Architecting Microprocessors in 3D Design Space

ABSTRACT

Interconnect is one of the major hindrances for current and future microprocessor designs from both a performance and a power consumption perspective. As the trend of microprocessor design is to extract more instruction-level parallelism by adopting out-of-order issue and large window sizes, the increases of communication latency and interconnect power associated with these two enlarged factors are inevitable. The emergence of three-dimensional (3D) chip architectures, with their intrinsic capability of reducing the wire length, is one of the promising solutions to mitigate the interconnect problem. We have implemented a few components of a microprocessor to show the potential benefits obtainable through 3D integration. In contrast to prior work, which mostly investigates single components of a processor, our work takes multiple components into consideration and the experimental results are promising in terms of delay and power reductions. By incorporating three 3D-optimized implementations, we demonstrate that the performance of these units is increased, allowing more complex implementations at equivalent clock rates. We then show that the performance of a microprocessor measured in IPC can be improved up to 11% when implemented on a 2-strata 3D technology.

1. INTRODUCTION

Although the size and switching speed of transistors benefit as technology feature sizes continue to shrink, interconnect wire does not scale accordingly with technologies. The increasing wire RC delays have become one major impediment in meeting both power consumption and performance targets with growing overall chip dimensions. According to the projection from International Technology Roadmap for Semiconductors (ITRS), for current and future technologies new design approaches are critically needed in order to offer on-chip communication to meet system level performance requirements.

There are various reasons why interconnect has substantial influence in microarchitecture designs. First, as the trend of designing processors in current and future generations is to adopt multiple pipeline stages, the increase of loop delay for many *loose loops* is inevitable [1]. That is, the result of the *resolution* stage has to travel multiple pipeline stages to feed the request from the *initiation* stage. Although the loop length is in part decided by microarchitectural tradeoffs, the physical wiring between stages is also an important aspect in determining communication latency. Second, global and semi-global signal wires are becoming exceptionally long as the result of scaling. In order to ensure systems operating at the right frequency, memory elements are inserted so as to distribute the delays of long wires over clock cycles. However, such an approach known as *wire-pipelining* has the possibility of changing the functionality of a circuit with varying latency presented arbitrarily along the paths in the circuit [3]. Finally, as Kapur et al. states in [2] that with the introduction of repeaters and vias to compensate the performance lost, interconnect power consumption almost doubles.

Since interconnect wires account for a major portion of power consumption and communication delay of microprocessors, solutions to lessen wire-related issues have been widely proposed in recent years. The use of three-dimensional (3D) integrated technology is one technique being vigorously researched to alleviate the problems of interconnects. The 3D technology vertically stacks device layers after processing each active device layer separately and the connections between dies are established by 3D vertical vias. One key advantage of 3D chips over traditional two-dimensional chips is the direct wire length reduction from the geometric point of view. With the increasing flexibility in vertical direction and the decreasing die area, the large capacitances induced from long wires are effectively lowered, which in turn reflect on power reductions and latency improvements in correspondence to long interconnects.

Recently, a large number of techniques for 3D technology have been explored in literature. For instance, work [8, 9] targeted on 3D cache designs. Word-line and bit-line partitioning were proposed to distribute different portion of a cache onto different strata. Potential benefits of implementing an IA32 in the 3D technology were described in [6]; however, details of 3D design for the processor's internal blocks are unavailable. Alternatively, researches have been actively investigating in physical design and automated tool design for the 3D technology. Microarchitecture evaluations [10, 11] through floorplanning often adopt Manhattan-based wire length estimation to guide algorithms toward the low

wiring direction. However, congestion and physical wiring are neglected with this approximation method, and such approaches can not be directly applied to the 3D technology due to the existence of through-wafer vias. Das et al. [5] have developed tools for supporting custom 3D layouts and 3D placements. Unlike much of academic work focusing on tool designs, some industry companies already start shipping out 3D-stacked SRAM, DRAM, and microcontroller products [16]. In total, we are not aware of any 3D microarchitecture studies for high-performance microprocessors. Essentially, there is a clear need for one to investigate how a 3D architecture affects the wire-bound components of a microprocessor in terms of power and delay.

