A Criticality-Driven Microarchitectural Three Dimensional (3D) Floorplanner

Srinath Sridharan Michael DeBole Guangyu Sun Yuan Xie Vijaykrishnan Narayanan
Dept. of Computer Science and Engineering
The Pennsylvania State University
University Park, PA 16802
{ssridhar, debole, gsun, xie, vijay}@cse.psu.edu

Abstract - As technology scales, interconnect delays begin to dominate the performance of modern microprocessors. The ability to reduce the length of global wires has become an important design constraint, however only a subset of those global wires is critical for determining performance. The introduction of three-dimensional (3D) ICs has created the opportunity to reduce global wiring lengths and shorter interconnect delays through the intelligent placement of functional blocks. In this paper, a floorplanner for 3D chips is proposed that organizes functional blocks according to critical microarchitectural communication paths. The floorplanner identifies the potential triggers, in the form of feedback delays, which are responsible for the largest communication costs and places the contributing functional blocks in such a way that those costs are minimized. With our criticality driven 3D placement there is an average IPC improvement of 22% over 2D placement. Over criticality un-aware 3D placement, criticality driven 3D placement shows an average IPC improvement of 8%.

I. INTRODUCTION

As technology scales, interconnect delays begin to play a key role in processor performance. In fact, as wire cross-sections decrease, resistance increases, causing longer propagation delays and reduced performance. This increase in communication delay between components will become significant enough that, in the future, more time will be spent communicating information then performing computation [1, 2]. For example, at 35nm, a chip operating at 5GHz will only have signals capable of travelling 1.4% of the chip area before suffering significant signal degradation [1].

Directly impacted by the increased wiring delays are the microarchitectural feedback loops found in superscalar processor architectures. Some of these loops, shown in Fig. 1, are critical for the communication of speculative information between one stage of the pipeline and the same, or earlier, stages of the pipeline [3]. These loops typically have an impact on the performance of the processor because the feedback delay, in-part, determines the mis-speculation loop latency. This situation is further complicated by the inclusion of multiple feedback loops within the pipeline architecture, which provide the processors with the ability to perform multiple speculations simultaneously.

Meanwhile, the emergence of 3D chip technology promises to reduce the delays of the global interconnect by vertically stacking multiple dies and connecting them with a shorter, vertical interconnect. This moderates the wiring delay problem by replacing the long global wires with vertical ones that are much shorter in length. Furthermore, as 3D package manufacturing has become more mature, many of the concerns with thermal viability and the reliability of inter-wafer vias have been resolved, making 3D ICs a feasible alternative for future IC's.

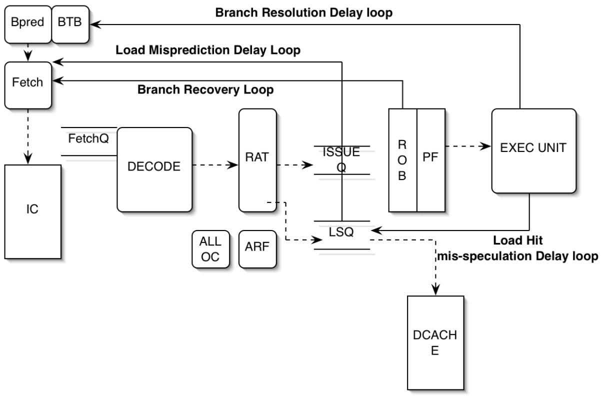

Fig. 1. Microarchitectural Feedback Loops

In this paper a criticality driven, three-dimensional (3D) floorplanning algorithm for deep submicron superscalar processors, that organizes the microarchitectural blocks according to critical communication paths, is proposed. Previous work in 3D microprocessor floorplanning have demonstrated that such and approach can lead to a reduction on overall wire length [4, 5]. However, in modern microprocessors, not all the global interconnects are equally important. For those wires that are on the critical path, wire reduction via 3D stacking can result in substantial performance improvements, while reducing wire lengths that are not on the critical path will have relatively no impact on performance. The floorplanner, which is implemented using a simulated annealing technique, uses a cost function that takes into consideration the wire criticality, chip area, and overlap power density.

