A Unified Methodology for Power Supply Noise Reduction in Modern Microarchitecture Design

Michael Healy, Fayez Mohamood, Hsien-Hsin S. Lee, and Sung Kyu Lim
School of Electrical and Computer Engineering
Georgia Institute of Technology
{mbhealy, fayez, leehs, limsk}@ece.gatech.edu

Abstract—**In this paper, we present a novel design methodology to combat the ever-aggravating high frequency power supply noise (di/dt) in modern microprocessors. Our methodology integrates microarchitectural profiling for noise-aware floorplanning, dynamic runtime noise control to prevent unsustainable noise emergencies, as well as decap allocation; all to produce a design for the average-case current consumption scenario. The dynamic controller contributes a microarchitectural technique to eliminate occurences of the worst-case noise scenario thus our method focuses on average-case noise behavior.**

I. Introduction

The power supply noise issue has become one of the leading concerns for the microprocessor industry [1]. With advances in process technology, power consumption and wire resistance have gone up while voltage, and thus noise tolerance, have gone down. Aggressive power dissipation limiting techniques are being implemented at an ever increasing rate. Low-power techniques such as clock/power gating and frequency/voltage scaling have been widely adopted. Although clock gating is an effective technique to control power dissipation, it suffers from the introduction of large high frequency power supply noise effects. To mitigate these undesirable effects, designers control the impedance of the power distribution network and insert sufficient amounts of decoupling capacitors (decaps), typically targeting the worst-case scenarios, throughout the entire chip to guarantee functioanl reliability that may be compromised by current surges.

The trend of ever increasing power consumption and decreasing supply voltages in future processor generations will cause this worst-case design strategy to become an insurmountable obstacle to progress. Performance will decrease due to power-on lag as fine-grained clock gating becomes increasingly necessary. Leakage power caused by the large decap requirements will eliminate the benefits of using clock gating, which is designed to reduce dynamic power consumption. Looking to these issues points the way to a design philosophy that avoids optimizing for the worst-case scenario and instead targets an average-case scenario, while leaving the avoidance of the causes of this worst-case to dynamic detection at the microarchitectural level.

The contribution of this paper is our novel design methodology that integrates architectural profiling, runtime noise controller, floorplanning, and decap planning as a holistic solution to address the ever-aggravating power supply noise problem while utilizing physical design for the average-case noise scenario. Our architectural profiler first identifies a subset of modules that are likely to draw a large amount of current simultaneously. Our floorplanner then tries to separate and evenly distribute those modules so that the current demand for the modules is met. A further contribution is that the physical basis of the dynamic noise controller is considered directly by the floorplanner for optimization. Our dynamic noise controller manages clock gating for the microarchitectural modules at runtime to dynamically limit the inductive noise by eliminating the occurence of worst-case behavior. This is the first work to directly consider IR drop and LdI/dt inductance noise with dynamic noise controller awareness that includes decap planning and allows for design for the average-case.

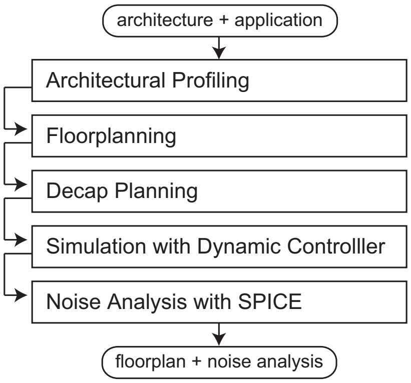
architecture + application
Architectural Profiling
Floorplanning
Decap Planning
Simulation with Dynamic Controlller
Noise Analysis with SPICE
floorplan + noise analysis

Fig. 1. Overview of our design flow

The rest of this paper is organized as follows. The overall design flow is presented in Section II. Our dynamic controller-aware floorplanning algorithm is presented in Section III. Section IV presents the experimentation and results. Next, Section V discusses this work's relation to previous works. Finally, conclusions are given in Section VI.

II. Unified Design Methodology

A. Design Flow

An overview of this work's design flow is shown in Figure 1. The input to the flow is an architectural description and a set of benchmark programs. The size of each module in the floorplan is estimated using GENESYS [2] and eCACTI [3]. The flow begins with cycle-level microarchitectural simulation using SimpleScalar [4] and integrated power consumption estimation using Wattch [5]. During this simulation the power consumption of each microarchitectural block, and its switching activity, are collected on a per-cycle basis. This collection

*This material is based upon work supported by the Focus Center for Circuit & System Solutions (C2S2), Semiconductor Research Corporation.

