layer heterogeneous partitioning is shown in Figure 9. Figure 11 compares the final costs with homogeneous and heterogeneous partitioning strategies. It is shown that heterogeneous partitioning results in higher wafer and 3D bonding costs, compared to the homogeneous partitioning with the same number of 3D layers. The higher wafer and 3D bonding costs also lead to higher final costs. As illustrated in Figure 10, this is due to larger number of TSVs resulted from placing the components of a tile in different layers. Partitioning the many-core processors into other number of layers also support these observations.

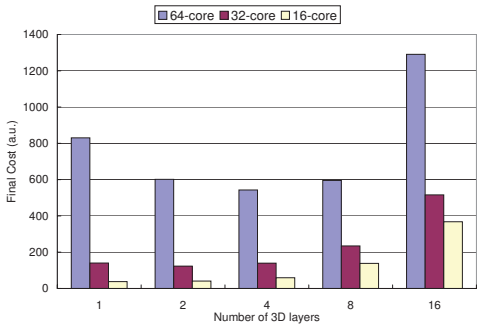


**Figure 8: Final cost of 16-, 32- and 64-core multiprocessors with homogeneous partitioning strategy.**

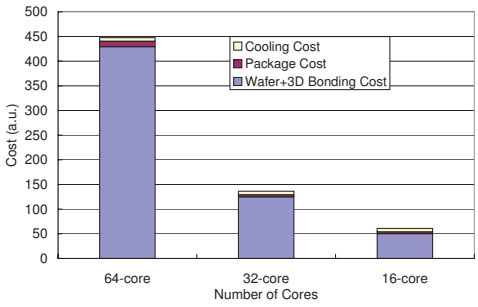


**Figure 9: Cost break-down of 16-, 32- and 64-core multiprocessors with 4-layer heterogeneous partitioning strategy.**

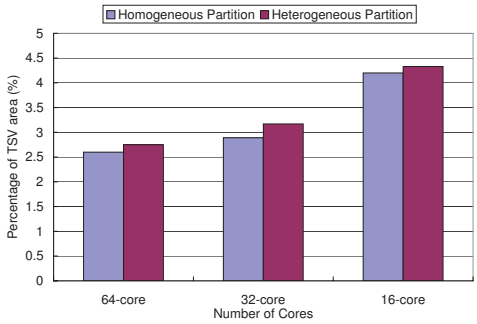


**Figure 10: Compare the TSV area of 16-, 32- and 64-core multiprocessors using homogeneous and heterogeneous partitioning strategies.**

# 5. CONCLUSION

Cost analysis of 3D many-core processor at the early design stage is critical for the decision making among the various design choices.

To facilitate the decision making and guide the 3D design to a cost-effective direction, we propose a 3D many-core processor cost model. Based on the proposed cost model, we compare the estimated cost of 3D many-core processors

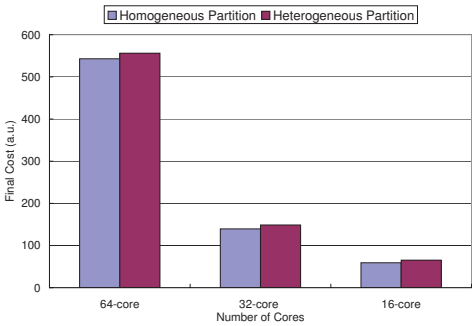


**Figure 11: Compare the final costs of 16-, 32- and 64-core processors using homogeneous and heterogeneous partitioning strategies. All are partitioned into 4 3D layers.**

against the baseline 2D processor. We show that some 3D partitioning strategies result in much lower cost than 2D. In addition, we investigate the cost of various partitioning strategies for 16-, 32- and 64-core 3D multiprocessors. Table 3 lists the optimal number of 3D layers for many-core processors in terms of cost.

**Table 3: Experiment results: the optimal partitioning strategy and the number of 3D layers for 16-, 32- and 64-core multiprocessors.**

| Number of cores | Partitioning Strategy | Number of Layers |
|---|---|---|
| 16 | – | 1 |
| 32 | *Homogeneous* | 2 |
| 64 | *Homogeneous* | 4 |

# 6. REFERENCES


[1] A. Coskun, A. Kahng, and T. S. Rosing. Temperature- and cost-aware design of 3D multiprocessor architectures. In *Proceedings of the Euromicro Conference on Digital System Design, 2009.*, 2009.

[2] Digikey. www.digikey.com. 2009.

