Quantifying and Coping with Parametric Variations in 3D-Stacked Microarchitectures

Serkan Ozdemir† Yan Pan† Abhishek Das†
Gokhan Memik† Gabriel Loh‡ Alok Choudhary†
† Northwestern University
Evanston, IL 60208
‡ Georgia Institute of Technology
Atlanta, GA 30332
{s-ozdemir, panyan, a-das, g-memik, a-choudhary}@northwestern.edu loh@cc.gatech.edu

ABSTRACT
Variability in device characteristics, i.e., parametric variations, is an important problem for shrinking process technologies. They manifest themselves as variations in performance, power consumption, and reduction in reliability in the manufactured chips as well as low yield levels. Their implications on performance and yield are particularly profound on 3D architectures: a defect on even a single layer can render the entire stack useless. In this paper, we show that instead of causing increased yield losses, we can actually exploit 3D technology to reduce yield losses by intelligently devising the architectures. We take advantage of the layer-to-layer variations to reduce yield losses by splitting critical components among multiple layers. Our results indicate that our proposed method achieves a 30.6% lower yield loss rate compared to the same pipeline implemented on a 2D architecture.

Categories and Subject Descriptors
C.1.0 **[Processor Architectures]**: General; B.8 **[Performance and Reliability]**

General Terms
Performance, Design, Reliability.

Keywords
Process Variations, Processor Pipeline, Cache Architectures, 3D Integration.

1. INTRODUCTION
In a demand to meet Moore's Law, integrated circuit manufacturers employ technology scaling where on-chip device and interconnect geometries are made smaller with every generation. This results in higher levels of variations in transistor and interconnect parameters, also known as parametric variations [7], which adversely affect the performance of manufactured chips, their power consumption levels, reliability, and yield rates.

Another important problem in deep-submicron manufacturing technologies is the scaling of wire latencies. Reduced load capacitances allow faster switching and increase the performance of a single transistor; the same effect can also be observed in short local interconnects. However, global interconnects linking components across the chip do not scale with technology, and thus limit the improvements gained from technology scaling. Increasingly smaller regions of a chip can be accessed within a single cycle, complicating the processor design significantly, and reducing the performance benefits of smaller technologies.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
DAC'10, June 13-18, 2010, Anaheim, California, USA
Copyright 2010 ACM 978-1-4503-0002-5 /10/06...$10.00

One attractive emerging technology that addresses the increasingly long interconnect latencies is 3D integrated circuits. 3D integration is inviting because it can potentially replace long, global interconnects within a chip with shorter (and faster) wires between different substrate layers, allowing faster communication.

Existing 3D-integration companies mostly focus on multi-layer memory designs. Industry has demonstrated that 3D chips are no longer theoretical constructs, and this has motivated computer architects to start researching new innovative 3D processor organizations. Academic research efforts have been focused on nearer-term coarse-grained stacking and farther-term, more aggressive fine-grained designs. By "coarse-grained," we mean 3D designs that stack multiple components, each of which is still inherently two-dimensional [20, 21]. Such coarse-grained designs are likely to be the first targets for this new technology, since maintaining 2D designs for the individual components allows for greater reuse of designs and reduces the risks of transitioning to the 3D technology. Other research efforts have considered "fine-grained" 3D designs where individual components, down to the functional unit blocks of a processor, may be partitioned across more than one layer of silicon [18]. Such designs will require substantial redesign of processor components, but can significantly reduce both circuit latencies and power consumption. We believe that after industry has developed and proved the design tools, infrastructure and verification/test methodologies for 3D on the easier-payoff coarse-grained designs, we will be able to pursue more aggressive, fine-grained microarchitectures. This is evidenced by recent work from Intel that explores both coarse-grained and fine-grained stacking techniques [5].

Independent of the choice for coarse- or fine-grained design approaches, 3D integration introduces a new risk due to the combinatorial effects of die stacking on manufacturing yield levels, especially due to the severe parametric variations expected of the technologies to be used by the time 3D goes mainstream. For example, layer-to-layer parametric variations can result in a slow layer being bonded to a fast layer, resulting in a chip with low performance since the system clock speed will be limited by the slow layer. These effects would be a severe obstacle to the wide-spread adoption of 3D stacking due to the tight relationship between yield rates and final product cost. One possible remedy to solve this problem would be to test the layers before bonding. However, this may be difficult, since individual layers of a 3D chip may not be functional prior to bonding [17].