In this paper, we explore the architectural design of a few important microarchitectural blocks from the 3D angle of view, which include instruction scheduler, Kogge Stone prefix adder, and logarithmetic shifter. Comparisons between 2D and 3D implementations of these components show promising outcomes. However, due to the complexity of a microarchitecture, the whole gain of 3D integration is much more difficult to determine; and more specifically, a full implementation for a specific technology is required. Therefore, our work here is trying to demonstrate the potential advantages of the 3D microarchitecture to architects and thus help to make decisions at a fairly early stage of the design process.

The rest of the paper is organized as follows. Section 2 briefly reviews the 3D technology. Section 3 investigates the possibilities of implementing the components of a microprocessor in 3D. Section 4 presents and discusses experimental results. We conclude this paper in the last section.

2. BACKGROUND OF 3D TECHNOLOGY

There are numerous novel 3D integrated technologies under development. In this paper, we have considered one of the promising styles of 3D technologies: *wafer-bonding technology* [12]. In the wafer-bonding technology, 3D integrated circuits are formed by vertical stacking of multiple strata where each stratum is an active device layer and is processed independently, and 3D vias provide die-to-die connections and act as the mechanism for holding dies together as well. There are two types of bonding techniques, face-to-face and face-to-back, which are shown in Figure 1. In face-to-face bonding, the 3D vias are processed and deposited on top of metal layers as the traditional metal etching technologies. Although this approach can provide higher via density due to the similarity of synthesizing the regular on-die interconnects, it allows only two active layers in a 3D stack. For face-to-back bonding, on the other hand, any number of dies can be stacked. However, vias in this configuration must be etched through the back side of a die and less via density is possible due to the less resolution of the etching process compared to its counterpart.

In the wafer-bonding 3D technology, the dimensions of the vertical through-wafer interconnect are not expected to scale at the same rate as feature size, because wafer-to-wafer alignment tolerances during bonding pose limitations on the scaling of the through-wafer interconnect. The distance between two top metal layers is about the height of 5 um to 20um and the dimension of die-to-die vias vary from 1um-by-1um to 10um-by-10um depending on the technology. The relatively large size of via makes the interconnect delay going through wafer to be relatively much smaller. As reported in [9], the communication delay of a die-to-die via is much less than a FO4 delay in the 70nm technology.

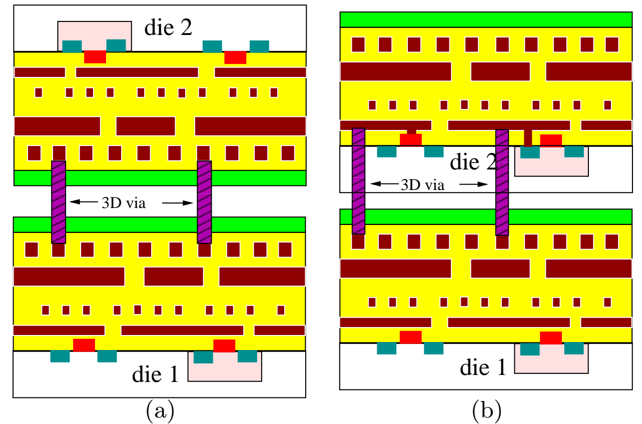

Figure 1: (a) Face-to-face bonding. (b) Face-to-back bonding.

