# Hetero$^2$ 3D Integration: A Scheme for Optimizing Efficiency/Cost of Chip Multiprocessors

Shivam Priyadarshi*, Niket K. Choudhary, Brandon Dwiel, Ankita Upreti, Eric Rotenberg, Rhett Davis, Paul Franzon
Department of Electrical and Computer Engineering
North Carolina State University, Raleigh, North Carolina, 27695
* Email: spriyad@ncsu.edu

***Abstract*—Timing the transition of a processor design to a new technology poses a provocative tradeoff. On the one hand, transitioning as early as possible offers a significant competitive advantage, by bringing improved designs to market early. On the other hand, an aggressive strategy may prove to be unprofitable, due to the low manufacturing yield of a technology that has not had time to mature. We propose exploiting two complementary forms of heterogeneity to profitably exploit an immature technology for Chip Multiprocessors (CMP). First, 3D integration facilitates a technology alloy. The CMP is split across two dies, one fabricated in the old technology and the other in the new technology. The alloy derives benefit from the new technology while limiting cost exposure. Second, to compensate for lower efficiency of old-technology cores, we exploit application and microarchitectural heterogeneity: applications which gain less from technology scaling are scheduled on old-technology cores, moreover, these cores are retuned to optimize this class of application. For a defect density ratio of 200 between 45nm and 65nm, Hetero$^2$ 3D gives 3.6× and 1.5× higher efficiency/cost compared to 2D and 3D homogeneous implementations, respectively, with only 6.5% degradation in efficiency. We also present a sensitivity analysis by sweeping the defect density ratio. The analysis reveals the defect density break-even points, where homogeneous 2D and 3D designs in 45nm achieve the same efficiency/cost as Hetero$^2$ 3D, marking significant points in the maturing of the technology.**

***Keywords*—3DIC, Heterogeneous microarchitecture**

## I. Introduction

The microprocessor industry continues to rely on the scaling benefits of CMOS technology. Each new generation of CMOS technology delivers smaller, faster, and more energy-efficient transistors, which computer architects translate into higher system-level performance/Watt. It is highly desirable to exploit a new technology node as early as possible. Doing so means higher performing systems can be brought to market sooner for a competitive advantage. Unfortunately, a new technology takes time to mature: manufacturing yields are poor early on and improve over time. Consequently, exploiting a new technology prematurely may not be profitable due to exorbitant cost.

In this paper, we propose exploiting two forms of heterogeneity – technology heterogeneity and microarchitectural heterogeneity – to enable extracting most of the performance/Watt benefit from a new technology, before it fully matures, while maintaining profitability.

1. *3D-enabled technology heterogeneity*: This work proposes to split the cores of a Chip Multiprocessor (CMP) across two dies, one fabricated in the old technology and the other in the new technology, with half the cores on each die. By combining technologies, the CMP derives some benefit from the new technology while limiting cost exposure.

2. *Microarchitectural heterogeneity*: The technology alloy means that half the cores do not benefit from the new technology. To compensate, we exploit application and microarchitectural diversity. First, we differentiate between applications that derive the most performance/Watt benefit from the new technology and those that derive the least performance/Watt benefit, relative to the old technology. Most notably, computation-intensive applications tend to benefit more from the new technology than do memory-intensive applications. Scheduling non-memory-intensive jobs on new-technology cores and memory-intensive jobs on old-technology cores partially compensates for the latter cores' handicap. Second, we can compensate even further by retuning the microarchitecture of the old-technology cores to perform better on memory-intensive applications. In particular, increasing the core's dynamic scheduling window size (issue queue, load/store queues, physical register file, and reorder buffer) and narrowing its superscalar fetch/issue widths, yields more memory-latency tolerance and exploits more memory-level parallelism, with comparable circuit complexity (larger window but narrower pipeline). The resulting increase in instructions-per-cycle (IPC) compensates for the lower frequency of the old-technology cores relative to the new-technology cores.

We call our solution Hetero$^2$ 3D Integration, in reference to its two complementary forms of heterogeneity. Crucially, in Hetero$^2$ 3D, core heterogeneity has a strong connection to technology heterogeneity. While core heterogeneity is a well-established idea [1], the selection of core designs depends on the objective. Our objective is novel – compensating for the old-technology tier's handicap – leading to the novel formulation of classifying applications as benefiting more or less from the new-technology tier, and exploiting this workload split to adjust the old-technology tier's core design. Because core heterogeneity is compensating for technology heterogeneity, the two are intertwined: technology heterogeneity influences the core heterogeneity. Moreover, core heterogeneity is modest by virtue of this confined objective. The microarchitecture is

Niket K. Choudhary and Ankita Upreti contributed to this paper when they were graduate students in the Department of Electrical and Computer Engineering at North Carolina State University. Currently they are with Qualcomm CPU Design Center, Raleigh, North Carolina, 27617.

retuned, as opposed to architecting a heterogeneous multi-core from scratch [2].

Hetero$^2$ 3D depends on 3D integration itself becoming routine, hence, cost-effective, both in terms of yield and nonrecurring engineering (NRE) cost of designing a 3D IC. There is evidence that 3D processes are maturing. For example, through-silicon-via (TSV) enabled 3D CMOS image sensors are already in mass production [3], [4] and stacked DRAMs are in the product sampling phase [5]. The progress of CAD for 2D IC design has been remarkable, and CAD for 3D IC design will likely follow a similar trajectory.