978-1-4244-1922-7/08/$25.00 ©2008 IEEE

is done without the dynamic noise controller activated. Next this switching activity information is utilized to optimize the floorplan. Next floorplaning is performed and a large set of candidate floorplans is generated during Simulated Annealing. Finally, decap planning is used to select the best among the candidates and this is the final floorplan used for evaluation.

B. Architectural Profiling

Architectural profiling is done using SimpleScalar, a cycle-accurate microarchitectural simulator. This work assumes aggressive and coarse-grained (module by module) clock gating on a cycle by cycle basis. Two statistics are collected about each module based on [6]. These statistics are the self-switching weight, and the correlation weight. The self switching weight is a normalized measure of how often a block changes power state and the correlated switching weight is a normalized measure of how often a pair of blocks switch in the same direction during the same cycle. Highly correlated blocks are likely to cause power noise problems and so should be placed far from each other in a noise optimized floorplan.

C. Floorplanning

This work uses Simulated Annealing based on the Sequence Pair [7] floorplan representation. Floorplanning can impact noise problems in a power distribution grid by moving noisier blocks both away from each other and closer to the power pins. The longer the current delivering path between a power pin and a noisy block is the more noise will be seen by that block's neighbors and by itself. Additionally, if the architecture utilizes a dynamic controller that is floorplan aware, as in this work, then it is possible to optimize the operation of that controller by providing it with a well formed floorplan. The dynamic controller specifically allows the floorplanner to target the average-case scenario and not the worst-case as is typical. This work uses a new annealing cost function that specifically targets three sources of noise, as well as the physical basis of the dynamic noise control algorithm. More details will be described in Section III-A.

D. Decap Planning

Decap planning in this work is based on [8], which utilizes a network flow based approach. Decap planning is used to select the best among a large group of the best, according to the cost function, floorplans found during annealing. The network flow is used to analyze how much decap each of the floorplans requires, and the floorplan with the smallest requirement is chosen as the overall best. All white space in the floorplan is utilized as decap in this work.

E. Runtime Noise Controller

Clock gating at the microarchitectural level is extensively used to control the total dynamic power dissipation of a processor. In this work a dynamic noise controller [6] based on a floorplan aware set of queues is implemented to address the noise problems created by clock gating. Each queue in the controller is based on modules close to each other in the floorplan and thus likely to cause noise problems for the other modules in the queue. The controller prevents modules within the same queue from switching in the same cycle and each modules request to power off is limited by a decay counter. The decay counter prevents modules that get used regularly but not constantly from switching on and off quickly in a short time. The worst-case noise scenario is when all modules attempt to switch simultaneously from one power state to another. The dynamic controller is designed and utilized in this work to eliminate occurences of this and other, less severe, worst-case noise behavior, thereby allowing for physical design targeted at the average-case.

III. Runtime dI/dt Controller-Aware Floorplanning Algorithm

Previous works have addressed the IR drop problem by including decap considerations during the floorplanning process. Other works have independently addressed the coupled dynamic inductance noise problem by separating blocks that switch during the same cycle. This is the first work to combine both direct IR drop consideration and LdI/dt dynamic noise consideration with dynamic controller awareness during a floorplanning process that specifically targets the average-case noise scenario. This is done through a combination of a novel design flow and a new cost function.

A. Annealing and Cost Function

A new annealing cost function is used that specifically targets three sources of noise, as well as the physical basis of the dynamic noise control algorithm. There are five terms in the cost function. The first two target traditional physical design objectives, area ($= A$) and wire-length ($= W$). The third term of the cost function addresses self induced inductance L and IR drop, ($= I$). Correlated switching factors are considered in the fourth term, ($= C$). And the final term, ($= Q$), includes the consideration of the dynamic noise control algorithm. The total cost function is given by:

$$Cost = \alpha \cdot A + \beta \cdot W + \gamma \cdot I + \delta \cdot C + \epsilon \cdot Q$$

where α,β,γ,δ, and ϵ are weighting constants. In this work the values for the weighting constants were empirically determined to be 1, 0.2, 0.5, 0.5 and 0.025 for α,β,γ,δ, and ϵ, respectively. The following sections describe the three final terms and how they are used to control the three sources of noise and optimize the performance of the dynamic noise control algorithm. The first two terms are defined in the usual way based on Manhattan distance and the bounding box of the floorplan.