3. PROCESSOR MODULES IN EXPLORATION

3.1 Instruction scheduler

The issue scheduler (logic) in a dynamically-scheduled superscalar processor is a complex mechanism which is in charge of starting execution of multiple instructions. The instruction scheduler consists of two major components, wake-up and selection logic. The wake-up logic is responsible for awakening instructions that are eligible for issuing when both of its source operands have been produced and the requested functional unit becomes available. The function of the selection logic is to determine which instructions should be issued up to the maximum issue width of a processor. Due to its complexity, a significant amount of energy is consumed. Moreover, as pointed out in [4], the logic associated with the issue scheduler will be one of the primary clock speed limiters because wake-up logic and selection logic form an atomic operation. As a result, the turnaround communication delay between these two components should be made as less as possible to meet the performance budget because the wire delay will soon dominate overall delay as feature size keeps scaling.

Both the architectural level structure of wake-up and selection logics are shown in Figure 2. Our implementations of these two logics adopted the architectural designs proposed in [4]. The delay of the wake-up logic consists of three components and can be expressed as $Delay = T_{tagdrive} + T_{tagmatch} + T_{ormatch}$, where $T_{tagdrive}$ represents the time taken by buffers to drive tag bits, $T_{tagmatch}$ represents the time for a comparison cell implemented as CAM structure to pull down the match line, and $T_{ormatch}$ represents the time needed to OR individual match line. The delay time of the selection logic is composed of the propagation time for request signals to get to the root arbiter cell, the time for the root arbiter cell to generate the grant signal, and the time to propagate the grant signal to the selected instruction for starting execution.

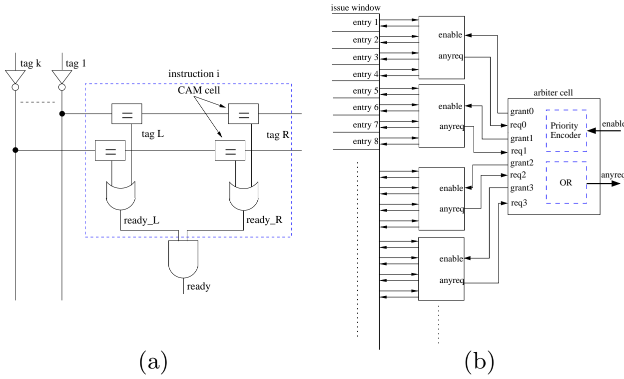

Figure 2: (a) Wake-up logic. (b) Selection logic.

The delay of wake-up logic is affected by both issue width and window size, whereas the selection delay is mostly influenced by the size of the instruction window. More specifically, the $T_{tagdrive}$ is the most influential one in deciding the overall delay of the issue logic based on following HSPICE simulations.

Figures 3(a), 3(b), and 3(c) show the delay breakdown of the wake-up logic under different issue widths and window sizes. The increased window size affects the delay of the $T_{tagdrive}$ most significantly as visible from these figures. Figure 3(d) shows the power consumption comparison of different issue widths and window sizes. As expected, power consumptions are higher with larger window sizes due to the big cumulative capacitances. Accordingly, larger issue width means more wires are needed to deliver the results from functional units back to the comparison cells, which also has influence on power. In addition to wires, more comparison cells are needed alone with the enlarged issue width.

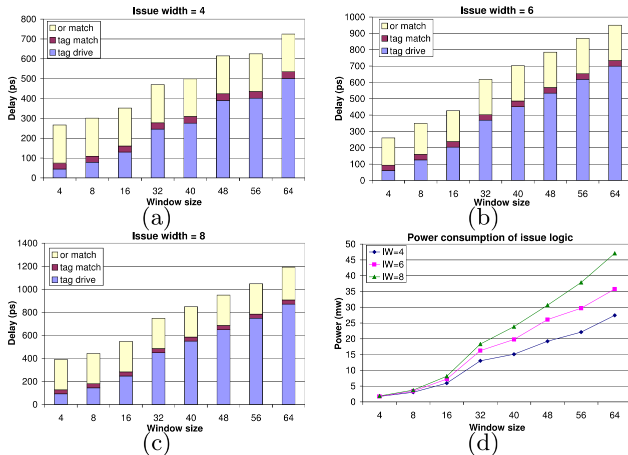

Figure 3: Wake-up logic delay breakdown and power with different issue widths and different window size.