In this paper, we show that by carefully allocating critical path devices across 3D-stacked layers, not only can we nullify yield losses due to 3D stacking, but we can also exploit the layer-to-layer parametric variations to actually improve yield rates. While

there have been numerous proposals on how to model parametric variations and come up with efficient architectures/circuit structures to handle issues raised by parametric variations for single layer structures, to the best of our knowledge, this is the first work that investigates and improves yield of 3D integrated circuits using architectural techniques.

To explain how we exploit parametric variations for 3D-stacked processors, we first review 3D fabrication techniques in Section 2 and then we detail our modeling of parametric variations on 3D structures in Section 3. Section 4 presents our cross layer critical path splitting designs. Section 5 presents the experimental results. Section 6 discusses the related work and Section 7 concludes the paper with a summary.

2. 3D FABRICATION TECHNIQUES

Die Stacking, is a recent approach that proposes to fabricate two (or more) separate dies and bond them afterwards [5] to achieve 3D Integration. This idea is beneficial in two ways. First, the individual dies can be manufactured through conventional manufacturing processes. Second, since each layer is fabricated separately, one can employ disparate fabrication methodologies other than CMOS across the different layers. For instance, designers may choose to build a higher density DRAM chip for the second silicon layer.

There are several options to perform die stacking: die-to-die bonding, die-to-wafer bonding, and wafer-to-wafer bonding. In die-to-die bonding, each die is cut from the wafer and then bonded with another die to form the 3D chip. In die-to-wafer bonding, individual dies are cut from one wafer and then placed on top of another wafer to form the 3D chips. Finally, in wafer-to-wafer bonding, two wafers are bonded together to form the 3D chips and then cut to form the chips. Among these, wafer-to-wafer bonding is the most attractive as it exhibits the highest possible manufacturing throughput. Therefore, in this paper, we focus on wafer-to-wafer bonding.

Table 1. Nominal and 3-Sigma Variation Values for each Source of Process Variations Modeled

	L_{eff}	V_t	W	T	H
Nominal Value	45 nm	220 mV	0.25 μm	0.55 μm	2.5 nm
3σ [%]	±10	±18	±33	±33	±35

3. MODELING PARAMETRIC VARIATIONS

Processes like sub-wavelength lithography and aggressive technology scaling result in statistical variations in circuit parameters like gate-oxide thickness, channel length, Random Doping Effects (RDE), etc. [6]. These parametric variations can be classified into die-to-die (D2D) variations and within-die (WID) variations. D2D variation refers to the variation in process parameters across dies and wafers (including between different layers of a 3D stack), whereas WID variation takes place in device features within a single die. Parametric variations can be of two categories: spatially-correlated (systematic) variations where devices close to each other have a higher probability of observing a similar variation level, and random (uncorrelated) variations causing random differences between various devices within a die. In this work, we model both systematic and random parametric variations.

To effectively model parametric variations in 3D chips, we account for five different variation parameters: metal thickness (T), inter-layer dielectric thickness (H), line-width (W) on interconnects, gate length (L_{gate}) and threshold voltage (V_t) for the MOS devices. We use the variation limits given by Nassif et al. [22], as shown in Table 1. To take into account the spatial correlation we use a range factor (φ) in the two dimensional layout of the chip. Thus, each process parameter can be expressed as a function of its mean (μ), standard deviation (σ), and the range (φ) values. If two points x_i and y_i are separated by a distance of d_i, the spatial correlation factor between them can be described as an inverse linear function involving φ and d_i. Note that there is no correlation between two points that are φ units or more apart. The range parameter (φ) is similar to the range parameter described in [25]. With this background, we have generated a spatial map of various parameter values using the R statistical tool [2]. We must note that the φ value has a considerable impact on the randomness of the parameter values. A higher φ means that the values are highly correlated, whereas a low φ value results in a highly random parameter value distribution. We picked a representative value for φ as 0.5 throughout our studies [11].

We use the Monte Carlo method to model a batch of chips. In a Monte Carlo simulation framework, the parametric variation parameters for the chip are first generated, followed by the extraction of the values that correspond to the particular locations of the components studied from this modeled chip. For the 2D process, we use the floorplan for an 8-core processor chip multiprocessor (CMP), with each core being identical to the Alpha EV7 architecture [16]. For the 3D process, we divide our processor into two layers each with 4 cores and map the corresponding components to two individually generated variation maps. To make a fair comparison between the 2D and 3D processes, we have generated 500 distribution maps. These distributions are directly mapped to the floorplan for the 2D process. Similarly, the subset of the map is used to generate the distribution of parameters for the 3D floorplan, producing 500 "layers". Then, a pair of layers from this pool is randomly selected to generate 500 3D chips (during this selection, each distribution map is used exactly twice).