Exploring the technology transition problem requires both computation-efficiency and cost metrics. For computation-efficiency – "efficiency" for short – we use BIPS$^3$/W to balance performance and power. We consider a spectrum of multiprogrammed workloads with different mixtures of computation-intensive and memory-intensive threads. For cost, we develop relative-cost models that account for not only die area but also defect-limited yield as a function of technology maturity (time). The models also account for cooling cost because of changes in thermal characteristics with 3D integration. Reporting cost in dollars is not possible, nor is it necessary. Instead, we report cost in relative-cost-units (rcu). Since designs may differ in both efficiency and cost, it is also convenient to compare efficiency/cost: BIPS$^3$/W/rcu.

We evaluate the following four-core CMP designs: a) 2D implementation of four homogeneous cores in 45nm, b) two-tier 3D implementation of the same design (two cores per tier), c) two-tier 3D implementation where cores on different tiers have the same microarchitecture but different technology (45nm and 65nm), d) two-tier 3D implementation where cores on different tiers have different microarchitecture and technology, i.e., Hetero$^2$ 3D Integration. For a defect density ratio of 200 between 45nm and 65nm, Hetero$^2$ 3D gives 3.6× and 1.5× higher efficiency/cost compared to the 2D and 3D homogeneous implementations, respectively, with only 6.5% degradation in efficiency. Our results also highlight the defect density break-even points, where homogeneous 2D and 3D designs in 45nm achieve the same efficiency/cost as Hetero$^2$ 3D, marking significant points in the maturing of the technology.

The paper is organized as follows. Section II presents related work in this area. The relative cost modeling approach is described in Section III. The CMP design space exploration methodology is presented in Section IV. The results are discussed in Section V. Section VI concludes the paper.

## II. Related Work

3D ICs have been widely explored by researchers for integrating different functional layers such as memory, digital logic, radio frequency and analog logic, MEMS, and optoelectronics in a single monolithic 3D die [6]. These functionalities can be manufactured in individually optimized fabrication processes. This work mainly focuses on digital CMOS processes. Madan et al. [7] presented a technique which uses heterogeneous integration of CMOS technologies for improving the reliability of processors. They propose to stack a checker core fabricated in an older process on top of the leading core to exploit the reliability benefits of the matured process. Work presented in this paper uses heterogeneous CMOS process integration for cost reduction, providing orthogonal benefits to those of Madan et al. [7].

Dong and Xie [8] presented a detailed cost model for 3D ICs and also suggested fabricating non-critical components of the die in a slower CMOS technology for cost reduction. However, this work does not suggest how to compensate for the loss in performance of using the slower technology. This paper proposes fabricating some of the component cores of a CMP, in their entirety, in an older technology (i.e., we are not splitting logic within a core across different tiers). Furthermore, we propose and evaluate a strategy to compensate for the losses of using the older technology. An extensive design space exploration considering efficiency/cost as the optimization metric is performed to find the suitable cores which can be fabricated in the older technology.

Weerasekera et al. [9] discussed the cost-effectiveness of a 3D IC fabricated in 65nm compared to the equivalent 2D implementation in matured 45nm. They argue that NRE cost (referring here to the cost of process development, fab set-up, etc.) increases as technology shifts to advanced nodes and implementing a system in the existing technology in 3D adds no additional NRE cost, hence, 3D implementation of the system may be more cost-effective up to several million units. The work presented in this paper assumes that the advanced process node is already in development, hence, its NRE cost is unavoidable. We propose a technique to exploit the scaling benefits of the advanced process node in a cost-effective manner.

## III. Cost Modeling

This section describes the yield and cost model used in this work. The cost model mainly includes die and cooling cost.

### A. Yield model

There are primarily two yield loss mechanisms: a) defect-limited yield loss and b) parametric yield loss. Defect-limited yield loss lead to functional failure. The defects are mainly introduced due to imprecise equipment calibration, material, and contamination impurities in the environment or with human contact. These defects can cause faults such as open or short circuits leading to functional failure. Parametric yield loss is due to dies failing to meet the specified electrical characteristics. This type of yield loss is mainly due to inter and intra-die process variations, resulting in variations in device parameters such as channel length, gate oxide thickness and threshold voltage. In this work we are considering only the defect-limited yield loss which has an inverse relationship with area and defect density. There are various defect-limited yield models such as the Poisson model, Murphy model and Binomial model [10]. The basic philosophy behind these models is the same. The differences in these models are attributed to different defect density distribution used in calculating the yield. We use the Binomial model, as this is supported by ITRS [11]. Based on this model, defect-limited yield can be calculated as:

$$Y = \left[1 + \frac{AD}{\alpha}\right]^{-\alpha} \quad (1)$$

In (1), $A$ is the area, $D$ is the defect density and $\alpha$ is the cluster parameter. The defect density for a given chip area and process is proprietary information and generally kept confidential by

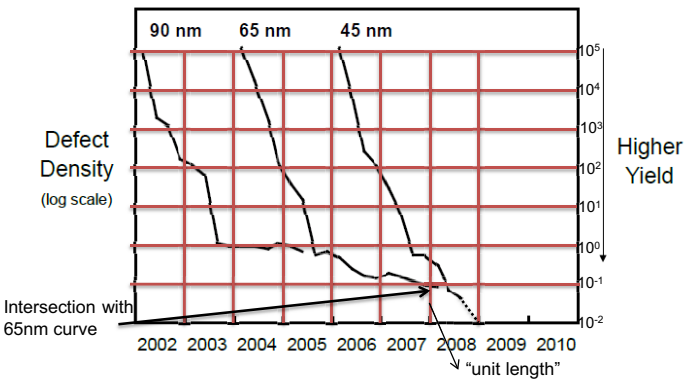


Fig. 1. Defect density trend for different Intel CMOS process nodes [12].

