

Place and Route Considerations for Voltage Interpolated Designs

Kevin Brownell, Ali Durlov Khan, David Brooks, Gu-Yeon Wei
School of Engineering and Applied Sciences, Harvard University, Cambridge, MA 02138
E-mail: {brownell, adkhan, dbrooks, guyeon}@eecs.harvard.edu

Abstract— Voltage interpolation is a promising post fabrication technique for combating the effects of process variations. The benefits of voltage interpolation are well understood. Its implementation in a VLSI-CAD flow has been considered through the synthesis stage. In this paper we study the implications of place and route on voltage interpolation. We evaluate multiple placement strategies, and conclude that a hybridization of forced placement and cluster boxing techniques results in minimum overhead.

I. Introduction

Process variation in CMOS circuits is a significant obstacle for circuit designers to maximize performance in nanoscale technologies. Variations at multiple spatial scales affect circuits by introducing random and systematic uncertainties in device parameters, resulting in an increase in the delays of critical paths and requiring larger timing margins to accommodate the slowest paths in the circuit. This results in lower clock frequencies or many paths and cells receiving a higher voltage than they would otherwise require to meet timing, leading to higher power consumption.

Several schemes utilizing two supply voltages have been explored, mainly with the intent to decrease power. Some schemes involve design-time static assignment of voltages [1], while others consider dynamic, post-fabrication voltage assignment [2–4]. These works require level shifters at some interfaces between circuit blocks using different voltage levels. *Voltage interpolation* (VI) is a two voltage supply technique that forgoes the use of level shifters and also enables post-fabrication voltage tuning in order to mitigate the effects of process variations [5]. An implementation of VI in [5] demonstrates ~30% frequency tuning range for a latch-based FPU, in 130nm CMOS, to combat process variation effects. Subsequently, we explored the potential benefits of VI using a standard VLSI-CAD design flow [6], by leveraging Synopsys Design Compiler [7] to cut blocks of combinational logic into substages for flip-flop based designs. We show that VI can achieve a nominal frequency target after process variations with a 10% power cost. However, that work only examines the benefits at the level of synthesized netlists, and does not consider how to implement place and route for designs with VI nor the additional overheads that this final part of the design flow can incur.

This work investigates three placement strategies for designs with VI—forced placement, cluster boxing, and a hybridization of the two. Moreover, we quantify the area, power, and delay overheads introduced by place and route. Area overhead is due to the two power switches needed for each substage of logic, while delay and power over-

heads are a result of the need to group substages adjacent to one another, to allow for the sharing of local supply nodes. These overheads are calculated in the context of two blocks from Sun Microsystem’s UltraSPARC T1 processor core [8]. Cadence SoC Encounter is used for place and route [9]. After accounting for all overheads, we show that a combination of placement restrictions and cell reassignments usually offers the best balance in overheads to implement voltage interpolation. This strategy incurs as little as 5% energy-delay overhead due to place-and-route.

The remainder of the paper is as follows: Section 2 lays out the background for this work with an overview of voltage interpolation, power gating, and the potential overheads encountered during place and route. Section 3 describes three placement strategies. Section 4 analyzes the overheads associated with each of these strategies and compares the results. Lastly, Section 5 summarizes the paper and discusses future work.

II. Background

A. Voltage Interpolation

Voltage interpolation is a technique that enables combinational logic within single flip-flop (FF) bounded stages to operate off of a fine-grain “effective” voltage, by providing two different voltage potentials on either VDD or GND. Figure 1 illustrates a basic implementation of VDD interpolation. Within a FF-bounded stage, multiple consecutive blocks of combinational logic can choose between either a high voltage (VDDH) or a low voltage (VDDL). A slow logic block, resulting from process variations, can select VDDH to speed up, while a faster logic block can save power by selecting VDDL. Note that while Figure 1 implements voltage interpolation using VDDH and VDDL, the same principle can be applied by using a fixed VDD, a low ground voltage (GNDL) and a high ground voltage (GNDH), as seen in Figure 2. Figure 3 illustrates how VDD and GND interpolation are interchangeable, and regardless of which method of interpolation is implemented, any given substage can choose a high voltage (VHIGH) or a low voltage (VLOW). Since we assume a common bulk node, ground interpolation can introduce body effect to nMOS devices, which slightly increases tuning range.