The modeled core architecture is based on Alpha EV7 [16]. The issue queue is modeled as an array structure with 40 entries. Our register file has 80-entries with 4 read and 2 write ports. The integer execution unit is modeled from the netlist obtained after synthesizing the corresponding components in the Sun OpenSPARC [26]. Our L1 cache is a 32 KB 4-way set-associative cache. Each of the 4 ways of our cache is further divided into 4 banks. Each bank has 128x128 cells of storage bits. The shared L2 cache, on the other hand, is a 2 MB 8-way set associative cache with each way consisting of 4 256x256 cell banks. For the remaining pipeline stages (fetch and rename), we use a combination of 16 fan-out of four (FO-4) gates. To study the impact of parametric variations on our processor architecture, we generated SPICE netlists modeling the 10 most critical paths for each component (or for each bank for L1 and L2 Caches).

4. CROSS LAYER PATH SPLITTING

In die stacking, different layers of a chip are manufactured in different wafers. As a result, they exhibit wafer-to-wafer variations that can result in layers with significantly different properties. This has an important effect on the performance and power consumption of the manufactured 3D chips. For example, consider a 3D chip multiprocessor (CMP) that consists of two silicon layers with one core each. The maximum frequency of the chip will be determined by the slowest core in the chip. In other words, the frequency of the slowest core will have to be set as the frequency of the chip to guarantee correct operation. Since the two layers originate from two different wafers, there is a possibility that a fast layer will be bonded with a slow layer. As the impact of

parametric variations increases, this probability will increase as well. The effect of this trend would be a greater number of processors with lower performance. In this paper, we show that splitting the critical components into different layers can eliminate this negative impact and in fact improve the average performance of the chips. We call this technique ***Cross LAyer Path Splitting (CLAPS)***.

4.1 Why Should CLAPS Work?

If we can split every component in a processor into n paths of equal length and distribute each of these sub-paths into a different layer, we should expect the overall performance to increase. The main reason for this expectation is:

$$Max(A_1, A_2, .., A_n) \geq \frac{A_1 + A_2 + .. + A_n}{n} \quad (1)$$

In equation (1), A_1 through A_n correspond to the maximum critical path latency of n layers that form the 3D processor. If we can equally split every path in the core across all n layers, the critical path latency of the resulting 3D processor will be $(A_1 + .. + A_n)/n$. On the other hand, if each critical path independently lies on one layer, the longest one will determine the maximum clock frequency, which is inversely proportional to $Max(A_1, A_2, ..., A_n)$. Hence, splitting critical paths to different layers should improve the clock frequency.

For increasing number of layers (i.e., n), the expected value of the $Max(A_1, ..., A_n)$ also increases, while the expected value of the $Average(A_1, ..., A_n)$ stays the same. In this work, we only focus on 3D structures with 2 layers for the sake of simplicity. However, it is easy to extend our splitting approach to more layers. For example, for a 3D architecture with 4 layers, each critical path will be divided into 4. The complexity of more layers would be dividing the paths to more than 2 parts. It is possible that dividing the critical paths into equal sub-segments may become complicated if n reaches a large number. Even in such cases, our approach would be applicable: one can logically separate the layers, e.g., split the layers into 2 sets and divide the critical paths into $n/2$ parts.

An alternative would be to assign different frequencies to each core so as to improve the yield and performance of the chip. This would result in increased costs for clock generation and distribution, inter-core communication, as well as difficulty in marketing the product as a whole (it is hard to price two chips each with multiple cores running at heterogeneous frequencies). Moreover, this heterogeneous frequency method can be applied on top of our approach to achieve further improvements in yield and/or performance.

4.2 Choosing Components to Split

When split paths are considered, a possible approach would be to divide each and every component in the chip across multiple layers. However, this may not be desirable as splitting each component will require vias to be used, and via densities are lower than the gate densities (and they are not expected to scale at the same rate as technology). Furthermore, this would require a complete redesign of every functional unit block in the processor. Therefore, we have first analyzed a two-layer 3D architecture and identified components that are likely to become critical components after manufacturing. By splitting only these components, we can achieve a high increase in performance of the manufactured chips and still maintain a low design complexity.

Although the level 2 cache is not in the critical memory loop of the EV7 processor pipeline, we included it in our analysis due to its very large size (and hence possibility of creating long critical paths) and is already reported to introduce new bottlenecks in 3D stacked chips [19].