manufacturing vendors. ITRS can be one source of this data but it provides values of defect density assuming the process is matured. It does not provide the evolution of defect density with time. After doing a literature search we found the defect density trend of Intel processes as shown in Figure 1 [12]. The figure shows that the defects are usually very high during the initial years of inception of a new technology. The original plot from Intel has shown just a trend without any data points. We have used this trend and mapped a data point (assuming technology has matured) from the ITRS on this plot. Using that data point, we have calculated other data points by overlaying a uniform grid over it. By combining Intel trend with ITRS data (saturating point) we calculated defect densities in 2007 as follows:

1. For demonstrating our concept, we considered two CMOS technologies namely: 65nm and 45nm. Based on ITRS [11] data, we assumed that finally the defect density on 65nm will saturate to 0.1395 $defects/cm^2$ in 2008.
2. For the year 2008, we drew a vertical line from the X-axis and marked the point of intersection with 65nm curve. The vertical distance of that intersection point from X-axis is considered as unit length. The plot is given on logarithmic scale and unit length approximately comes out to be 1.
3. Based on the unit length measured with a ruler, the right hand Y-axis is marked with values as shown in Fig. 1.
4. Using the values on the Y-axis, the defect density in 2007 is approximated as: $D_{65}$ = 0.25 $defects/cm^2$and $D_{45}$ = 100 $defects/cm^2$.

Point 4 shows that the defect density of 45nm is pretty high in 2007 compared to 65nm. We take these values as starting point in our experiments. To address possible extrapolation errors, we further sweep the defect density (sensitivity analysis) of 45nm to show its impact on efficiency/cost. We also show a crossover point from where the use of 65nm is no longer cost effective and technique presented in this work is not applicable. The reason for choosing 45nm was the availability of an RTL-based processor infrastructure, FabScalar [13] around this technology node for experimentation. The yield argument presented here by taking an example of 65nm and 45nm holds also true for current advance CMOS process nodes such as 28nm and 22nm.

### B. Die Cost Model

The cost of a 2D die can be expressed as

$$C_{die} = \frac{C_{wafer}}{N_{die} * Y_{wafer} * Y_{die}} \tag{2}$$

In (2), $C_{wafer}$ is the cost of a wafer, $N_{die}$ is number of dies, $Y_{wafer}$ is yield of wafer, and $Y_{die}$ is yield of a single die. $N_{die}$ can be calculated as [15]

$$N_{die} = \frac{\pi * \left(\phi_{wafer}/2\right)^2}{A_{die}} - \frac{\pi * \phi_{wafer}}{\sqrt{2 * A_{die}}} \tag{3}$$

In (3), $A_{die}$ is area of a single die and $\phi_{wafer}$ is diameter of wafer taken as 300mm in all the experiments in this work. The second term in equation (3) accounts for the area wasted on the edges of wafer where a full die cannot fit. The equations (2) and (3) show that the area of die ($A_{die}$) impacts overall cost on two levels. Number of dies ($N_{die}$) and the yield of a die ($Y_{die}$) both are the function of area. During the initial years of a new technology defect density is usually high which overshadows the benefits of their smaller area on cost. Hence using an older technology is beneficial even if they have higher die area. However, as defect density of a new technology goes down, the downside of using an older technology for its alternative gets visible on cost.

In this work, Through Silicon Via (TSV) based 3D integration is considered. TSV-enabled 3DIC has the potential to reduce the cost compared to conventional 2D-CMPs because it provides the flexibility of stacking older matured technologies in a monolithic 3D die whose manufacturing and design cost is less. Here the die-to-wafer (D2W) bonding is considered because it allows stacking of dies with different areas which is necessary for integrating different CMOS process nodes. Cost of a $N$-tier 3D die can be calculated as

$$C_{3D} = \frac{\sum_{i=1}^{N} C_{2D_i} + C_{bonding}}{\prod_{i=1}^{N-1} Y_{TSV_i}} \tag{4}$$

where $C_{2D_i}$ is cost of a 2D die fabricated on tier-$i$ which can be calculated using (2). $Y_{TSV_i}$ is manufacturing yield of TSVs fabricated between tier-$i$ and tier-$(i+1)$. $C_{bonding}$ is 3D bonding cost formulated as [9]

$$C_{3D} = \sum_{i=2}^{N} \frac{C_{3Dprocess_i}}{N_{die_i}} \tag{5}$$

In (5), $C_{3Dprocess}$ represents 3D process cost which is taken as $0.2\times C_{wafer}$(cost of wafer) [9]. We have assumed $TSV-last$ approach for TSV fabrication because it isolates the fabrication of individual tiers from 3D bonding. However, this approach has an additional TSV area overhead because it consumes some of the metal routing tracks. To model this overhead, we have included the area of TSVs in 2D-die area while calculating their yield using (1) and $N_{die}$ using (3). The value of the cluster parameter $\alpha$ is taken as 2 in all the experiments. We choose TSV of diameter 5$\mu$m and depth 10$\mu$m whose capacitance is 28.3$f$F. The number of TSVs are determined based on the bus width required for communication between tiers and power and ground routing. For TSV-based

3DIC to be cost effective the manufacturing yield of TSVs should be high. Researchers in [16] have shown fabrication techniques achieving 100% TSV yield. Hence we assumed TSV manufacturing yield to be 100%.