There are several important design considerations when implementing VI. Among these are the number of substages per FF stage, the relative area, delay, and power balance between substages, and the difference between VHIGH and VLOW (ΔV). Since VI avoids using level shifters between substages, there is a potential for increased static power loss when a VLOW stage drives a VHIGH stage. In the case of VDD interpolation, this is due to the weak “1” from the VLOW stage failing to com-

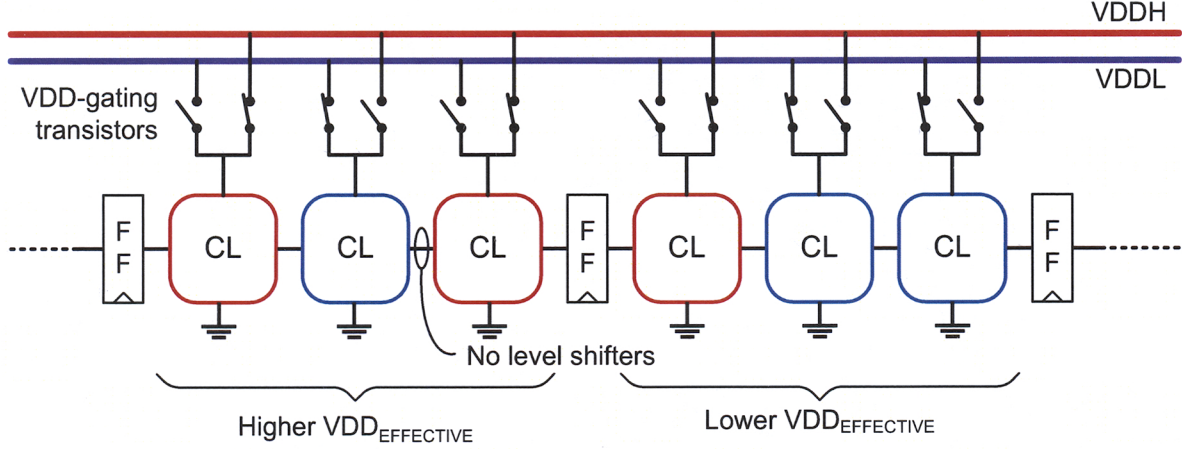


Fig. 1. Block diagram of VDD interpolation.

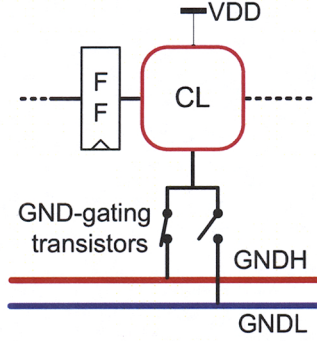


Fig. 2. Example of GND interpolation.

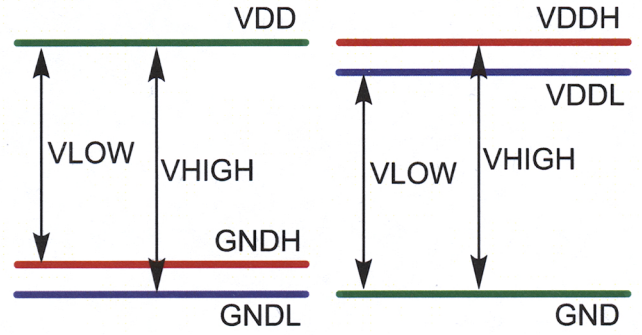


Fig. 3. Equivalence of VDD interpolation and GND interpolation.

pletely turn off the PMOS devices of the VHIGHS stage. In GND interpolation, this is due to the weak “0” failing to turn off the nMOS devices. The number of substages and the value of ΔV directly impact this static power overhead [5,6].

In order to place and route an implementation of VI, power switches must be placed in close proximity to the blocks they gate. Additionally, a scannable latch must hold the current voltage setting (VHIGHS or VLOW) of each VI substage, e.g., four latches for a three cut design. The outputs of these latches must be routed to the control inputs of the power switches. Since VI seeks to address the effects of process variations, these control bits can be set once after fabrication and do not switch frequently.