Table 2. Distribution of Critical Paths Across Pipeline Components for 2D and 3D Base Cases

Components	2D [%]	3D [%]
Level 2 Cache	37.6	52.5
Level 1 Data Cache	43.5	30.7
Register File	17.7	16.8
Others	1.1	0.0

Table 2 shows the distribution of critical paths among various pipeline components for the 500 chips generated using the method described in Section 3. For both 2D and 3D cases, the critical path lies in one of the caches or the register file for more than 98% of the simulated chips. This is not a surprising result, as previous work has shown that structures with many independent paths with low logic depth are most likely affected by parametric variations according to the FMAX theory [7]. Thus, our main focus in this work is to *split the caches and the register file*. However, if any other component becomes critical, it is possible to split it as well. In fact, we explore the impact of splitting the critical paths on latency of the other components in the processor pipeline in Section 5.2.

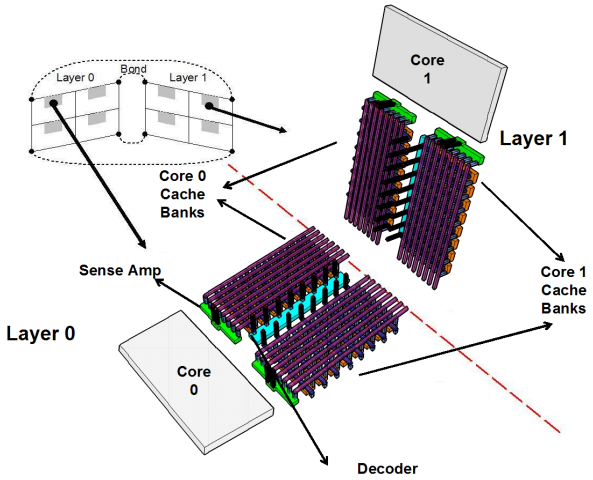

Figure 1. Illustration of Cross Layer Path Splitting (CLAPS) Design.

4.3 CLAPS Designs

The biggest issue with implementing the CLAPS architectures is the size requirements and density constraints on the number of inter-layer vias that are needed. In all our CLAPS designs, we are implementing a vertical split dividing the wordline along with the columns connected to it across two layers. We have also tried to split across the bit-lines, but as described in the following sections, a wordline split provided better overall performance on average.

Figure 1 shows how the vias (shown in black) will be placed on the core for the implementation of the CLAPS design. Note that the two bonding layers (coming from two different wafers) shown on Figure 1 are identical. To allow dies from different wafers to be bonded to each other with minimal wiring, the cores of the CMP in a single layer are laid out as mirror images of each other. Throughout the rest of this paper, we refer to the layer a processor core is located on as the *core layer* for all the components of that core and the other layer as the *opposite layer*. For example, in Figure 1, layer 0 is the core layer for components of core 0 and

layer 1 is their opposite layer and vice versa. In all of our schemes, the wordline signals are generated on the core layer (i.e., the decoder always lies in the same layer as the core). We devised three split strategies where part of the cache banks and/or column mux/sense amp/output drivers are placed on the opposite layer (e.g., in Figure 1, half of cache banks for both cores are sent to their respective opposite layers). Note that, for any CLAPS design, we can partition the register file exactly the same way we partition the cache. This is called *register file splitting*, for which the results are presented in Section 5.1.

4.3.1 Wordline CLAPS (W-CLAPS)

In our first scheme, we place half of the banks (2 banks in this case) on the opposite layer while the remaining banks reside in the core layer. For all banks, the decoder is located in the core layer generating the word-line signals. These signals are then routed to the other layer through vias for the banks that are placed on the opposite layer. For both layers, the sense amplifiers and output drivers are located on the same layer with the corresponding bank. Finally, for the blocks in the opposite layer, the data is re-routed back to the core layer to be sent to the rest of the core. Notice this scheme needs *(#Rows + OUTPUT_WIDTH/2)* inter-layer vias for correct operation. For the level 1 data cache we are using, this corresponds to 160 vias (128 rows + 64/2 bits). The additional parasitics for these vias are included in our SPICE model. For our current architecture, we assume 1 µm x 1 µm dimensions, 2.4 µm pitch for the vias and 0.22 μm^2 for the SRAM cell area as given in ITRS [1], hence the vias span roughly 15% of the total area of the cache *(Note that vias do not increase the cache area by 15%. All the vias can be laid on top of 15% of the total cache area, while allowing plenty of space for other die to die connections such as clock and power distribution)*. Figure 2(a) shows the original 2D structure, whereas Figure 2(b) shows the 3D structure of one of the caches divided into two layers in detail (W-CLAPS). Vias are represented with black vertical prisms. Note that Figure 2 shows face-to-back connection for better viewing, while in our models we use face-to-face stacking.