This paper is organized as follows: A review of related work is provided in Section 2. A short background on superscalar feedback loops is given in Section 3. Section 4 presents the criticality-driven floorplanner. In Section 5, a description of the experimental setup is given and Section 6 provides the results. Finally the paper is concluded in Section 7.

II. RELATED WORK

In the past, there has been a significant amount of work looking into multi-objective floorplanning. In [4], Micheal, et. al. proposed a multi-objective floorplan which attempted to maximize performance while trying to minimize the footprint and peak temperature simultaneously. In [6], Hung, et. al. presented a 3D

978-1-4244-2749-9/09/$25.00 ©2009 IEEE

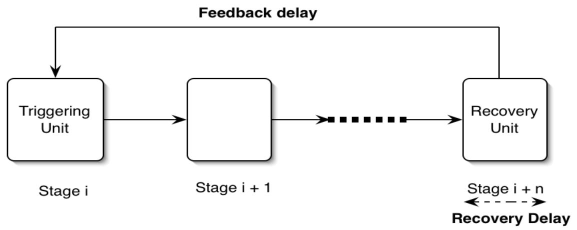

Fig. 2. Feedback delay loop

floorplanning algorithm. However, none of these techniques took into consideration the feedback delay in order to further improve performance. In the case of [4], [7], and [5], area and wire length optimizations are considered, but this does not necessarily translate into minimum feedback delays, as demonstrated in the results section.

[8], [9], and [10] attempted to optimize multiple critical paths, or loops, in the micro-architecture by inserting the pipeline stages optimally in the blocks and by placing wires, of those loops, on a 2D plane in order to meet the clock frequency requirements. In contrast to inserting pipeline stages, in this paper, the floorplanner exploits the additional dimension of proximity provided by 3D technology to reduce the length of critical paths.

In [11] sets of wires are identified that are frequently used during cycle-accurate, architectural simulation which are then used to guide the subsequent floorplanning process to effectively optimize those wires. However, frequently used wires are not necessarily critical wires.

There has also been work exploring the impact that different pipeline lengths and configurations have on processor performance [3, 12]. The authors of [3] proposed a Distributed Register Algorithm (DRA) aiming to reduce the length of the loose loops for critical portions of the pipeline by improving upon the recovery delay. This differs from the 3D floorplanner in this paper which attempts to reduce the feedback delay with intelligent, 3D placement of microarchitectural blocks.

In [13], Awasti et. al. provides a criticality analysis of feedback loops in an Alpha 21264 based, out-of-order processor. While the goal of the floorplanner found in [13] is similar to the one in this paper, the methodology adopted here is fundamentally different. [13] employs weighted wire delay as a cost function to represent criticality and assigns an instructions-per-cycle (IPC) penalty as weights to the wire delay cost which is estimated by predicting the *theoretical* IPC slowdown for each set of critical wires. In contrast, this paper uses a metric which incorporates a physically measurable quantity (wire-length) to be used with the floorplanner. Further In this paper, IPC is used only to evaluate the quality of the placement and not as a metric to the floorplanner.

Similar to [13], the MEVA-3D [14] framework also annotates the critical path with latency measurements in order to predict the IPC degradation due to extra latency along the critical path.

III. Background

A critical operation of high performance superscalar architectures is that of prediction and speculation. Arguably more important is the ability to communicate that information to different stages of the pipeline. In the case of correct speculation there is no penalty incurred on performance by communication latencies. However, In the case of mis-predictions, the performance relies significantly on the time to communicate the correct result and recover from the incorrect execution. This time, known as the mis-prediction loop latency, is a function of two components, the *recovery delay* and *feedback delay*. The *recovery delay* is defined as the time taken by the recovery unit to revert back to the closest, non-speculative state. The *feedback delay* is defined as the time needed to communicate the information from the recovery unit to the triggering unit. In this context, the triggering unit is the module where speculation is originally initiated and where the mis-speculated instruction is later resumed. An illustration of this mechanism is shown in Fig. 2.