B. Self Switching Current

The self switching current term, I, is defined as follows:

$$I = \sum_{\forall i \in B, \forall j \in P} curr_i * sw_i * dist_{i,j} * reg_{i,j}$$

where B is the set of all blocks, P is the set of all pins, $curr_i$ is the current requirement of block i, sw_i is the self switching factor of block i, $dist_{i,j}$ is the Euclidean distance between block i and pin j, and $reg_{i,j} = 1$ if and only if block i is in the current drawing region of pin j and zero otherwise. The current drawing region is defined to be half the distance to the next nearest pin. An explanatory graphic is provided in Figure 2. Previous work that considered the LdI/dt problem [9] did not directly consider the IR drop problem. The pin

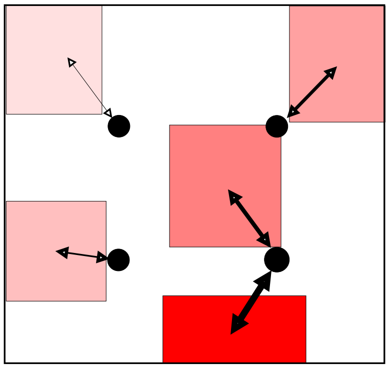

Fig. 2. Self switching current term. The black dots are power pins. high weighted blocks (darker), based on current demand and switching activity, are drawn to the power pins more strongly.

capacity force described in [9] focused on satisfying the current drawing requirements of each pin, pushing blocks away from pins that were overloaded and pulling them towards pins that were underloaded. There was no weighting of blocks that needed more current than others. The self switching term used here, I, considers both the current requirements of each block and the amount of switching that block will incur. It therefore considers both the IR drop seen by each block as well as the self induced inductance noise seen by each block. When blocks are farther away from the pins, the resistance of IR drop is increased. In a complementary fashion L of LdI/dt inductance noise is increased when blocks are farther away from pins. By minizing I, the distance between pins and blocks that have high current demand and high switching activity is minimized. Therefore, this minimizes the IR drop and LdI/dt noise seen by the chip as a whole.

C. Correlated Switching Factor

The correlated switching factor term, C, is defined as follows:

$$C = \sum_{\forall i,j \in B} \frac{curr_i * curr_j * corr_{i,j}}{dist_{i,j}}$$

where $corr_{i,j}$ is the correlated switching activity described above and in [9] between blocks i and j, and the rest of the terms are as defined in section III-B. B is the set of all blocks, $curr_i$ is the current draw of block i, and $dist_{i,j}$ is the Euclidean distance between blocks i and j. The minimization of this term maximizes the distance between blocks that have both high current and switch together frequently. If two blocks sit near each other and draw current from the same power pin, then both of these blocks switching simultaneously would exacerbate the dI/dt seen by that power pin even more than the switching of a single block. A table of the correlated switching weights for each module is shown in Figure 3.

D. Dynamic Control Queue Factor

The dynamic controller queue factor, Q, is defined as follows:

$$Q = \sum_{\forall i,j \in B} -(curr_i * curr_j * corr_{i,j} * q_{i,j})$$

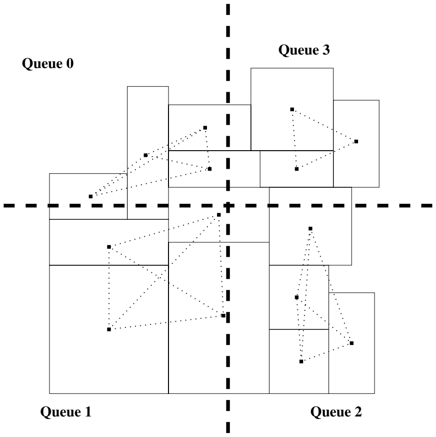

Fig. 4. Queue factor term. Each quadrant of the chip has a different queue and only modules within the same queue are given a weighted bonus based on their correlated switching activity and current requirements when evaluating the cost function. Queues are defined spatially and blocks have no movement restrictions during annealing.