Figure 4(a) shows the delay breakdown of the wake-up logic with different issue widths and a fixed instruction window size of 64. The overall delay increases from issue width of four to six is 23.7% whereas 20.3% is observed for issue width of six to issue width of eight. One thing to note from this figure is that the time of $T_{ormatch}$ is only affected by the issue width, albeit slightly. This is because the delay of an OR gate is mainly decided by the number of input pins it has, which corresponds to the issue width. On the other hand, window size has greater impact on the delay than issue width because the wire delay is more eminent in advanced technologies. Note that, neither the window size nor the issue width affects the time of tag-match since the comparison cell is designed based on CAM structure and its delay is solely dependent on discharging the match line. Figure 4(b)shows the delay breakdown of the selection logic for different issue widths with a fixed window size of 64. As can be observed, the delay of the arbiter is independent of the window size and the increase in delay for forward and backward paths are not 100% for window size 8 to 16 and window size 32 to 64 due to the log_4 structure of the selection logic.

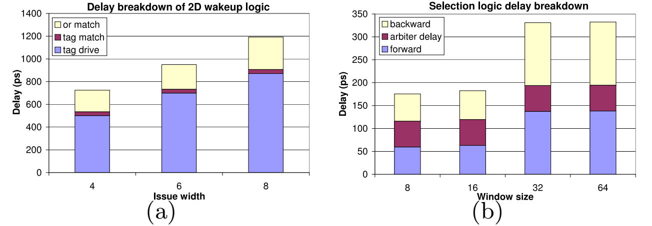

Figure 4: (a) Wake-up logic delay breakdown. (b) Selection logic delay breakdown.

Based on the HSPICE simulation results shown above, we know that the delay time of $T_{tagdrive}$ dominates the overall delay of the issue logic, and the wake-up logic itself is one of major barriers in boosting the performance of a microprocessor. One solution to tackle the wire-induced problems is the move to 3D technology, and thus implementing this logic in 3D to mitigate the derivative issues of $T_{tagdrive}$. From this point of view, two possible partitioning approaches can be applied to the long tag-drive lines. The first one is referred to as *horizontal partitioning*, which cuts the tag-drive lines in half horizontally and place one-half length of tag-drive lines on the first layer and that of the other on the second layer assuming two active device strata are used. In other words, we duplicate the tag-drive lines with only half-long length of tag-drive lines onto each layer and thus lowered wire capacitance can be acquired. We refer the second approach to as *vertical partitioning*, which separates the tag-drive lines vertically into two halves. That is, we can assume tag[0:3] is on one stratum while tag[4:7] is on another stratum. Both partitioning approaches are conceptually shown in Figure 5.

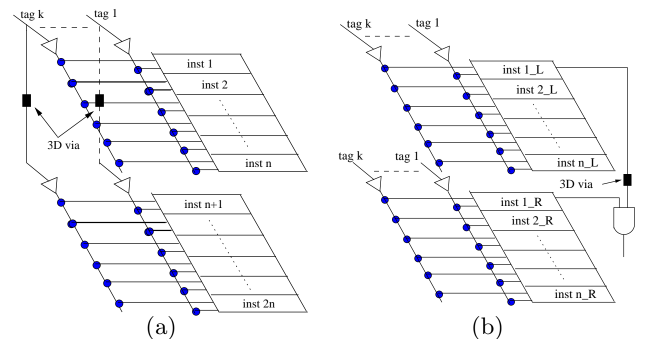

Figure 5: (a)Horizontal partitioning of tag-drive. (b)Vertical partitioning of tag-drive.

3.2 3D Arithmetic Design

In this section, we look at 3D arithmetic function unit designs. According to [7], highly parallel circuits benefit more from the increased number of neighboring gates in 3D systems than those highly serial circuits. Therefore, we only

investigate some arithmetic function units that can have potential improvement on critical paths while implemented on 3D.