Shown in Fig. 3 is the importance of the mis-speculation latency on the performance (IPC) of a general superscalar processor containing one feedback loop. This graph demonstrates that as the latency increases to 8 cycles (normalized to 1 cycle) the performance observed degrades on the average of 10.38% across all 26 SPEC CPU 2000 benchmarks. The exceptions (Swim, Mgrid, and Lucas) show less then 1% degradation due to either high branch prediction accuracy or a small number of branches.

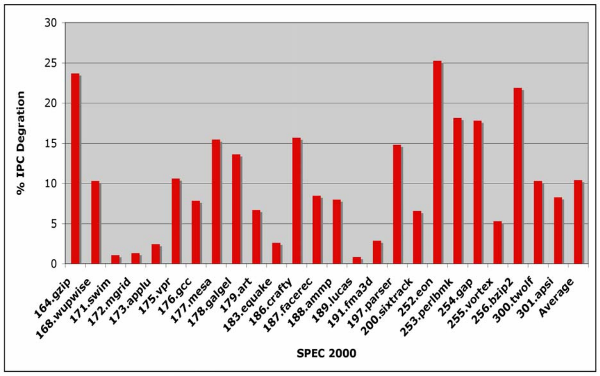

Fig. 3. IPC Degradation in percentage as the branch mis-speculation feedback delay increases from 1 to 8 cycles

Feedback delay has not traditionally been one of the more dominating factors, but with the scaling of technology the impact of feedback delay is becoming as important as recovery delay. As a result, it is highly imperative to be able to create floorplans which can improve upon the latency of the feedback delay loops.

TABLE I

MICROARCHITECTURAL LOOPS

Feedback Loops	Microarchitectural blocks
Branch Resolution loop	ALU to BTB
Branch recovery loop	ROB to Fetch
Forwarding loop	ALU to Issue queue
Load mis-prediction loop	LSQ to fetch
Load resolution loop	D-Cache to LSQ

IV. CRITICALITY-DRIVEN FLOORPLANNER

The criticality driven floorplanner is implemented on top of the 3D thermal-aware floorplanner proposed in [6]. This floorplanner uses a simulated annealing engine to partition and perturb the blocks across the layers in order to satisfy thermal, wiring, and area constraints. The primary contribution of this work is the modification of the wire length cost function to provide higher weights to critical nets forming part of the feedback loops. The netlists fed to the floorplanner are obtained from the HDL model of a modern superscalar architecture [15] similar to Alpha 21264. Table 1 identifies the four feedback loops modeled in this work and the associated blocks which are connected to nets forming these loops.

The input to the floorplanner is the area of all functional modules and a single bit of information determining whether the block is critical or not. If two blocks are critical and are connected together then those wires become critical and are denoted the connecting wires, C_{wires}. The remaining wires are denoted by NC_{wires}. The importance of C_{wires} and NC_{wires} is given by the weight function, Ψ. The cost of criticality, $Cost_{Critical}$ is:

$$\begin{gathered} Cost_{Critcal} = \psi \sum C_{wires} + (1-\psi) \sum NC_{wires} \\ \Psi \leq 1 \end{gathered} \tag{1}$$

$Cost_{Critical}$ is framed with the intuition that just minimizing the overall wire length may not be sufficient for increasing speedup. Identifying and minimizing the lengths of C_{wires} might have a greater impact on performance than just optimizing on the overall wire length. In the results, the effectiveness of this simple metric is demonstrated by performing a comparison with the objective function used in prior work [4, 5]. The value of Ψ was carefully chosen to ensure that there is no unwarranted increase in the wire length of NC_{wires}, which can otherwise result in performance degradation. To guarantee that NC_{wires} played a significant role in determining $Cost_{Critical}$, Ψ was chose to be *0.70* for all the experiments and the sensitivity analysis is shown in Section 6.