### C. Cooling Cost Model

The increased volumetric density in 3DICs leads to large heat fluxes and the 3D stack increases thermal resistances relative to that of conventional 2DICs. This results in higher on-chip temperature compared to its 2D counterpart. This necessitate the need for modeling the cooling cost. We have used cooling cost model proposed in [15]. Assuming only one type of cooling solution is adopted for all the temperature ranges, the cooling cost is calculated as [15]

$$C_{cooling} = K_c T + c \tag{6}$$

where T is the temperature, $K_c$ and $c$ are cooling cost parameters taken from [15] depending upon the chip temperature. We have used stack technology provided in FreePDK3D45 [17] which is an open-source design kit compiler for stacked dies. In our experiments different tiers of 3D stack are connected face-to-back. The Pathfinder 3D [17] toolset is used for calculating the temperature rise which takes floor plan, power and technology information as input. For wafer and stack technologies, we have used materials provided in the design kit. A heat sink based cooling solution with natural convection is assumed in thermal simulations. The convection resistance is taken as 1.3 $K/W$.

## IV. Methodology

We have used the processor simulator, frequency, area, and power models provided by FabScalar toolset for our experiments [13] [14]. The FabScalar toolset provides the capability to generate synthesizable RTL designs of arbitrary superscalar processor configurations within a particular template. The frequency, area, and power model are based on the detailed RTL designs. We have considered 65nm process node, with defect density of 0.25 $defects/cm^2$, as a matured technology and 45nm process node, with continuously improving defect density, as an immature technology.
The rest of this section describes our design space and outlines methodology for creating workload mix and metrics used for the study.

### A. Design Space

A design space of 12 microarchitecturally diverse out-of-order (OoO) cores using FabScalar toolset is created for experimentation. Core names and their configurations are presented in Table I. It also includes their frequency, area, and peak power on the 45nm process. The fetch width is equal to issue width and the size of load-store queue is same as issue queue size. All the caches use a uniform line size of 64B. The core types are not trained for any specific application and they represent configurations required to target diverse instruction-level parallelism (ILP). The superscalar width is fundamental to the core complexity and determine maximum achievable instruction throughput for an application. We choose the superscalar width to be two, three, four and five. For each width a small, medium and large ILP-extracting structures, e.g. issue-queue and ROB, are considered to target different forms of ILP: *near*, *average*, and *far*. The structure sizes are determined by constraining a superscalar width for three different clock frequencies. For example, C0, C1 and C2 represent three 2-wide OoO cores with decreasing clock frequency and increasing structure complexity. The clock frequency of cores range from 2GHz (for cores C0 and C3) to 1.25GHz (for cores C8 and C11). Choudhary et al. [13] proposed a similar approach to create a workload-agnostic heterogeneous CMP. On average, the clock frequency of a core reduces by 0.7×, area increases by 2×, and power increases by 1.4× on 65nm compared to 45nm.

TABLE I. Microarchitecture Design Space. Core Configuration (Issue-width, Issue-queue, ROB, L1-ICache(KB), L1-DCache(KB), Pipeline Depth), Frequency(GHz), Area($mm^2$), and Peak Power(W)

| Core Name | Core Configuration | Freq. | Area | Power |
|---|---|---|---|---|
| C0 | 2, 32, 96, 16, 16, 16 | 2.00 | 1.48 | 1.96 |
| C1 | 2, 48, 192, 32, 32, 14 | 1.66 | 1.89 | 1.91 |
| C2 | 2, 64, 384, 64, 64, 13 | 1.42 | 2.53 | 1.91 |
| C3 | 3, 16, 64, 16, 16, 18 | 2.00 | 1.44 | 2.22 |
| C4 | 3, 48, 128, 32, 32, 14 | 1.66 | 1.88 | 2.26 |
| C5 | 3, 64, 384, 64, 64, 15 | 1.42 | 2.64 | 2.57 |
| C6 | 4, 32, 128, 32, 32, 16 | 1.66 | 1.93 | 2.84 |
| C7 | 4, 48, 192, 64, 64, 15 | 1.42 | 2.69 | 3.12 |
| C8 | 4, 64, 384, 64, 64, 15 | 1.25 | 2.83 | 2.36 |
| C9 | 5, 24, 64, 32, 32, 16 | 1.66 | 1.99 | 3.01 |
| C10 | 5, 48, 192, 64, 64, 16 | 1.42 | 2.73 | 3.57 |
| C11 | 5, 64, 384, 64, 32, 16 | 1.25 | 2.63 | 3.73 |