where $q_{i,j} = 1$ if and only if block i and block j reside within the same dynamic noise control queue and is zero otherwise, and the rest of the terms are as defined above. B is the set of all blocks, $curr_i$ is the current draw of block i, and $corr_{i,j}$ is the correlated switching weight between blocks i and j. An explanatory graphic is provided in Figure 4. The dynamic noise controller is floorplan aware through the use of spatially organized queues. Therefore, by specifically optimizing which blocks are in which queues it is possible to optimize through physical design the operation of the dynamic noise controller. Including this type of optimization into a force-directed approach would be extremely difficult; hence this work's move to the more flexible Simulated Annealing. The queue factor takes its form from the correlated switching factor. It adds a current weighted correlated switching activity to the cost function based on whether two blocks share the same queue or not.

Now a discussion of the sign of the queue factor is provided. The initial motivation behind the form of the queue factor was that it was deemed to be more problematic if highly correlated blocks resided within the same queue. These blocks should be as far away from each other as possible, and if they reside within the same queue, this is bad because the queues are spatially designated. Experimentation proved this assumption to be wrong. Floorplans with a positive queue factor (+Q) had uniformly worse power supply noise than floorplans without any queue factor (No Q) at all. However, if the queue factor is included as a bonus (-Q) instead of as a cost function penalty the noise characteristics are improved over a noise-aware only floorplan as is demonstrated in the experimental results. This can be explained by the fact that no matter how far away highly correlated blocks are from each other they still reside on the same power distribution grid and can cause coupled inductive noise by switching simultaneously. It is also possible that blocks are next to each other, but just slightly over a queue boundary. This scenario would cause noise problems but be given lower cost with a positive queue factor. Therefore by adding a negative bonus (-Q) to the cost function when

	LSQ	RUU	BTB	L2$	IRF	L1D$	ALU0	ALU1	ALU2	ALU3	ALU4	ALU5	L1I$	Bpred	DTLB	ITLB	FALU0	FALU1	Freg
LSQ	*28*	0	20	13	20	2	10	10	10	10	10	10	11	20	0	11	10	10	12
RUU		*26*	8	4	13	2	0	0	0	0	0	0	5	8	2	5	0	0	5
BTB			*18*	7	29	17	13	13	13	13	13	13	37	100	17	37	13	13	13
L2$				*16*	14	28	12	12	12	12	12	12	21	7	26	21	4	4	7
IRF					*10*	17	7	7	7	7	7	7	23	29	17	23	8	8	24
L1D$						*7*	6	6	6	6	6	6	11	17	93	11	5	5	6
ALU0							*3*	100	100	100	100	100	15	13	6	15	66	66	4
ALU1								*3*	100	100	100	100	15	13	6	15	66	66	4
ALU2									*3*	100	100	100	15	13	6	15	66	66	4
ALU3										*3*	100	100	15	13	6	15	66	66	4
ALU4											*3*	100	15	13	6	15	66	66	4
ALU5												*3*	15	13	6	15	66	66	4
L1I$													*3*	37	12	100	11	11	5
Bpred														*3*	17	37	13	13	13
DTLB															*2*	12	5	5	6
ITLB																*1*	11	11	5
FALU0																	*1*	100	5
FALU1																		*1*	5
Freg																			*0*

Fig. 3. Correlated switching factors between the modules. Self switching weight is shown along the diagonal.

Parameters	Values
Fetch/Decode width	8-wide
Issue/Commit width	8-wide
Branch predictor	Combining: 16K entry Metatable Bimodal: 16K entries 2-Level: 14 bit BHR, 16K entry PHT
BTB	4-way, 4096 sets
L1 I- and D-Cache	16KB 4-Way 64B line
I- and D-TLB	128 Entries
L2 Cache	256KB, 8-way, Unified, 64B line
L1/L2 Latency	1 cycle / 6 cycles
Main Memory Latency	500 cycles
LSQ Size	64 entries
RUU Size	256 entries
Functional Units	8 IntAlu (only 2 can be used for IntMult) 4 FPAlu (only 2 can be used for FPMult)

TABLE I
MICROARCHITECTURE PARAMETERS

blocks that are highly correlated are in the same queue, but still separated from each other, the dynamic controller is allowed to deal with the noise problem more effectively.