3.2.1 Kogge Stone adder

The Kogge Stone (KS) adder is one of the fastest adders in CMOS design. Since the interconnect length in the critical path increases linearly with the number of inputs, wire delay dominates its performance in the current deep submicron technologies [13].

Figure 6(a) shows the 2D placement of the 16-bit KS adder and Figure 6(b) shows the corresponding schematic 3D placement of this adder in 4 layers. For the sake of clarity only the bottom 3 layers are shown. The 3D layers are shown in different shades in order to match the corresponding 2D design. Note that, 3D via contacts, not shown in the figure, are needed when signals travel across multiple layers. The critical path (highlighted in bold line) in the 2D adder spans across 12 cells against 3 cells in 3D. It is visible from the 3D placement that the cell placement wraps around for every 4 cell, which gives a maximum of 4x wire length reduction in 3D along the critical paths.

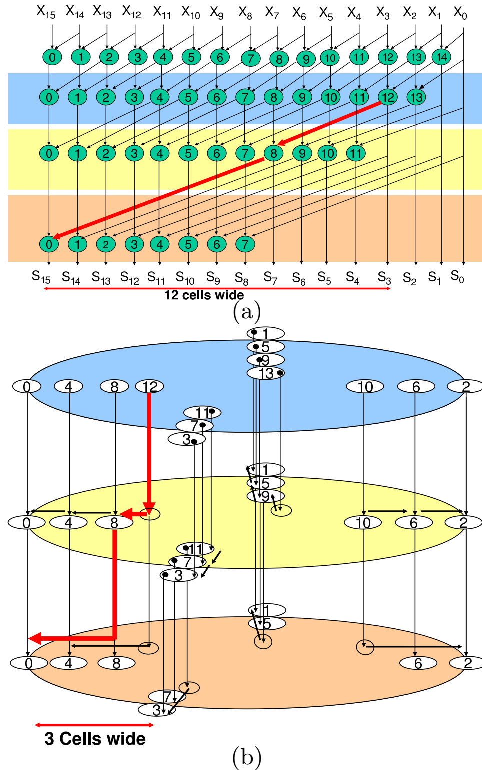

Figure 6: 16-bit KS adder in 2D (a) and 3D (b), critical path is shown in bold line

3.2.2 Logarithmic shifter

Another design which has wire delay impacts on performance is the logarithmic shifter. The 2D layout of the 8-bit log shifter in Figure 7(a) shows the linear dependence of wire length on the number of inputs. The metric used here to calculate the wire length is based on the number of cells crossed by the wire before reaching the destination (i.e., wire length is calculated in number of cell units).

The cells in the 8-Bit log shifter are 2-1 muxes which gets it select signal from s0, s1, and s2. Figure 7(b) shows the placement of the shifter cells in 2 strata. The 2D implementation of log shifter has a critical path highlighted in bold line of 10 cells, while as the corresponding path in 3D spans only 4 cells and 2 vias as shown in the figures.