In addition to the criticality metric, area cost, deviation of area in different layers, and temperature were factored in through the following cost function:

$$\begin{aligned} Cost_{Total} &= \alpha * area + \beta * Cost_{Critical} \\ &+ \gamma * dev(f) + \sigma * TOP \end{aligned} \tag{2}$$

Where area is the chip area, *dev(F)* is the dimension deviation factor as mentioned in [6, 16] and α, β, γ and σ are weights such that $\alpha + \beta + \gamma + \sigma = 1$. *TOP* is the total overlap power density and it is given by $TOP = \sum OP(T, P_i)$.

Where $OP(T, P_i)$ is the summation of the power densities of module P_i and it is found through the following equation [6]:

$$OP(T, P_i) = \sum (P_m + P_{mi}) * Area_{overlap} \tag{3}$$

Where P_m is the power density of module *m* and P_{mi} are the power densities of the module overlapping with module *m*.

The floorplanner treats each block as a soft module and uses the cost function in perturbing the locations and dimensions of the blocks.

V. EXPERIMENTAL SETUP

The microarchitectural configuration used in all the experiments is shown in Table 2. The design flow, shown in Fig. 4, takes into consideration performance, individual dynamic block power, leakage power, and thermal analysis. The inputs to the floorplanner are the criticality, delay, and area estimates of each module along with their power values. All circuits are assumed to be fabricated in 35nm technology. The dynamic and leakage power consumption of each module was estimated using Sim-wattch [17].

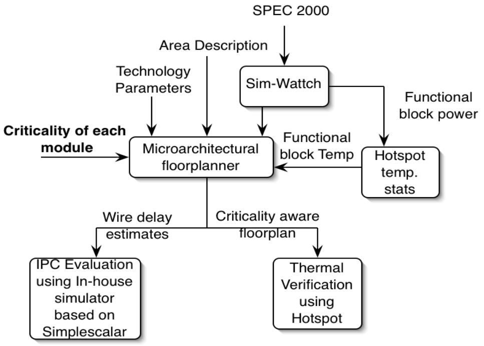
SPEC 2000
Area Description
Technology Parameters
Sim-Wattch
Functional block power
Criticality of each module
Microarchitectural floorplanner
Functional block Temp
Hotspot temp. stats
Wire delay estimates
Criticality aware floorplan
IPC Evaluation using In-house simulator based on Simplescalar
Thermal Verification using Hotspot

Fig. 4. Overview of the floorplanner

The output of the floorplanner is the criticality-aware floorplan along with the wire length values between the microarchitectural modules. To calculate the wire delay for a given wire length *L*, the following wire delay model from [1] and [18] is used:

$$D_{wire} = \frac{L}{l_0}\left(R_0\left(C_g + C_W\right) + \rho + R_W\left(\frac{C_w}{2} + C_g\right)\right) \tag{4}$$

Where R_0 is the on-resistance of the repeater, C_g is the gate capacitance of the repeater, l_0 is the length of the wire segment between repeaters, ρ is the intrinsic delay of the repeater, and R_w and C_w are the resistance and capacitance of the wire segment between two repeaters. Each repeater is assumed to be an inverter with PFET and NFET sizes chosen to minimize the overall wire delay. For example, the

TABLE II
MICROARCHITECTURAL PARAMETERS

Parameter	Value
Fetch/Decode/Issue/Commit Width	8
Pipeline Stages	15
Fetch Queue size	128 entries
ROB Size	512 entries
PRF size	512 entries
LSQ Size	128 entries
L1 D-Cache	64KB, 4-way with 32B block
Branch Predictor	Combined predictor with 16K entry meta-table. 2-lev predictor with 16K-entry L1, 16K-entry L2, 14-bit history XORed with address
L1 I-Cache	64KB, 4-way with 32B block
L2 Unified Cache	512 KB, 4-way with 64B line-size
I-TLB	512-entries 4-way set-associative
D-TLB	1K-entries 4-way set-associative
TLB Miss-Latency	30 cycles
Memory Latency	200 cycles
Integer ALU	6 (1-cycle latency)
Integer Multipliers/Dividers	2 (3,20 cycles respectively)
FP ALUs	4 (2 cycles)
FP Mult./Div./Sqrt.	2 (4,12,24 cycles respectively)
L1 D-Cache Ports	4
BTB Size	2K-entry 4 way
Store Set Predictor	4K entries
Clock frequency	3GHz