IV. EXPERIMENTATION AND RESULTS

A. Experimentation Details

Experiments were carried out using the SimpleScalar tool suite [4] and hSpice circuit simulator. The microarchitecture parameters used in our experiments are listed in Table I. However, the technique described is general enough to be applied to any architecture or process technology. For physical parameter estimation a process technology of 70nm was used with a clock frequency of 5GHz. For the purposes of comparison the architecture and physical parameters were chosen to match that of Noise-Direct [9]. The experimental flow is as follows. First, correlation and self switching weights are captured using SimpleScalar without a dynamic noise controller by fast forwarding 4 billion instructions and then simulating the next 100 million. Then the various floorplanning algorithms are run. Next SimpleScalar is run again with the floorplan information inserted to collect the per cycle switching activity of each module for the 5000 cycles (out of 100 million simulated during the profiling phase) that have the worst noise characteristics. Longer sampling periods were tested, but results were almost identical to this shorter sampling period. Subsequently, decaps are inserted based on the whitespace of the floorplan. Finally hSpice is simulated using the collected switching activity to obtain the number of noise violations seen by each module.

The power distribution grid is identical to that used in [9]. The power supply is set to 1 Volt and the noise violation margin is 10%. The distribution network is a 5x5 grid with power bumps placed on every other node in an alternating pattern. Between each node of this grid resides a resistor and inductor with values proportional to the distance between grid points. Each module is connected to its nearest grid point with another pair of resistors and inductors proportional to the distance between the module and the grid point. During annealing a grid is generated for each candidate floorplan and pin locations are calculated based on the dimensions of the floorplan.[1] A graphical representation of the power distribution grid is shown in Figure 5.

B. Results Analysis

In order to compare with previous works we replicate the simulation and parameter infrastructure of Noise-Direct [9]. The comparison of the voltage swing of controller aware floorplanning and the previously published numbers for Noise-Direct is shown in Figure 6. Queue aware floorplanning has overall smaller voltage swing as compared to Noise-Direct. However, most significantly, it reduces the voltage swing to be below the 10% voltage violation threshold of $0.1V$ and therefore there are zero noise constraint violations compared to the approximately 10% noise violations per cycle reported by Noise-Direct. For the purposes of comparison there were no decaps included in the spice netlist for these voltage swing numbers. Therefore, no direct comparison between

[1] Some may argue that the number of power pins created in today's architectures is in the hundreds and so our model with 13 pins is egregiously wrong. However, the level of granularity of our simulations is designed to match the granularity of our designs. Our work resides at the floorplan and microarchitectural level, not the gate or cell level. There are billions of gates on these processors compared to the thousands of power pins. We have 13 pins and 23 modules so in fact we are being more than generous with our distribution.

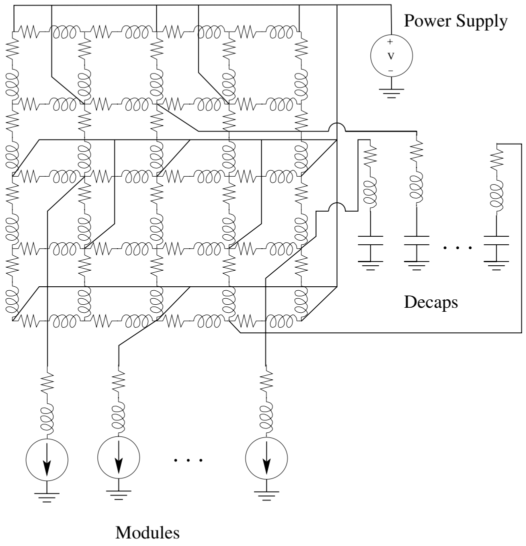

Fig. 5. The power distribution grid. Voltage bumps are spaced at every other grid point. Decoupling capacitors and modules (current sources) are connected to the gridpoint nearest them in the floorplan.