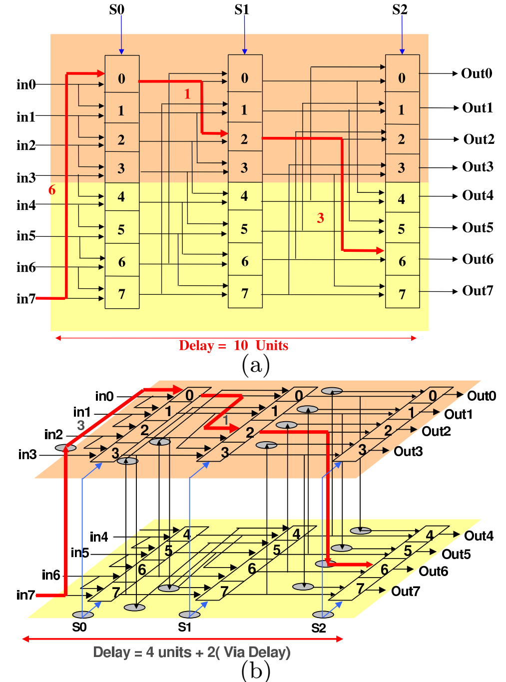

Figure 7: Log shifter in 2D (a) and 3D (b), critical path is shown in bold line

4. EXPERIMENTAL RESULTS

The processor's components mentioned in Section 3 were all implemented in 70nm technology with BSIM transistor models from UC Berkeley [17]. The latency and power of all components in 2D and 3D were acquired through a combination of circuit-level HSPICE simulations. The wire model was scaled to 70nm because the parameters processed by HSPICE is TSMC 180nm technology. To model the performance and delay impacts of the 3D via more accurately, the RC delay of 3D via is added to the circuits to reflect its influence. The resistance of 3D via is estimated to be 10^{-8} ohm-cm^2 based on actual resistance measurement [15], and the capacitance is estimated as the capacitance of a 1um-by-1um contact using the top metal layer and the height of the interlayer via is assumed to be 10um. Based on the benefits of transferring to 3D, we evaluated the performance impact of 3D microprocessor with some applications from SPEC2000 benchmark suite.

4.1 Issue scheduler

Figure 8 shows the latency benefits for tag-drive by using different numbers of strata. For the horizontal partitioning, we observed the latency improvement of 44% when moving from 2D to 2-strata 3D implementations. We also noticed the improvement of 22% from two to three strata and an additional 16% improvement with the move from two to four strata. For the vertical partitioning, we only show the result of 4% improvement for 2 strata because adding more device layers has little benefit. Therefore, only the horizon-

tal partitioning will be considered in the following experiments. Also based on this figure, the delay of the tag-drive scales linearly as the number of window sizes increases for two strata. However, in some cases the performance is the same. This is because the number of window entries located on different strata are the same in some cases and thus results in the same tag-drive latency. For example, window size 40 has partitions of 3, 3, and 4 sections of window entries and each of the section has four window entries (due to the selection logic structure) whereas window size 48 has partitions of 4, 4, and 4. So both of window sizes 40 and 48 have the tag-drive delay of travelling 4 sections of window entries and reaching down the tag comparison cells.

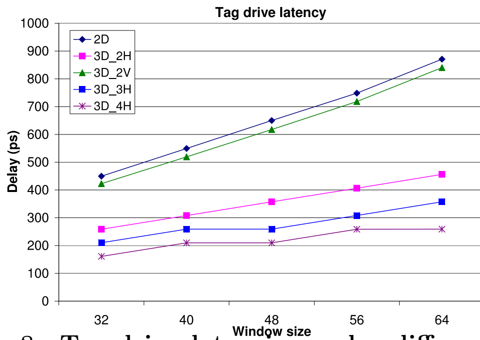

Figure 8: Tag drive latencies under different stratum configurations.

Since a significant portion of the overall delay of an issue logic comes from the tag-drive, we have evaluated the delay reduction obtained from 3D integration and the result is shown in Figure 9. We show only the results of window size larger than 32 with issue width of 8 because the smaller window sizes also exhibit the same trend and window size larger than 32 is more realistic in current generation of processor designs. The difference of a single loop delay (wake-up and select) is obvious between 2D and 3D implementations. The average delay reduction is 23% across all five window sizes when comparing 2D and 2-strata 3D implementations. Additional reductions, 6% and 10%, can be observed when implemented upon three and four strata. Note that, the selection logic is pitch-matched to the wake-up logic and 3D vias are added as needed. The delay time of the selection logic is not changed significantly in 3D due to its log_4 structure.

Figure 10 shows the power comparison between 2D and 3D implementations. We can easily see, from 2D and 2-strata 3D implementations, the power is effectively lowered; going beyond two strata also benefits on power reduction except slightly. The average power reduction is 16% for all five window sizes with two strata, whereas additional reductions, 6% and 8% are obtainable with three and four strata implementations.