delay incurred for a 10mm top-level wire, in 35nm technology, is 1560ps. Assuming a 3GHz clock frequency, the number of clock cycles incurred for wire delay is 5 cycles. This clock cycle value is fed to a cycle accurate, in-house simulator based on Simplescalar 3.0 [19], which estimates and analyses the impact on performance. The criticality aware floorplan is fed to the Hotspot 3.0 tool [20] to verify the peak temperature of the design. The SPEC CPU 2000 benchmark suite was used for all experiments.

VI. RESULTS

For each benchmark, the simulator was warmed up using the first 3 billion instructions and then simulated for 100 million instructions. The baseline 3D ($3D_{base}$) floorplan is shown in Fig. 5. The floorplan is generated based on the algorithm defined in [6]. Fig. 6 shows the criticality placed ($3D_{critical}$) floorplan with the cost function accounting for feedback delay. The weight functions for both $3D_{base}$ and $3D_{critical}$ are shown in Table 3. Note that in the $3D_{base}$ case, all wires are assumed to be equally critical. In $3D_{critical}$, microarchitectural blocks that are critical are placed as close together as possible to negate the adverse effect on performance. At the same time, the floorplanner ensures that they are far enough apart to avoid thermal runaway. For example, in $3D_{critical}$, the units: *ROB*, *Fetch0*, *Fetch1*, and *Fetch2* are placed together in order to curb the negative effect triggered by the branch mis-speculation feedback delay. Similarly, *LSQ* is placed closer to the fetch units to mitigate the impact of the load mis-speculation feedback delay. In $3D_{base}$, these critical units are placed far apart in-order meet the area constraint.

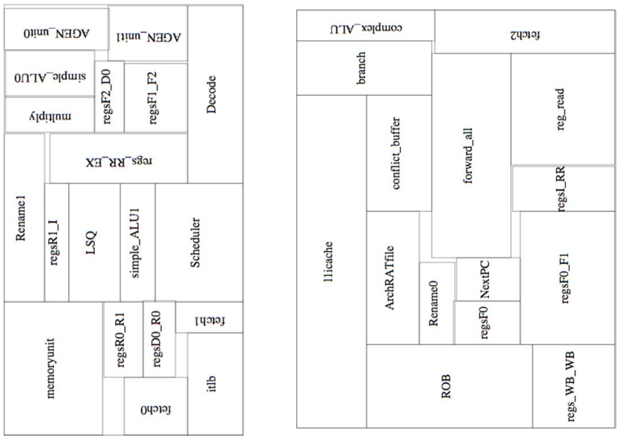

Fig. 5. Two layers of $3D_{Base}$ Floorplan

Since the floorplanner also take into account the overlap power density cost, $3D_{critical}$ does not exhibit any thermal runaway. The peak temperature values from Table 5 justifies that $3D_{critical}$ is devoid of thermal runaway. The overall wire lengths obtained for $3D_{critical}$ and $3D_{base}$ were 243.12mm and 212.73mm respectively. However, optimizing on the overall wire length does not guarantee an improvement in performance.