We have explored four different CMP configurations shown in Table II. Each CMP has four cores where each core has a 2MB of private L2 cache. We intentionally don't consider a last-level shared cache to simplify the simulation and to remain focussed in evaluating our proposal i.e. heterogenity in technology and core-microarchitecture. Moreover, prior work have shown that the 3D integration can be used to improve bandwidth and efficiency of memory sub-system [18] [19]. CMP-1 is the baseline 2D design on 45nm node. All the cores on CMP-1 have homogeneous microarchitecture. CMP-2 is a 2-tier 3D implementation of CMP-1 where each tier has two cores and their private L2 cache (both tiers at 45nm). CMP-3 and CMP-4 are also 2-tier 3D system. Tier one and two, in CMP-3 and CMP-4 are assumed to be fabricated on 45nm and 65nm node, respectively. The microarchitecture configurations of inter and intra-tier cores in CMP-3 are same as cores of CMP-1. Finally, in CMP-4 cores across the tiers have heterogeneous microachitecture and it represents Hetero$^2$ 3D integration. We assumed two cores on a tier have homogenous microarchitecture to simplify the design space exploration. Allowing heterogeneity among the cores on a tier would further improve the overall energy effciency. We have considered face-to-back bonding of tier one and two where tier two is near the heatsink. A bus of 512 TSVs for inter-tier routing of signal and power lines is used. For simplicity TSVs are equally divided between power and signal lines.

### B. Workloads and Metrics

The integer and floating-point benchmarks from SPEC-2000 are used to evaluate our proposal. The SimPoint tool [20] was used to generate up to four or five 10 million instruction SimPoints from each integer or floating-point benchmark,

TABLE II. CMP CONFIGURATIONS. HOU: HOMOGENEOUS MICROARCHITECTURE, HEU: HETEROGENEOUS MICROARCHITECTURE, HOT: HOMOGENEOUS TECHNOLOGY, AND HET: HETEROGENEOUS TECHNOLOGY

| CMP Name | CMP Type |
|---|---|
| CMP-1 | 2D, HoU, HoT |
| CMP-2 | 3D, HoU, HoT |
| CMP-3 | 3D, HoU, HeT |
| CMP-4 | 3D, HeU, HeT |

TABLE III. WORKLOAD MIX SUMMARY

| Category | Workload Description |
|---|---|
| CI4-MI0 | four CI and zero MI benchamrks |
| CI3-MI1 | three CI and one MI benchamrks |
| CI2-MI2 | two CI and two MI benchamrks |
| CI1-MI3 | one CI and three MI benchamrks |
| CI0-MI4 | zero CI and four MI benchamrks |

respectively. In all, there are 59 SimPoints in the experiments, which we refer to as benchmarks from now on. Further, we have classified benchmarks in compute-intensive (CI) and memory-intensive (MI) groups. Benchmarks in the CI group typically have a large percentage of low-latency instructions e.g simple arithmetic, logical and branch instructions. Moreover, they are characterized by short dependence chains and high branch misprediction in the program. Primarily integer benchmarks, e.g. gcc.473, gzip.779, mcf.2018, and twolf.8155, fall in this group [21]. Benchmarks in the MI group typically have a large percentage of long-latency instructions e.g. floating-point arithmetic and loads missing in level-1 cache. Moreover, they are characterized by long dependence chains and low branch misprediction in the program. Primarily floating-point benchmarks, e.g. ammp.2945, equake.2796, swim.1582, and mgrid.3657, fall in this group [21]. Benchmarks are randomly selected from each category to form a four-threaded workload. To gain insights in our evaluation, we classify these workloads into five categories depending on how many *CI* benchmarks and how many *MI* benchmarks are present in the workload. Table 3 describes this classification. A random sample of 1000 workloads are created in each category.

The *Hmean-fairness* metric [22] is used to quantify the aggregate billions-of-instructions-per-second (BIPS) of a CMP. The Hmean-fairness metric balances throughput and fairness of multiprogram workloads running on CMPs. Our approach for quantifying aggregate BIPS for a workload category is as follows. We calculate harmonic-mean BIPS of the threads in a four-threaded workload for all the samples in a category. Further, we calculate harmonic-mean of all samples' BIPS to get aggregate BIPS. The average power consumption of a workload category is measured by averaging total energy consumed by all the samples over total time to execute all the samples. For 3D implementation we also consider energy consumed by TSVs in case of L2 misses on tier-1. Our metric for computation-efficiency is $BIPS^3/W$ (inverse of energy-delay$^2$). We have assumed ideal benchmark-to-core mapping i.e. the best performing core for a benchmark is known *a priori*.

## V. RESULTS AND DISCUSSION

A design space exploration is performed to select the most efficient core on 45nm across the five workload categories presented in Section 4.2. This creates our baseline configuration (i.e. CMP-1). We have not considered cost in this exploration because our goal was to find the most efficient core assuming 45nm technology is matured. This allows us to truly model the degradation in efficiency due to introducing 65nm for cost advantage. The $BIPS^3/W$ of each core in the design space is calculated for five different workload categories as described in Section 4.2. Further, we take the average of $BIPS^3/W$ of a core across all categories. Core C6 (see Table I) yields the best efficiency. CMP-2 is constructed by creating a two tier 3D structure where each tier has two C6 cores and their private L2 cache fabricated on 45nm. This is the simplest 3D implementation of CMP-1 that has potential to reduce the overall cost. CMP-3 is also a two tier 3DIC where one tier (tier-2) consists two C6 cores and their private L2 caches fabricated on 65nm and another tier (tier-1) consists the same cores but fabricated on 45nm. The introduction of 65nm in CMP-3 brings cost savings but reduces the efficiency compared to CMP-1 and CMP-2 because 65nm cores are slower, more power consuming and bigger compared to their 45nm counterpart. To this end, we propose to use cores with heterogenous microarchitecture across the tiers to compensate the losses in efficiency by exploiting workload diversity. With this aim of creating a Hetero$^2$ 3D integration, another set of design space exploration is performed for constructing CMP-4. In this exploration we have fixed the cores on tier-1 (core C6 on 45nm) and cores on tier-2 were allowed to vary among twelve 65nm cores. Here cost is also taken into account and we have explored a quad-core having maximum $BIPS^3/W/rcu$. The exploration yields in a quad-core having two C5 cores on 65nm and two C6 cores on 45nm. In the later parts of this Section we will discuss the more insights of this result.