Based on the preceding results, we observe that the issue logic is a wire-bound structure. Thus, the move from 2D to 3D is essentially helpful in relieving the wire-related impacts.

4.2 3D arithmetic modules

Table 1 shows the power and performance results of both the KS adder and the log shifter when implemented on 2D and 3D. The performance improvements of 16-bit Kogge Stone placed in two, three and four strata over the 2D design

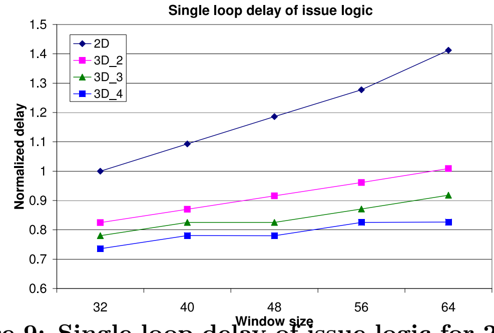

Figure 9: Single loop delay of issue logic for 2D and 3D implementations.

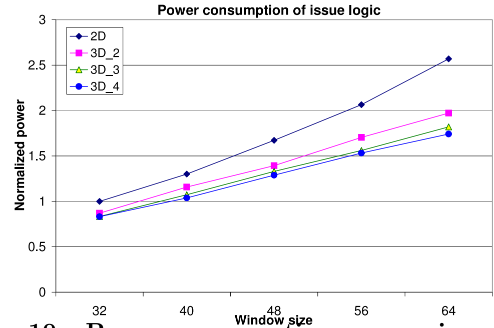

Figure 10: Power consumption comparison of 2D and 3D implementations.

are 20.238%, 23.611%, and 32.738%, respectively. Similarly, the corresponding power reductions are 8.14%, 14.67%, and 22.24%. Unlike the other components, we have only considered a 2-strata implementation for the log shifter because only marginal benefit is observed when going beyond two strata. However, we also considered the 32-bit log shifter to demonstrate the potential improvements. The same improvements are noticeable for the 16-bit log shifter on 2-strata 3D with 13.39% on performance whereas 14.28% on power, and for the 32-bit log shifter, the improvements are 28.39% and 6.99%, respectively.

	16-bit KS adder		Log 16		Log 32	
	delay(ps)	power(mw)	delay	power	delay	power
2D	504	0.87	224	0.88	398	2.0
2-strata	402	0.80	194	0.75	285	1.86
3-strata	385	0.74				
4-strata	339	0.68				

Table 1: 2D and 3D implementations of adder and shifter.

4.3 Performance impact

Although it is possible to implement microprocessors on multiple dies, we only consider the performance impact from a 2-strata processor in which we observe the largest gain.From the results above, we observe that the latency of the issue logic can be reduced by 23%, while 20% and 28% delay reduction is achieved for KS adder and 32-bit shifter, respectively, in a 2-strata 3D technology. Based on the recent results reported in [8, 9], the cache can be clocked 10%∼13% faster when implemented upon a 2-strata 3D architecture. Since the structure of a register file is similar to that of cache, we assume similar benefits can be obtained. We also assume the Load/Store queue has the same latency reduction as in issue logic due to their similarities. According to the suggestion in [14], duplication can be beneficial in helping latency; We have thus enlarged certain structures in 3D and have assumed their latencies to equal that of a

corresponding smaller structure in 2D. Table 2 lists the parameters for both 2D and 3D processors.

We used an architectural level simulator, SimpleScalar, with applications from the SPEC2000 benchmark suite to evaluate the performance impact. The result is shown in Figure 11. As can be observed from the figure, the enlarged structures and the lowering latencies in 3D can effectively extract more IPC compared to the conventional 2D implementation of processors. The average IPC speedup is 11% across all five applications; however, we believe more improvements can be achieved if more 3D-optimized components can be incorporated.