Although the overall wire length of $3D_{critical}$ is greater than that of $3D_{base}$, with the criticality based placement there is an IPC improvement on an average of 8% and up to 25% of $3D_{critical}$ over $3D_{base}$. Over $2D_{base}$, $3D_{critical}$ shows

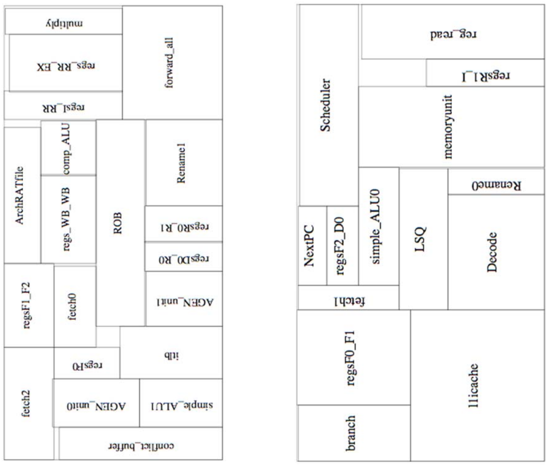

Fig. 6. Two layers of $3D_{Critical}$ Floorplan

TABLE III

WEIGHT FUNCTIONS

Floor plan	Weight functions
$3D_{base}$	α=0.25, β=0.25, γ=0.25, σ=0.25, Ψ=0
$3D_{critical}$	α=0.25, β=0.25, γ=0.25, σ=0.25, Ψ=0.70

an improvement on an average 22% and up to 64%. This proves the effectiveness of the criticality metric. Figure 7 shows the IPC improvement of $3D_{critical}$ over $2D_{base}$ and $3D_{base}$.

In $3D_{critical}$, *swim, mgrid, applu, mcf,* and *lucas* show a negative improvement in performance over $3D_{base.}$ This can be explained through the fact that the floorplan is primarily optimized according to the feedback delay. Therefore, in some cases, non-critical wire lengths increase in the pipeline causing a decrease in overall performance when these paths dominate. This means that in the cases of high prediction accuracy, where the non-critical path delay primarily determine performance, there is an effective decrease in the performance of the microarchitecture. Since these benchmarks exhibit high prediction accuracy, the non-critical paths effectively determine the performance and it results in a negative IPC improvement.

A. Impact on wire length and IPC for varying Ψ

Table 4 demonstrates the impact on performance across varying Ψ values. As Ψ increases, the ratio of critical wire length to overall wire length decreases. While the average IPC is expected to increase as the Ψ value increases, for Ψ=0.8 the average IPC degrades due to the domination of the non-critical wire lengths.

B. Impact of uneven weight to cost functions

To demonstrate the sensitivity of the criticality metric, the weight function β of $Cost_{Critical}$ was varied as follows:

Case 1: β varies from 0.25 to 0.35
($\alpha = \gamma = \sigma$ and $\alpha + \beta + \gamma + \sigma = 1$)

Case 2: β varies from 0.25 to 0.15
($\alpha = \gamma = \sigma$ and $\alpha + \beta + \gamma + \sigma = 1$)

Fig. 8 reports the results for β (in given range) that has the best and worst impact on IPC for cases 1 and 2, respectively. As β decreases, there is 5% degradation in performance over $3D_{critical}$ on average. However, reducing β can potentially optimize the area and reduce the overall wire length. On the other hand as β increases, there is a 6.2% increase in IPC over to $3D_{critical}$ on average. However, a higher β is not preferred because it can compromise on other optimization metrics. In the case of *186.crafty*, the peak temperature increases to 383K from 350K for a β of 0.25. As a result, the parameters can be varied to meet the desired design specification.

TABLE IV

IMPACT ON WIRE LENGTH AND IPC DUE TO DIFFERENT Ψ VALUES OVER $3D_{Critical}$

Ψ value	Overall wire length(mm)	Critical to overall wire length ratio	Average IPC
0.50	219	0.34	1.70
0.60	231	0.29	1.72
0.70	243	0.24	1.79
0.80	250	0.21	1.70