4.3.2 Decoder CLAPS (D-CLAPS)

It is clear from Figure 2(b) that only half of the critical paths in the component are being split across two layers. However, the other half of the paths has all of their critical paths (decoder, memory array, sense amplifiers and output drivers) within a single layer, thereby limiting the benefits of the split architecture. To further increase the benefits we achieve from splitting, we experimented with a second scheme where all paths are divided into two layers. To achieve this, we placed all banks on the opposite layers. To limit the number of vias required, we decided to take advantage of the vias already used to route the wordline signals to the opposite layer. As can be seen in Figure 2(c), all of the paths are split right after the decoder and all banks are located on the opposite layer along with the sense amplifiers and output drivers. Although this splitting scheme has the advantage of dividing all critical paths into two, the number of vias needed to route the data back to the original layer increases to *(#ROWS + OUTPUT_WIDTH)*; or 192 vias (128 rows + 64 bits) for the level 1 data cache (*about 18% of the total area of the cache*) because the second sub-bank is also moved to the opposite layer. Another drawback with this strategy is that it does not take advantage of 3D placement to shorten the long wires. Particularly, the global wordline must cross the two banks as in the 2D architecture, whereas in the W-CLAPS design, the wordline length is divided into two.

4.3.3 Bitline/Wordline CLAPS (B-CLAPS)

To simultaneously minimize long wires and take advantage of split critical paths, we devised a third architecture where we split all the paths while avoiding the use of a long global wordline. At the cost of increased via-count, we split half of the paths after the decoder as in W-CLAPS design. For the remaining paths we keep the banks as in W-CLAPS but move the sense amplifiers and output drivers to the opposite layer. Figure 2(d) shows the details of this implementation. As can be seen in the figure, we now need to augment each column in the core layer with two vias, one for each Bit-Line and Bit-Line-Bar. In this scheme the new via count can be expressed as *(#Rows + OUTPUT_WIDTH + 2*#COLUMNS/2)*.

For the L1 Data Cache this corresponds to 256 vias (128 rows +64 bits +(2 × 64 columns)/2), *roughly spanning 24% of total cache area*. For a face-to-face bonding technology, this splitting does not incur any area overhead since the vias fit within the cache footprint.

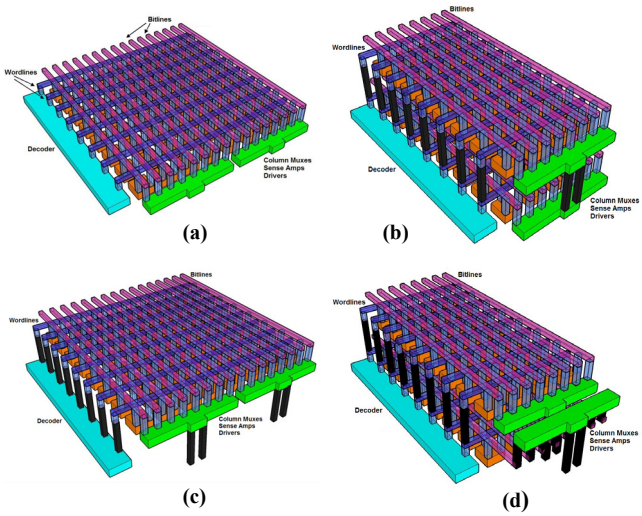

Figure 2. Three Dimensional Structures of Studied Designs: (a) 2D, (b) W-CLAPS, (c) D-CLAPS, and (d) B-CLAPS.

5. EVALUATION

To analyze the effectiveness of the proposed schemes, we measure their impact on yield, performance, and temperature. In the next section, we explain the improvements in yield and average chip frequency we achieve through our CLAPS designs. Then, in Section 5.2 we explore the impact of CLAPS on the latency of the critical paths of non-SRAM based components in the processor pipeline. Finally, in Section 5.3 we discuss the impact of CLAPS designs on the thermal profile of 3D integration.