	2D	**3D**
Issue width	8	8
Window size	32	64
ROB Size	128	128
IL1,DL1	32KB,3 cycle	64KB,3 cycle
Register File	128	128
Load/Store Queue	16	32
Unified L2	1MB,8 cycle	1MB,7 cycle

Table 2: Processor parameters for 2D and 3D implementations.

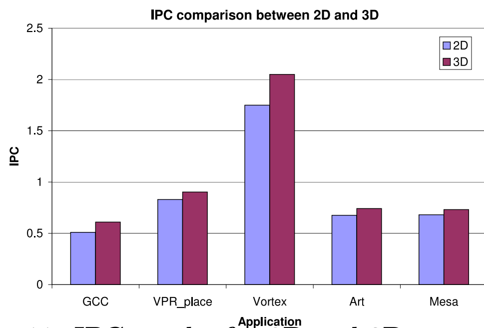

Figure 11: IPC results for 2D and 3D processors.

5. CONCLUSION

The 3D technology can reduce wire length effectively and this technique is especially prominent for wire-bound functional units in bringing power down from charging and discharging long wires. In this paper, we have explored the potential benefits of a few components of a microprocessor when implemented on 3D. From our experimental results, both delay and power can be brought down as the number of active layers used increases. Based on the latency improvement, we evaluated the performance impact through architectural level simulator with SPEC2000 applications and the result shows an average speedup of 11% can be achieved compared to the conventional 2D implementation of microprocessor.

6. REFERENCES

[1] Eric Borch and Eric Tune. Loose Loops Sink Chips. In *GPA*, 2002.

[2] P. Kapur, G. Chandra, and K. C. Saraswat. Power Estimation in global interconnects and its reduction using a novel repeater optimization methodology. In *DAC*, 2002.

[3] V. Nookala and S. S. Sapatnekar. Correcting the functionality of a wire-pipelined circuit. In *DAC*, 2004.

[4] S. Palacharla, N. P. Jouppi, and J. E. Smith. Complexity-Effective Superscalar Processors. In *ISCA*, 1997.

[5] S. Das, A. Fan, K.-N. Chen, C. Tan, N. Checka, and R. Reif. Technology, Performance, and Computer-Aided Design of Three-Dimensional Integrated Circuits. In *ISPD*, 2004.

[6] B. Black, D. W. Nelson, C. Webb, and N. Samra. 3D Processing Technology and Its Impact on IA32 Microprocessors. In *ICCD*, 2004.

[7] J. W. Joyner and J. D. Meindl. Opportunities for Reduced Power Dissipation Using Three-Dimensional Integration In *Proc. of Interconnect Technology Conference*, 2002.

[8] Y.-F. Tsai, Y. Xie, N. Vijaykrishnan, and M. J. Irwin. Three-Dimensional Cache Design Exploration Using 3DCacti. In *ICCD*, 2005.

[9] K. Puttaswamy and G. H. Loh. Implementing Caches in a 3D Technology for High Performance Processors. In *ICCD*, 2005.

[10] M. Ekpanyapong, et al. Profile-guided microarchitectural floorplanning for deep submicron processor design In *DAC*, 2004.

[11] A. Jagannathan, et al., Microarchitecture Evaluation With Floorplanning And Interconnect Pipelining. In *ASPDAC*, 2005.

[12] R. Reif, et al. Fabrication Technologies for Three-Dimensional Integrated Circuits. In *ISQED*, 2002.

[13] Z. Huang and M. D. Ercegovac. Wire delay analysis in deep-submicron prefix adder design. In *Proc. Asilomar Conf. on Signals, Systems and Computers*, 2000.

[14] David A. Patterson. Latency Lags Bandwidth. In *Communication of the ACM*, Oct. 2004/vol.47, No.10.

[15] K. N. Chen, et al. Contact Resistance Measurement of Bonded Copper Interconnects for Three-Dimensional Integration Technology. In *IEEE Electron Devices Letters*, 25(1), 2004.

[16] Tezzaron Semiconductor. http://www.tezzaron.com

[17] http://www-device.eecs.berkeley.edu/∼bsim3