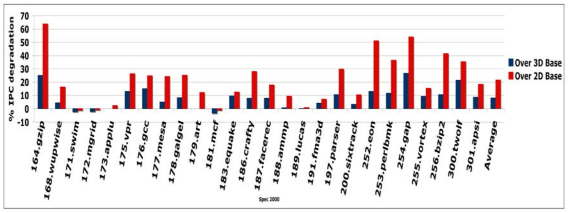

Fig. 7. IPC improvement of $3D_{Critical}$ over $2D_{base}$ and $3D_{base}$

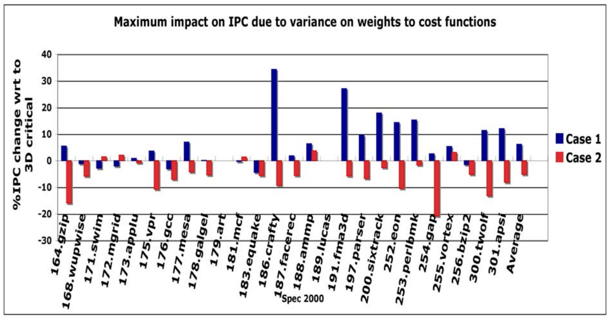

Fig. 8. Maximum impact on IPC due to increase or decrease in the value of β over $3D_{Critical}$

VII. CONCLUSION

Providing a micro-architectural floorplanner that can exploit the strengths of 3D integration by taking into account the communication cost between critical modules can provide rich dividends in terms of performance. In this paper, a 3D criticality-driven microarchitectural placement technique is demonstrated which identifies feedback delays that are responsible for incurring high communication costs. By intelligently placing the functional blocks in 3D, performance is improved without compromising on area, overlap power density, or thermal reliability. Using the 3D floorplanner proposed in this paper an IPC improvement of 22% is achieved over a conventional 2D floorplanner on average, with an 8% improvement over a criticality unaware 3D placement on average.

VIII. ACKNOWLDEGEMENTS

We would like to thank the anonymous reviewers whose comments helped improve the quality of the paper. This research was funded partly by IBM's Faculty Award, NSF grant 0702617, and NSF Career Award 0643902.

TABLE V

Peak Temperature(°K) Values of $2D_{base}$, $3D_{base}$, and $3D_{critical}$

Benchmark	$\mathbf{2D_{base}}$	$\mathbf{3D_{base}}$	$\mathbf{3D_{critical}}$
164.gzip	347.29	366.41	375.50
168.wupwise	342.12	359.87	367.83
171.swim	342.19	358.49	366.24
172.mgrid	344.65	362.24	370.62
173.applu	339.58	354.39	361.42
175.vpr	327.73	335.90	339.76
176.gcc	333.33	344.61	349.96
177.mesa	346.52	365.13	373.97
178.galgel	345.81	364.09	372.74
179.art	333.81	345.51	350.00
181.mcf	320.54	324.74	326.71
183.equake	332.28	342.01	348.07
186.crafty	333.57	345.02	350.44
187.facerec	343.56	360.57	368.64
188.ammp	334.03	345.72	351.26
189.lucas	329.14	338.17	342.43
191.fma3d	334.17	345.94	351.51
197.parser	335.06	347.32	353.12
200.sixtrack	350.84	371.90	381.96
252.eon	334.47	346.41	352.05
253.perlbmk	333.94	345.61	351.12
254.gap	336.31	349.27	355.41
255.vortex	338.42	352.56	349.23
256.bzip2	353.91	376.61	387.43
300.twolf	327.62	335.77	339.62
301.apsi	333.99	345.69	351.22

VIIII. REFERENCES

[1] V. Agarwal, M. S. Hrishikesh, S. W. Keckler, and D. Burger, "Clock rate versus IPC: the end of the road for conventional microarchitectures," presented at *Proceedings of the 27th International Symposium on Computer Architecture*, June 2000.

[2] R. Ho, K. W. Mai, and M. A. Horowitz, "The future of wires," *Proceedings of the IEEE*, vol. 89, pp. 490, 2001.

[3] E. Borch, E. Tune, S. Manne, and J. Emer, "Loose loops sink chips," presented at High-Performance Computer Architecture, 2002. Proceedings. Eighth International Symposium on, 2002.