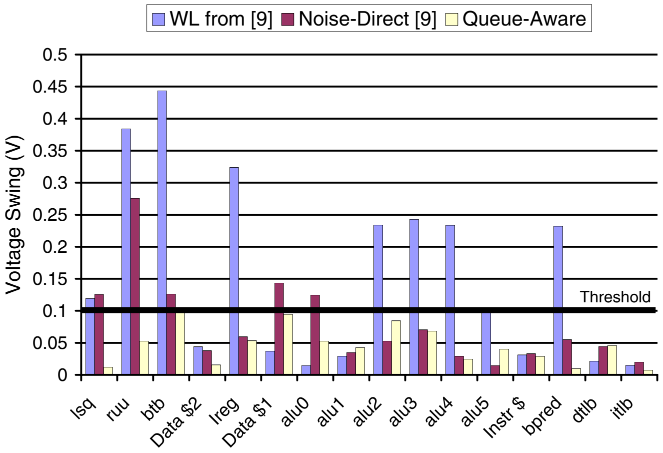

Fig. 6. Comparison with Noise-Direct [9]. Voltage violation threshold is 0.1V. The first two bars are taken directly from [9] and so this comparison does not include the use of decaps.

these values and those of the other experiments is logical. Given that Noise-Direct had no decap consideration at all there is very little that could be comparable to the remaining experiments.

Next, a comparison between the traditional area and wirelength objective (A+W), floorplanning with *positive Q factor* (+Q), floorplanning *without the queue weights* (No Q), and the new controller aware floorplanning with *negative Q factor* (-Q), all with decoupling capacitors added, is shown for voltage swing in Figure 7 and noise violations in Figure 8. A comparison between the +Q and -Q bars indicate a change in the cost function switching the Queue factor from positive to negative and shows that our initial intuition about the form of the Queue factor was incorrect. As a reminder, the Queue Factor provides a bonus (in negative form) to the cost function whenever blocks with high correlation and current demand reside within the same dynamic controller queue. The No Q bars show a floorplan that has 0 for the ϵ weight and thus is

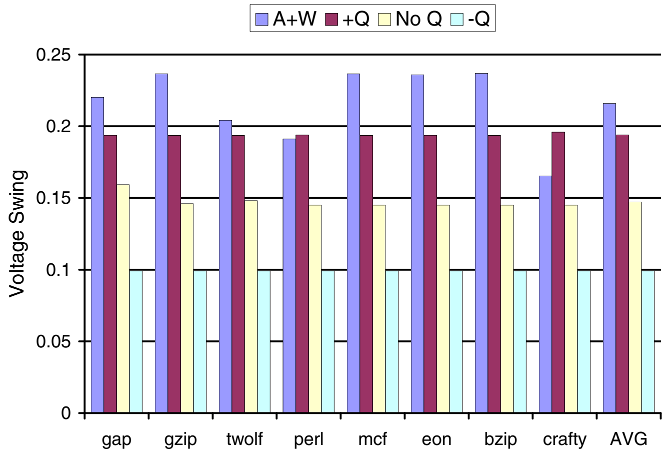

Fig. 7. Voltage swing comparison between Area and Wirelength, Positive Queue Factor (+Q), Noise-only (No Q), and Negative Queue Factor (-Q). Decoupling capacitors and the decap allocation network flow are used here.

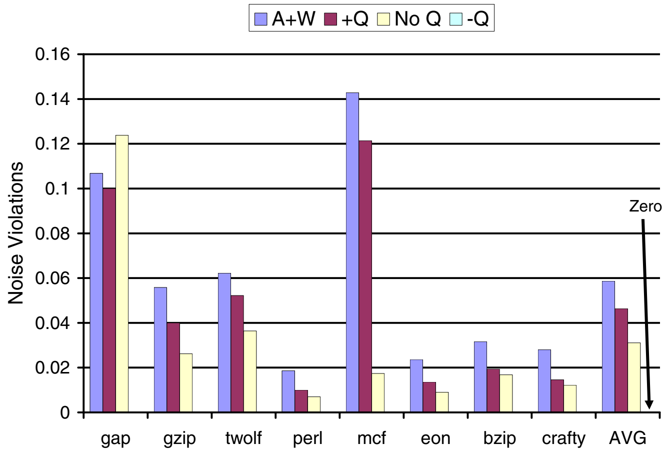

Fig. 8. Noise violation comparison between Area and Wirelength, Positive Queue Factor (+Q), Noise-only (No Q), and Negative Queue Factor (-Q). (-Q) has zero violations. As in Figure 7 this data utilizes the decap allocation flow.