Figure 2 presents cost sensitivity analysis which shows how the cost of different CMPs is varying with the ratio of defect densities on 45nm and 65nm. We assumed 65nm defect density to be 0.25 $defects/cm^2$ and swept the defect density of 45nm. The starting defect density of 45nm is taken as 100 $defects/cm^2$ (see Section 3.1) which represents an immature 45nm technology. Here the total cost is the sum of die and cooling cost. Figure 2 shows that for the defect density ratio of 400, CMP-1 is approximately 3$\times$ more expensive than CMP-2 and 5.4$\times$ more expensive than CMP-3 and CMP-4. At this ratio, microprocessor vendors are constrained by cost in building a 2D CMP-1 system even if they want to exploit the scaling benefits of 45nm. CMP-2 gives them one way to introduce 45nm, where cost is reduced due to a smaller footprint facilitated by 3D integration. We propose the cost can be further reduced by 1.8$\times$ by constructing CMP-3 and CMP-4 which allows to get the cost benefit of 65nm. Note that even if 65nm cores are almost twice the area of 45nm, they are cost beneficial. This is because their yield is much better compared to 45nm. As the 45nm technology matures and the defect density ratio goes down, the cost benefits of CMP-3 and 4 starts diminishing. At the ratio of 4, CMP-1 is the cheapest and the scheme presented in this paper is no longer beneficial. However, it provides a way to exploit the scaling benefits of 45nm in a cost effective manner without

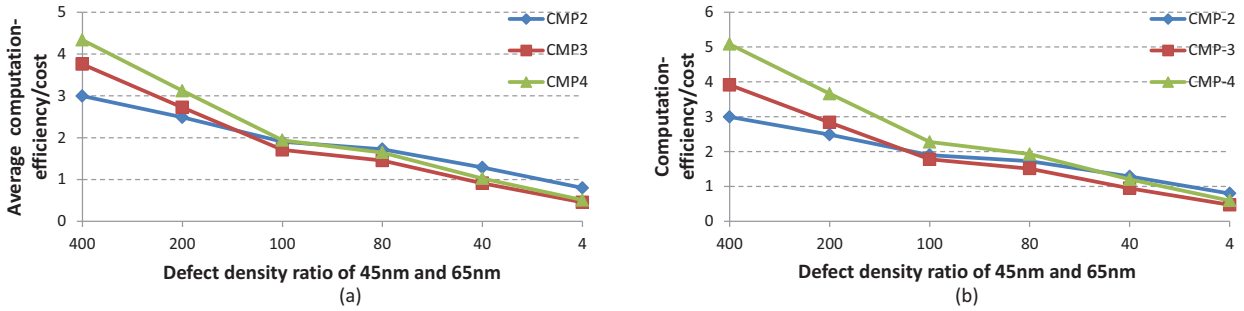


Fig. 3. Computation-efficiency/cost normalized with CMP-1: a) average across all five workload categories b) for workload category CI2-MI2.

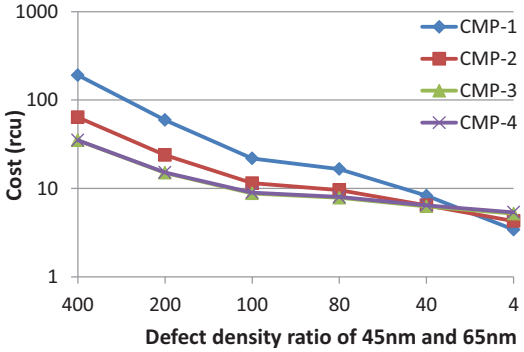


Fig. 2. Cost of different CMPs with the ratio of defect densities on 45nm and 65nm.

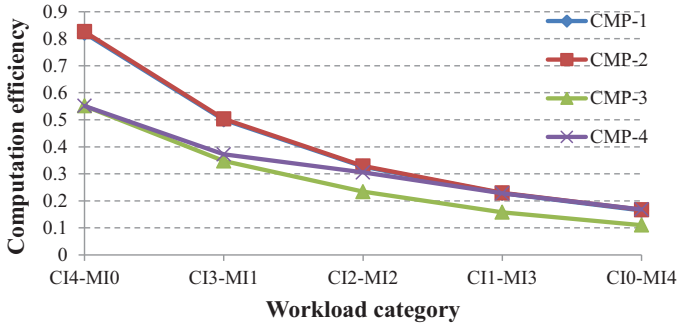


Fig. 4. Computation-efficiency of CMPs for five different workload mix categories

waiting for 45nm to reach this defect density ratio, which can be very useful in time-to-market sensitive product dynamics. In this work we have not considered the parametric yield loss on advance process nodes due to process variation. Considering this will further reduce 65nm cost compared to 45 nm.