Table 3. Yield Levels Achieved for Proposed Schemes

Yield [%]	BASE	Cache Split	Cache+Register File Split
2D	76.39	NA	NA
W-CLAPS	72.13 (3D)	75.74	78.51
D-CLAPS		77.02	77.02
B-CLAPS		81.70	83.62

Table 4. Normalized Average Frequency for Each Scheme

BP	BASE	Cache Split	Cache+Register File Split
2D	1.000	NA	NA
W-CLAPS	0.907 (3D)	1.003	1.050
D-CLAPS		0.941	0.941
B-CLAPS		1.087	1.131

5.1 Yield and Performance Results

In this section, we investigate how different CLAPS designs affect yield levels and average frequency. In this work, we focus on analyzing parametric yield loss due to frequency or power constraints, which is the most dominant factor in yield losses [15]. We must also note that, in our experiments, the total area of 3D chips are the same as the original 2D chip hence defect related yield losses will be similar. We also assume a mature manufacturing technology where the yield is not sensitive to the number of inter-layer vias in the design. Thus we assume CLAPS would face the same amount of manufacturing yield loss as the base line 3D architecture without splitting. In order to study the effectiveness of CLAPS designs we performed a Monte Carlo study of a set of SPICE simulations. In total, we simulated 500 chips with 8 cores, each of which is split into 4 cores per layer for the 3D studies. Then, we calculated the critical path latency for each chip by finding the pipeline stage with the maximum latency in each core. The maximum allowed latency is set to be $\mu_{2D}+0.5\times\sigma_{2D}$ and the maximum allowed leakage power is set to $3\times average\_leakage_{2D}$ [24]. Chips exceeding these limits are considered yield losses. The performance, on the other hand, is calculated as the average frequency across all the chips.

Table 3 shows the yield levels over all chips simulated for each scheme. Similarly, Table 4 shows the normalized average frequency among all chips for that particular scheme relative to the average frequency of the original 2D case. When we compare the yield levels of the 2D and 3D base cases in Table 3, we observe that the 3D base architecture causes approximately 16% increase in yield losses. However, the new yield, 72.13%, is considerably higher than one might expect from randomly merging two dies, which would be $(0.7639)^2 = 58.35\%$. This is mainly due to the reduction in the chip area per layer. A smaller footprint allows higher correlation and less variation between two points in a chip layer preventing critical path latencies from taking extreme values.

When we utilize our CLAPS designs over the 3D base case, they show a significant improvement in total yield. However, since only half of the critical paths are divided across two layers in W-CLAPS, its yield level falls short of 2D case. The best scheme, B-CLAPS, reduces the yield losses of the base 3D case by 34.3%, achieving a yield rate exceeding the 2D architecture. The reason for achieving the best results for this scheme lies in the fact that all the paths are split into two in B-CLAPS. As a result, the probability of "averaging effect" is maximized for B-CLAPS.

Another important point is the ineffectiveness of the register file split in 3D D-CLAPS. The main reason behind this is the increased latency in level 1 data cache and the level 2 cache due to the additional delay of inter-layer vias on top of the global wordline that needs to be kept for this splitting scheme. Although register file splitting helps with all these chips, caches dominate the latency canceling the effects of register file splitting. This can also be observed in the change of average frequencies (presented in Table 4), where the increased latencies of level 1 and level 2 caches dominate critical path latency reducing the effect of the scheme. In the best case, we can improve the yield up to 83.6%, which corresponds to a 30.6% reduction in yield loss compared to the 2D architecture; while the average chip frequency for the batch of chips simulated for that scheme is increased by 13.1%.

5.2 CLAPS Designs for non-SRAM Based Components

In this section, we study how CLAPS performs when it is implemented on the non-SRAM based components. Although our processor and process variation models resulted in SRAM dominated critical paths; using different models and/or optimization targets and constraints may end up in a different distribution for the critical paths among components. Hence, it is important to quantify the effectiveness and limitations of our proposed scheme if such a scenario is encountered.

We particularly focus on the branch predictor, register rename unit, issue queue, and integer execution units representing the fetch, rename, issue and execute stages of the processor pipeline. While implementing CLAPS, we concentrate on splitting the critical paths evenly into two layers in terms of latency. We perform the split operations manually on our designs. However, one can easily transform this process into a partitioning problem optimizing for number of vias, portion of path latency on each layer as well as percentage of the component area on each layer.

Through Monte Carlo simulations, we observed that the average latency of the fetch and rename stages are reduced by 8.1% and 99% of the chips performed better than the 3D without splitting case. On the other hand, for the issue queue the average reduction in latency was only 4.3% but CLAPS still outperformed 3D without splitting 96% of the time. An interesting result was achieved for the integer execution unit where we observed a 0.2% increase in the critical path latency. Despite the increase in latency, CLAPS still increased the performance of 49% of the chips. Overall, these results show that CLAPS is feasible and effective for all components in the pipeline.

5.3 Impact on Temperature

3D integration increases power density and hence may cause higher worst-case temperatures. A higher layer in the stack needs to cross a higher thermal resistance to reach the ambient temperature making it harder to cool down [20]. Various techniques have been proposed [5, 21] to mitigate this impact. In this section, we evaluate the thermal impact of our proposed CLAPS designs and demonstrate that our approach yields reasonable temperature levels consistent with other studies.