most similar to the work of Noise-Direct. However, as stated previously, due to the inclusion of decoupling capacitors here no direct comparison of values between the two is logical. As shown in Figure 7, one can observe that the negative Q controller aware floorplan has better noise characterstics than those of the traditional A+W objective, the positive Q objective, and the Noise only objective. The Queue Aware floorplan has approximately 30% smaller voltage swing than the Noise only objective. This demonstrates that adding queue awareness to the floorplanner has a substantial impact for the simplicity of the change. The negative Q factor floorplan also reduces the voltage swing to be below the violation threshold and therefore there are no voltage violations for this floorplan as shown in Figure 8. Additionally, the voltage swing graph reveals that the swing is independant of the benchmark for several experiments. This is the result of the dynamic controller fully controlling the coupled voltage swing of the processor and IR drop being fully responsible for all voltage swing seen. This indicates that the negative Q aware floorplanning, for example, is the most effective method to use with the dynamic controller.

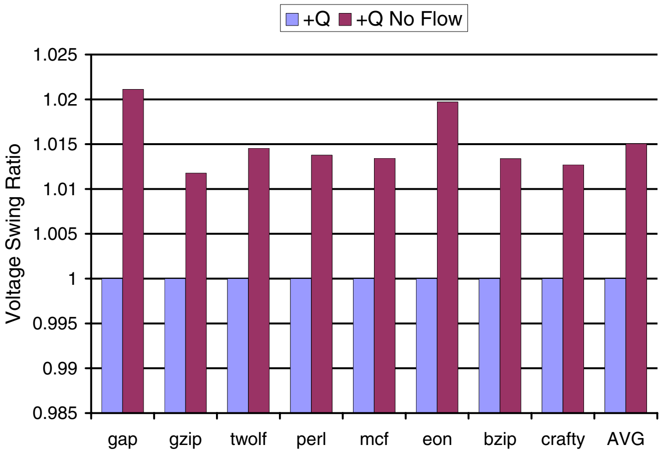

Fig. 9. Voltage ratio comparison between using the decap allocation flow (Queue-Aware) and the best according to the cost function (NoFlow). In the NoFlow case decaps are added in all the white space of the floorplan with the lowest cost.

Finally, we show that the inclusion of the network flow-based decap allocation is indeed a profitable move. A comparison of the voltage swing between the Queue-Aware floorplan and the top floorplan according to the cost function (NoFlow) is shown in Figure 9. In the NoFlow case decaps are added in all the white space of the floorplan that has the lowest cost function value. One can observe that for every benchmark the floorplan that utilizes the decap allocation flow has improved voltage swing. And in fact without the use of the decap allocation flow the floorplan does violate the noise threshold.

V. Relation to Previous Work

Prior works have attempted to address some of the issues discussed here individually. These prior techniques are useful contributions to the state of the art, however the new technique presented here is more effective and holistic than these. This is the first work to directly consider IR drop and LdI/dt inductance noise with dynamic noise controller awareness. Runtime management of power supply noise has been presented in [10], [6], [11], [12], [13], [14], [15]. Decoupling capacitor-aware floorplanning and design is presented in [16], [17], [18], [8], [19]. Power noise-aware microarchitectural floorplaninning was first studied in [9]. The authors in [9] floorplan and analyze results in conjunction with the use of a dynamic controller but do not specifically adjust the floorplan to work with the dynamic noise controller. We overcome this shortcoming with our noise-controller-aware floorplanner, where the floorplanning process is guided based on the characteristics of the underlying dynamic controller [6]. Our floorplanner is no longer constrained by the worst-case scenario because the controller is designed to respond to these emergencies. Related experiments show that we outperform [9] by 30% in terms of voltage swing.

VI. Conclusions

Processor designers will agressively battle power issues for the forseeable future. Without considering these problems at every level of the design hierarchy, advancement will be slow. As noise margins become smaller due to process technology shrinks, the worst case design will become increasingly inefficient and even ineffective. We have presented the first floorplanner to specifically work *with* a queue-based dynamic power supply noise controller to enable a design for the average noise condition. The controller prevents large simultaneous power switching events from occurring, thus guaranteeing a well behaved current demand profile. Our design flow is a holistic solution that considers the design alternatives together vertically and efficiently. Our work is also the first to include decap considerations at the microarchitectural level. The results we presented demonstrate that initial intuition about the form of controller awareness in a cost function may be wrong, and that our approach beats the state of the art significantly while being relatively simple to implement and is therefore a necessary addition to any floorplan being used with this dynamic noise controller.