Figure 4 presents efficiency (BIPS$^3$/W) of CMPs for five different workload mix categories. CMP-1 is better than other CMPs for most of the workload categories attributing to faster and more efficient transistors at 45nm. CMP-2 is slightly less efficient (0.9% less across five workload categories on average) than CMP-1. This is because of extra power dissipated in TSVs. CI0-MI4 exhibits the highest power dissipation in TSVs because of higher L2 misses incurred by benchmarks in the MI group. CMP-3 is the least efficient across all the categories. The benchmarks executing on tier-2 in CMP-3 consumes higher power as tier-2 is implemented using 65nm. On an average, CMP-3 is 28.5% less efficient than CMP-1 across five workload categories. In CMP-4, by employing microarchitecturally diverse cores, we exploit the ILP diversity that exists among two benchmark groups. In particular, benchmarks in the MI group gain significant instruction throughput (IPC) benefits from a large instruction window (ROB size) and scheduling window (Issue-queue size) of core C5. As described in Section 4.2, benchmarks in MI group have high percentage of long-latency instructions. Such instructions and their dependence chains fill the instruction window of core C6 very frequently. Once the window is full, the instruction dispatch stalls, which leads to no-forward progress made by the core. The microarchitecture of core C5 alleviates this specific bottleneck of core C6 for MI benchmarks by allowing independent instructions to execute. On an average, CMP-4 yields 26.9% better efficiency than CMP-3 and it is 12.5% less efficient than CMP-1. For CI2-MI2 category CMP-4 is only 6.6% less efficient than CMP-1. Moreover, CMP-4 is equally efficient to CMP-1 for CI1-MI3 category and 2.3% more efficient than CMP-1 for CI0-MI4 category. Note that the core C6 is a compromise design that accommodates ILP characteristics of both CI and MI benchmarks. For CMP-4, two threads from a CI0-MI4 sample execute on a compromise design and two threads execute on a well-tuned design, leading to slightly better efficiency than CMP-1.

Finally, we present our overall metric, i.e. efficiency/cost, in Figure 3: a) average across all five workload categories and b) for CI2-MI2 workload category. Average efficiency/cost of CMP-3 and CMP-4 are better than CMP-1 and CMP-2 for high defect density ratio (above 100). Despite CMP-4 being more expensive than CMP-3, it yields a better overall metric than CMP-3 because heterogeneous cores give much better efficiency. This shows the overall benefit of Hetero$^2$ 3D integration on efficiency/cost. As the defect density of 45nm improves, CMP-2 becomes a better design. CMP-2 still has a cost advantage compared to CMP-1 up to a ratio of 20 because of its smaller footprint. After this, the thermal cost of 3D implementation offsets the footprint benefit (more on this in next paragraph). As technology matures, CMP-1 yields the best efficiency/cost metric. For equally diverse workloads, i.e. CI2-MI2, CMP-4 yields better efficiency/cost for an even lower defect density ratio (shown in Figure 3(b)).

Figure 5 presents the breakdown of total cost of different CMPs for workload mix CI2-MI2. In this experiment, defect density ratio of 4 is considered. The rise in channel temperature

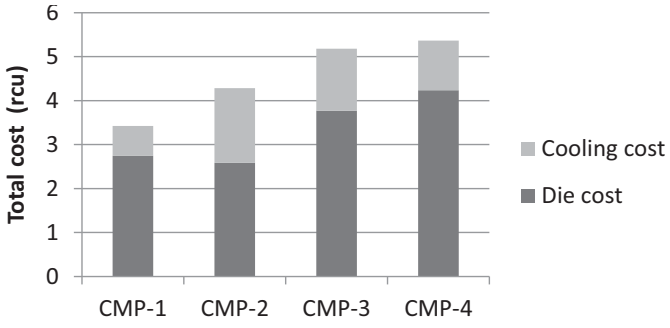


Fig. 5. Breakup of total cost into die and cooling cost for defect density ratio of 4.

was 42°K, 60°K, 55°K and 50°K for CMP-1, 2, 3 and 4 respectively. The die cost of CMP-2 is less than CMP-1 because of a smaller footprint enabled by 3D integration. However, total cost is more because of the higher temperature rise due to higher power density of CMP-2 compared to CMP-1. This is the downside of using 3D integration. Moving from CMP-2 to CMP-3, the cooling cost reduces even if 65nm is more power consuming. This is due to the increased area in 65nm which reduces the overall power density. Temperature rise in CMP-4 is less than CMP-3 because core C5 has lower power density compared to C6. Hetero$^2$ 3D integration helped in reducing some of the thermal cost compared to homogenous 3D integration.

## VI. Conclusion

This work presents a cost effective technique for taking the scaling benefits of advance process nodes in designing CMPs during their initial years of inception when their manufacturing yield is low. The results show that better computation-efficiency per unit cost can be achieved with a 3D stack of heterogenous CMOS processes consisting of older matured and emerging immature technology compared to a 2D implementation using immature technology. Furthermore, we propose to use microarchitecturaly diverse cores for compensating for the loss in efficiency due to using older technology. Using 65nm as a representative of mature and 45nm as an immature technology, computation-efficiency/cost for different ratios of their defect densities are presented. Furthermore, variations in this metric with workload diversity is also shown. Finally, we show the inflection point in terms of defect density ratio (after performing sensitivity analysis by sweeping defect density ratio), until which the technique presented in this work is beneficial. The arguments presented in this paper are also valid for current ultra-deep submicron CMOS technologies.