To verify the thermal impact of our proposed architecture, we simulated a processor consisting of 8 Alpha cores using the M5 simulator [4] integrated with Wattch [8]. This provides us the steady state power numbers for the various components in each core. HotSpot [14] is then used to simulate the temperature profile of the chip. We use a four-layer model for the 2D case and a 7-layer model for the 3D case. Parameters of the layers are set in correspondence to previous work [5].

Table 5. Max, Mean, and Min Temperatures and Total Power Consumption for Different Benchmarks

Mix	Floor-plan	Max Temp [K]	Mean Temp [K]	Min Temp [K]	Total Power [W]
Extreme	**2D**	369.6	343.1	338.2	141.8
Extreme	**3D**	378.3	349.7	346.5	150.6
High	**2D**	368.9	341.8	337.4	125.9
High	**3D**	374.7	349.2	346.4	136.7
Medium	**2D**	362.5	340.4	336.7	109.1
Medium	**3D**	366.6	348.7	346.2	121.1
Low	**2D**	360.5	338.8	335.9	90.9
Low	**3D**	367.5	348.2	346.1	104.3

Leakage power is modeled to be exponentially dependent on temperature and iteratively updated to get the steady state power. To model the power intensities of different benchmarks, we sorted the IPC of all of the SPEC2K benchmarks and created four mixes of the workloads for our 8-core CMP and simulate them using the M5 simulator. "Extreme" assigns the highest IPC benchmark to all eight cores; "High" contains the top 8 highest IPC benchmarks; "Medium" executes the next 8 highest IPC applications on the cores; "Low" contains the next 8 highest IPC benchmarks. As the two active silicon layers are close to each other, there is negligible temperature difference (around 0.1K according to our simulation) between layers. Thus, the three CLAPS designs share almost identical thermal and power profiles, and are represented as "3D" in Table 5, which provides the maximum, minimum and mean temperatures and power consumptions of the 2D and 3D architectures for the studied workloads.

In Table 5, it can be observed that the mean temperature in the 3D case is 6°C to 10°C higher than the 2D case due to the overlapping of the two active silicon layers, which matches previously reported studies [23]. This corresponds to 10 to 15 Watts of extra power consumption due to the increased leakage.

6. RELATED WORKS

Multiple research groups proposed different ways to partition caches and SRAM [27] and DRAM [19] structures across multiple layers of silicon. Other studies have examined the 3D implementation of other processor structures including register files, arithmetic units and instruction schedulers [28]. In another work, researchers examine different coarse-grained 3D stacking assignment options (processor on one chip, L2 cache on another chip) using die-to-die bonding [10]. A more recent study investigates the effects of 3D process variations on temperature which in turn affects performance [3] while another work analytically investigates 2D and 3D critical paths for both within die and die-to-die paths, reaching to similar conclusions as in our work. However, they do not propose any mitigation techniques addressing the problem [12]. Our CLAPS caches and register files have structural similarities to some of the previous designs, but whereas the prior work focused on power and performance, our approach is primarily targeted at yield which had not been previously considered. The design automation community has researched algorithms for automated floor-planning, placement and routing of microprocessor components [9, 13]. These designs are particularly prone to yield losses due to layer-to-layer parametric variations because the critical paths for entire blocks and pipeline stages are assumed to be assigned to only a single layer. Design automation tools targeting variation-induced yield losses may be an interesting direction for future research.

7. CONCLUSIONS

Continued technology scaling is causing problems with interconnect and device latency variations, forcing researchers to find alternative means to keep up with Moore's Law. 3D integration, which is only starting to become practical/cost-effective, may be a very effective means to achieve this. In this paper, we explored some of the design issues that arise from the combination of 3D structures and parametric variations. We argue that, if left untreated, these factors will cause a significant drop in yield levels. Through Cross LAyer Path Splitting (CLAPS) we can overcome these factors and reach higher yield levels compared to the 2D base case. Experimental results show that by utilizing our scheme we can achieve 30.6% lower yield loss and a 13.1% increase in average performance while keeping the chip thermally controllable.

ACKNOWLEDGEMENTS

This work was in part supported by NSF grants CCF-0916746, CCF-0747201, and CNS-0720691. We would also like to thank all the anonymous referees for their helpful comments.

REFERENCES

[1] *ITRS Roadmap Update*. 2006 Available from: http://www.itrs.net/Links/2006Update/2006UpdateFinal.htm.

[2] *The R Project for Statistical Computing*. Available from: http://www.r-project.org.