References

[1] K. Aygun, M. J. Hill, K. Eilert, K. Radhakrishnan, and A. Levin, "Power delivery for high-performance microprocessors," *Intel Technology Journal*, pp. 273–283, 2005.

[2] J. C. Eble, V. K. De, D. S. Wills, and J. D. Meindl, "A Generic System Simulator (GENESYS) for ASIC Technology and Architecture Beyond 2001," in *Int'l ASIC Conference*, 1996.

[3] eCACTI, http://www.ics.uci.edu/~maheshmn/eCACTI/main.htm.

[4] T. M. Austin, "Simplescalar tool suite," http:/www.simplescalar.com.

[5] D. Brooks, V. Tiwari, and M. Martonosi, "Wattch: A framework for architectural-level power analysis and optimizations," in *Proc. IEEE Int. Conf. on Computer Architecture*, 2000.

[6] F. Mohamood, M. B. Healy, S. K. Lim, and H.-H. S. Lee, "A Floorplan-Aware Dynamic Inductive Noise Controller for Reliable Processor Design," in *Proc. Annual Int. Symp. Microarchitecture*, 2006.

[7] H. Murata, K. Fujiyoshi, S. Nakatake, and Y. Kajitani, "Rectangle packing based module placement," in *Proc. IEEE Int. Conf. on Computer-Aided Design*, 1995, pp. 472–479.

[8] E. Wong, J. Minz, and S. K. Lim, "Decoupling Capacitor Planning and Sizing for Noise and Leakage Reduction," in *Proc. IEEE Int. Conf. on Computer-Aided Design*, 2006.

[9] F. Mohamood, M. B. Healy, S. K. Lim, and H.-H. S. Lee, "Noise-Direct: A Technique for Power Supply Noise Aware Floorplanning Using Microarchitecture Profiling," in *Proc. Asia and South Pacific Design Automation Conf.*, 2007.

[10] E. Grochowski, D. Ayers, and V. Tiwari, "Microarchitectural Simulation and Control of di/dt-induced Power Supply Voltage Variation," in *Proc. ACM Design Automation Conf.*, 1998.

[11] Q. Wu, M. Pedram, and X. Wu, "Clock-gating and its application to low power design of sequential circuits," *IEEE Trans. on Circuits and Systems*, pp. 415–420, 2000.

[12] K. Hazelwood and D. Brooks, "Eliminating Voltage Emergencies via Microarchitectural Voltage Control Feedback and Dynamic Optimization," in *Proc. Int. Symp. on Low Power Electronics and Design*, 2004.

[13] M. D. Pant, P. Pant, D. S. Wills, and V. Tiwari, "Inductive Noise Reduction at the Architectural Level," 2000.

[14] ——, "An Architectural Solution for the Inductive Noise Problem Due to Clock-gating," in *Proc. Int. Symp. on Low Power Electronics and Design*, 1999.

[15] M. D. Powell and T. N. Vijaykumar, "Pipeline Muffling and A Priori Current Ramping: Architectural Techniques to Reduce High-Frequency Inductive Noise," in *Proc. Int. Symp. on Low Power Electronics and Design*, 2003.

[16] M. D. Pant, P. Pant, and D. S. Wills, "On-chip decoupling capacitor optimization using architectural level prediction," *IEEE Trans. on VLSI Systems*, vol. 10, no. 3, pp. 319–326, 2002.

[17] S. Zhao, C. Koh, and K. Roy, "Decoupling capacitance allocation and its application to power supply noise aware floorplanning," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, pp. 81–92, 2002.

[18] H. Chen, L. Huang, I. Liu, and M. Wong, "Simultaneous power supply planning and noise avoidance in floorplan design," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, pp. 578–587, 2005.

[19] Y. Chen, K. Roy, and C.-K. Koh, "Current demand balancing: A technique for minimization of current surge in high performance clock-gated microprocessors," *IEEE Trans. on VLSI Systems*, pp. 75–85, 2005.