## Acknowledgment

The authors would like to thank Riko Radojcic (Qualcomm, San Diego, CA) and Rick Hofmann (Qualcomm, Raleigh, NC) for their valuable suggestions. This research was supported in part by Qualcomm, NSF grant CCF-0811707 and gifts from Intel and IBM. Any opinions, findings, and conclusions or recommendations expressed herein are those of the authors and do not necessarily reflect the views of the National Science Foundation.

## References

[1] R. Kumar, K. Farkas, N. Jouppi, P. Ranganathan, and D. Tullsen, "Single-$ISA$ heterogeneous multi-core architectures: the potential for processor power reduction," in *Proceedings. 36th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*, dec. 2003, pp. 81–92.

[2] R. Kumar and D. Tullsen, "Core architecture optimization for heterogeneous chip multiprocessors," in *International Conference on Parallel Architectures and Compilation Techniques (PACT)*, 2006, pp. 23–32.

[3] STMicroelectronics, Part No. VD6803, $http://www.st.com/internet/imag\_video/product/222854.jsp$

[4] U. Kang et al., "8 Gb 3-D DDR3 DRAM Using Through-Silicon-Via Technology," *IEEE Journal of Solid-State Circuits*, vol. 45, no. 1, pp. 111 –119, Jan. 2010.

[5] Elpida Memory Inc., $http://www.elpida.com/pdfs/E0001EE0.pdf$

[6] K.-W. Lee, A. Noriki, K. Kiyoyama, T. Fukushima, T. Tanaka, and M. Koyanagi, "Three-Dimensional Hybrid Integration Technology of $CMOS$, $MEMS$, and $Photonics$ Circuits for Optoelectronic Heterogeneous Integrated Systems," *IEEE Transactions on Electron Devices*, vol. 58, no. 3, pp. 748 –757, March 2011.

[7] N. Madan and R. Balasubramonian, "Leveraging $3D$ technology for improved reliability," in *40th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*, dec. 2007, pp. 223–235.

[8] X. Dong and Y. Xie, "System-level cost analysis and design exploration for three-dimensional integrated circuits," in *Asia and South Pacific Design Automation Conference*, jan. 2009, pp. 234 –241.

[9] R. Weerasekera et al., "Comparative Cost Analysis of 3-D Integrated Circuits," in *Special Interest Workshop on 3D Integration, DATE, 2011*.

[10] J. Cunningham, "The Use and Evaluation of Yield Models in Integrated Circuit Manufacturing," *IEEE Transactions on Semiconductor Manufacturing*, vol. 3, no. 2, pp. 60 –71, May 1990.

[11] ITRS 2007, Yield Enhancement, $www.itrs.net/links/2007itrs$

[12] M. Bohr, "Silicon Technology for 32 nm and Beyond System-on-Chip Products," in *Intel Developer Forum, 2009*.

[13] N. Choudhary, S. Wadhavkar, T. Shah, H. Mayukh, J. Gandhi, B. Dwiel, S. Navada, H. Najaf-abadi, and E. Rotenberg, "Fabscalar: Composing synthesizable RTL designs of arbitrary cores within a canonical superscalar template," in *38th Annual International Symposium on Computer Architecture (ISCA)*, june 2011, pp. 11–22.

[14] N.K. Choudhary, S.V. Wadhavkar, T.A. Shah, H. Mayukh, J. Gandhi, B.H Dwiel, S. Navada, H.H. Najaf-abadi and E. Rotenberg, "FabScalar: Automating Superscalar Core Design," in *IEEE Micro*, vol. 32, no. 3, pp. 48–59, May-June 2012.

[15] X. Dong, J. Zhao, and Y. Xie, "Fabrication Cost Analysis and Cost-Aware Design Space Exploration for 3-D ICs," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 29, no. 12, pp. 1959–1972, Dec. 2010.

[16] K. F. Yang et al., "Yield and Reliability of $3DIC$ Technology for Advanced 28nm Node and Beyond," in *VLSI Technology Symposium*, june 2011, pp. 140–141.

[17] S. Priyadarshi, J. Hu, W. H. Choi, S. Melamed, X. Chen, W. Davis, and P. Franzon, "Pathfinder 3D: A flow for system-level design space exploration," in *2011 IEEE International 3D Systems Integration Conference*, feb. 2012, pp. 1–8.

[18] D. Park, S. Eachempati, R. Das, A. Mishra, Y. Xie, N. Vijaykrishnan, and C. Das, "Mira: A multi-layered on-chip interconnect router architecture," in *35th International Symposium on Computer Architecture (ISCA)*, june 2008, pp. 251–261.

[19] D. H. Woo, N. H. Seong, D. Lewis, and H.-H. Lee, "An optimized 3D-stacked memory architecture by exploiting excessive, high-density tsv bandwidth," in *16th International Symposium on High Performance Computer Architecture (HPCA)*, jan. 2010, pp. 1–12.

[20] T. Sherwood, E. Perelman, G. Hamerly, and B. Calder, "Automatically Characterizing Large Scale Program Behavior," in *10th international conference on Architectural support for programming languages and operating systems (ASPLOS)*, 2002, pp. 45–57.

[21] A. Phansalkar, A. Joshi, L. Eeckhout, and L. John, "Measuring program similarity: Experiments with SPEC CPU benchmark suites," in *IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS)*, march 2005, pp. 10–20.

[22] K. Luo, J. Gummaraju, and M. Franklin, "Balancing thoughput and fairness in SMT processors," in *IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS)*, 2001, pp. 164–171.