[3] Akopyan, F., et al. *Variability in 3-D Integrated Circuits*. in *CICC*. Sep. 2008. San Jose, CA.

[4] Binkert, N.L., et al., *The M5 Simulator: Modeling Networked Systems.* IEEE Micro, July 2006. **26**(4): p. 52-60.

[5] Black, B., et al. *Die Stacking (3D) Microarchitecture*. in *MICRO*. Dec. 2006. Orlando, FL.

[6] Borkar, S., et al. *Parameter variations and impact on circuits and microarchitecture*. in *DAC*. June 2003. Anaheim, CA.

[7] Bowman, K.A., S.G. Duvall, and J.D. Meindl, *Impact of Die-to-Die and Within-Die Parameter Fluctuations on the Maximum Clock Frequency Distribution for Gigascale Integration.* Journal of Solid-State Circuits, Feb. 2002. **37**.

[8] Brooks, D., V. Tiwari, and M. Martonosi. *Wattch: A Framework for Architectural-Level Power Analysis and Optimizations*. in *ISCA*. 2000.

[9] Cong, J., et al. *An Automated Design Flow for 3D Microarchitecture Evaluation*. in *ASPDAC*. Jan. 2006. Yokohama, Japan.

[10] Ferri, C., S. Reda, and R.I. Bahar. *Strategies for Improving the Parametric Yield and Profits of 3D ICs*. in *ICCAD*. 2007. San Jose, CA.

[11] Friedberg, P., et al. *Modeling Within-Die Spatial Correlation Effects for Process-Design Co-Optimization*. in *ISQED*. Mar. 2005.

[12] Garg, S. and D. Marculescu. *3D-GCP: An analytical model for the impact of process variations on the critical path delay distribution of 3D ICs*. in *ISQED*. Mar. 2009. San Jose, CA.

[13] Healy, M., et al., *Multi-Objective Microarchitectural Floorplanning for 2D and 3D ICs.* TCAD, Jan. 2007. **26**(1): p. 38-52.

[14] Huang, W., et al., *HotSpot: A Compact Thermal Modeling Method for CMOS VLSI Systems.* TVLSI, May 2006. **14**(5): p. 501-513.

[15] Jones, H.H. *A Delayed 90-nm Surprise*. Available from: http://www.designchain.com/column.asp?id=2&issue=fall04.

[16] Krewell, K., *Alpha EV7 Processor: A High-Performance Tradition Continues*, Apr. 2002

[17] Lewis, D. and H.S. Lee. *A Scan-Island Based Design Enabling Pre-bond Testability in Die-Stacked Microprocessors*. in *Int'l Test Conf.* Oct. 2007. Santa Clara, CA.

[18] Liu, Y., et al. *Fine Grain 3D Integration for Microarchitecture Design Through Cube Packing Exploration*. in *ICCD*. Oct. 2007. Lake Tahoe, CA.

[19] Loh, G. *3D-Stacked Memory Architectures for Multi-Core Processors*. in *ISCA*. June 2008. Beijing, China.

[20] Loi, G.L., et al. *A Thermally-Aware Performance Analysis of Vertically Integrated (3-D) Processor-Memory Hierarchy*. in *DAC*. June 2006. Anaheim, CA.

[21] Mysore, S., et al. *Introspective 3D Chips*. in *ASPLOS* 2006. San Jose, CA.

[22] Nassif, S.R. *Modeling and Analysis of Manufacturing Variations*. in *CICC*. May 2001.

[23] Puttaswamy, K. and G.H. Loh. *Thermal Analysis of a 3D Die-Stacked High-Performance Microprocessor*. in *GLSVLSI*. 2006. Philadelphia, PA.

[24] Rao, R.R., et al., *Modeling and Analysis of Parametric Yield under Power and Performance Constraints.* Design and Test of Computers, Aug. 2005. **22**(4): p. 376-385.

[25] Sarangi, S.R., et al., *VARIUS: A Model of Process Variation and Resulting Timing Errors for Microarchitects.* Trans. on Semiconductor Manufacturing, Feb. 2008. **21**(1): p. 3-13.

[26] Sun. *OpenSPARC T1*. Available from: http://www.opensparc.net/opensparc-t1/index.html.

[27] Tsai, Y.-F., et al. *Three-Dimensional Cache Design Exploration Using 3DCacti*. in *ICCD*. Oct. 2005. San Jose, CA.

[28] Vaidyanathan, B., et al. *Architecting Microprocessor Components in 3D Design Space*. in *Conf. on VLSI Design*. Jan. 2007. Bangalore, India.