[4] M. Healy, M. Vittes, M. Ekpanyapong, C. S. Ballapuram, S. K. Lim, H. H. S. Lee, and G. H. Loh, "Multiobjective Microarchitectural Floorplanning for 2-D and 3-D ICs," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 26, pp. 38, 2007.

[5] J. Cong, W. Jie, and Z. Yan, "A thermal-driven floorplanning algorithm for 3D ICs," presented at Computer Aided Design, 2004. ICCAD-2004. IEEE/ACM International Conference on, 2004.

[6] W. L. Hung, G. M. Link, X. Yuan, N. Vijaykrishnan, and M. J. Irwin, "Interconnect and thermal-aware floorplanning for 3D microprocessors," presented at Quality Electronic Design, 2006. ISQED '06. 7th International Symposium on, 2006.

[7] K. Balakrishnan, V. Nanda, S. Easwar, and L. Sung Kyu, "Wire congestion and thermal aware 3D global placement," presented at Design Automation Conference, 2005. Proceedings of the ASP-DAC 2005. Asia and South Pacific, 2005.

[8] M. R. Casu and L. Macchiarulo, "Throughput-driven floorplanning with wire pipelining," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 24, pp. 663, 2005.

[9] M. Yuchun, L. Zhuoyuan, C. Jason, H. Xianlong, G. Reinman, D. Sheqin, and Z. Qiang, "Micro-architecture Pipelining Optimization with Throughput-Aware Floorplanning," presented at Design Automation Conference, 2007. ASP-DAC '07. Asia and South Pacific, 2007.

[10] V. Nookala and S. S. Sapatnekar, "Designing optimized pipelined global interconnects: algorithms and methodology impact," presented at Circuits and Systems, 2005. ISCAS 2005. IEEE International Symposium on, 2005.

[11] M. Ekpanyapong, J. R. Minz, T. Watewai, H. H. S. Lee, and L. Sung Kyu, "Profile-guided microarchitectural floor planning for deep submicron processor design," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 25, pp. 1289, 2006.

[12] E. Sprangle and D. Carmean, "Increasing processor performance by implementing deeper pipelines," presented at Computer Architecture, 2002. Proceedings. 29th Annual International Symposium on, 2002.

[13] M. Awasthi, V. Venkatesan, and R. Balasubramonian, "Understanding the Impact of 3D Stacked Layouts on ILP," *The Journal of Instruction-Level Parallelism (JILP)*, vol. 9, June 2007.

[14] J. Cong, A. Jagannathan, M. Yuchun, G. Reinman, W. Jie, and Z. Yan, "An automated design flow for 3D microarchitecture evaluation," presented at Design Automation, 2006. Asia and South Pacific Conference on, 2006.

[15] "http://www.rhc.uiuc.edu/ACS/tools/ivm/about.html."

[16] P. H. Shiu, R. Ravichandran, S. Easwar, and S. K. Lim, "Multi-layer floorplanning for reliable system-on-package," presented at Circuits and Systems, 2004. ISCAS '04. Proceedings of the 2004 International Symposium on, 2004.

[17] D. Brooks, V. Tiwari, and M. Martonosi, "Wattch: a framework for architectural-level power analysis and optimizations," presented at Computer Architecture, 2000. Proceedings of the 27th International Symposium on, 2000.

[18] B. S. Amrutur and M. A. Horowitz, "Speed and power scaling of SRAM's," *Solid-State Circuits, IEEE Journal of*, vol. 35, pp. 175, 2000.

[19] D. Burger and T. Austin, "The SimpleScalar Toolset, Version 3.0. http://www.simplescalar.com."

[20] K. Sankaranarayanan, S. Velusamy, M. R. Stan, and K. Skadron, "A Case for Thermal-Aware Floorplanning at the Microarchitectural Level," *The Journal of Instruction-Level Parallelism (JILP)*, Sept. 2005.