[3] X. Dong and Y. Xie. System-level cost analysis and design exploration for three-dimensional integrated circuits (3D ICs). In *Proceedings of the Asia and South Pacific Design Automation Conference, 2009.*, pages 234–241, 2009.

[4] H. Factory. www.heatsinkfactory.com. 2009.

[5] B. Goplen and S. Sapatnekar. Placement of thermal vias in 3-D ICs using various thermal objectives. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 25(4):692–709, 2006.

[6] S. Gunther, F. Binns, D. M. Carmean, and J. C. Hall. Managing the impact of increasing microprocessor power consumption. *Intel Technology Journal*, 5(1):1–9, 2001.

[7] H. Hua, C. Mineo, K. Schoenfliess, A. Sule, S. Melamed, R. Jenkal, and W. R. Davis. Exploring compromises among timing, power and temperature in three-dimensional integrated circuits. In *Proceedings of the Design Automation Conference, 2006.*, pages 997–1002, 2006.

[8] 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 *Proceedings of the International Symposium on Performance Analysis of Systems and Software, 2009.*, pages 1–10, 2009.

[9] IC Knowledge LLC. IC Cost Model, 2009 Revision 0906. 2009.

[10] T. Kgil, S. D'Souza, A. Saidi, N. Binkert, R. Dreslinski, T. Mudge, S. Reinhardt, and K. Flautner. PicoServer: using 3D stacking technology to enable a compact energy efficient chip multiprocessor. In *Proceedings of the International Conference on Architectural Support for Programming Languages and Operating Systems, 2006.*, pages 117–128, 2006.

[11] B. S. Landman and R. L. Russo. On a pin versus block relationship for partitions of logic graphs. *IEEE Transactions on Computers*, 20(12):1469–1479, 1971.

[12] S. Li, J. H. Ahn, R. D. Strong, J. B. Brockman, D. M. Tullsen, and N. P. Jouppi. McPAT 1.0: an integrated power, area, and timing modeling framework for multicore architectures. *HP Labs*, 2009.

[13] 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 *Proceedings of the International Symposium on Microarchitecture, 2009.*, 2009.

[14] C. C. Liu, J.-H. Chen, R. Manohar, and S. Tiwari. Mapping system-on-chip designs from 2-D to 3-D ICs. In *Proceedings of the International Symposium on Circuits and Systems, 2005.*, pages 2939–2942, 2005.

[15] G. Loh, Y. Xie, and B. Black. Processor design in three-dimensional die-stacking technologies. *IEEE Micro*, 27(3):31–48, 2007.

[16] 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 *Proceedings of the Design Automation Conference, 2006.*, pages 991–996, 2006.

[17] P. Mercier, S. Singh, K. Iniewski, B. Moore, , and P. O'Shea. Yield and cost modeling for 3D chip stack technologies. In *Proceedings of the Custom Integrated Circuits Conference, 2006.*, pages 357–360, 2006.

[18] S. Peng and R. Manohar. Yield enhancement of asynchronous logic circuits through 3-dimensional integration technology. In *Proceedings of the Great Lakes Symposium on VLSI, 2006.*, pages 159–164, 2006.

[19] P. A. Sandborn, M. S. Abadir, and C. F. Murphy. The tradeoff between peripheral and area array bonding of components in multichip modules. *IEEE Transactions on Components, Packaging, and Manufacturing Technology, Part A*, 17(2):249–256, 1994.

[20] B. Vaidyanathan, W.-L. Hung, F. Wang, Y. Xie, V. Narayanan, and M. J. Irwin. Architecting microprocessor components in 3D design space. In *Proceedings of the International Conference on VLSI Design, 2007*, pages 103–108, 2007.

[21] R. Weerasekera, D. Pamunuwa, L.-R. Zheng, and H. Tenhunen. Extending systems-on-chip to the third dimension: performance, cost and technological tradeoffs. In *Proceedings of the International Conference on Computer-Aided Design, 2007.*, pages 212–219, 2007.

[22] R. Weerasekera, D. Pamunuwa, L.-R. Zheng, and H. Tenhunen. Two-dimensional and three-dimensional integration of heterogeneous electronic systems under cost, performance, and technological constraints. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 28(8):1237–1250, 2009.

[23] E. Wong and S. K. Lim. 3D floorplanning with thermal vias. In *Proceedings of the Design, Automation and Test in Europe, 2006.*, pages 878–883, 2006.

[24] T. Zhang, Y. Zhan, and S. S. Sapatnekar. Temperature-aware routing in 3D ICs. In *Proceedings of the Asia and South Pacific Design Automation Conference, 2006.*, pages 309–314, 2006.