B. Power Gating

Power gating is a technique that has been extensively applied to substantially decrease the idle power consumption of inactive CMOS circuits [10]. Different circuit size granularities have been explored for power gating, although primarily coarse-grain techniques have been applied in industry. Many works have examined issues surrounding power switch sizing, and implemented schemes to accurately assess how much area is required for a given circuit block [11–13], by taking into account input patterns and timing criticality of the cells being power gated. In the context of VI, we explore the tradeoffs between circuit

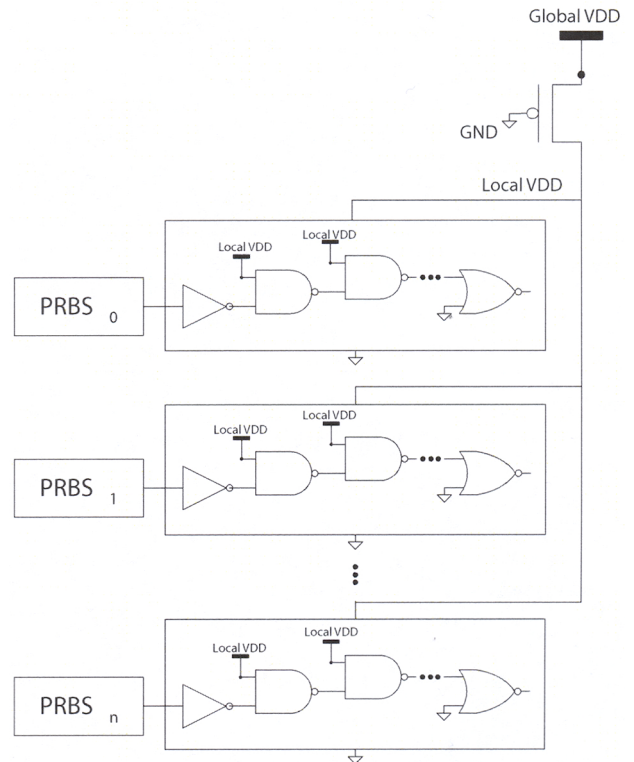


Fig. 4. Simulation setup to assess VDD-gating overheads.

size granularities for power gating.

Since VI requires two power switches for each substage, these power MUXes introduce area overhead and a non-zero impedance between the global supply net and the local supply nodes. Larger power switches can minimize this impedance, but incur higher area overhead. We quantify this overhead via circuit simulations of the setup presented in Figure 4. The framework includes multiple copies of a block of combinational logic. Each copy consists of a number of gates configured as a delay chain, and fed by a pseudo random bit sequence (PRBS). Each circuit block's PRBS is initialized with a different seed, and configured to have 20% transition density in order to model 20% activity rate of typical circuit blocks. The gates within each block are chosen from the most common gates seen along critical paths of the UltraSPARC T1 floating point adder, synthesized using a 130nm UMC standard cell library. A high threshold voltage power gating device connects the global supply to the shared local supply node shared by all of the blocks. This power gating device, although depicted by a pMOS connecting to VDD in Figure 4, can also be realized with an nMOS device connecting the local GND nodes to the global GND. The number of circuit blocks is varied from 10 to 100, and the area of the power gating device is varied from 1% to 50% of the circuit area. Each simulation is run for 5000 cycles, while recording the worst-case voltage droops.

The resulting voltage droops are used to construct Figure 5, which relates the area overhead required by circuit blocks of a certain size for a maximum voltage droop of 5%. For a given power switch size, larger current draw leads to larger average voltage droop in the local supply nodes, which degrades margins. Hence, power switches must be sized sufficiently large to minimize this droop. Interestingly, simulation results show that the relative area overhead of the power switches increases as circuit area decreases. This is because, in addition to average droop, the power switches can exacerbate local voltage noise. While large blocks can benefit from current averaging across the large number of switching logic gates, smaller blocks are more susceptible to worst-case switching scenarios. Our simulations also show that increasing power switch size (and reducing impedance) is much more effective than adding bypass capacitors to the local supply nodes, for equal area overhead.

As expected, the higher mobility of nMOS devices result in lower resistance for a given device area and introduce lower area overhead with GND interpolation. However, there may be some concern that GND interpolation could result in higher static power leakage at VLOW to VHIGH substage boundaries than pMOS devices for a given ΔV . Our simulations show that this increase in static power is negligible and the significantly lower area overhead of using nMOS gating outweighs this penalty. Hence, we choose to use GND interpolation for the remainder of this paper.

To implement these GND switches required by voltage interpolation, we propose a strategy of populating empty space in the layout with unit-sized power MUX cells, oth-

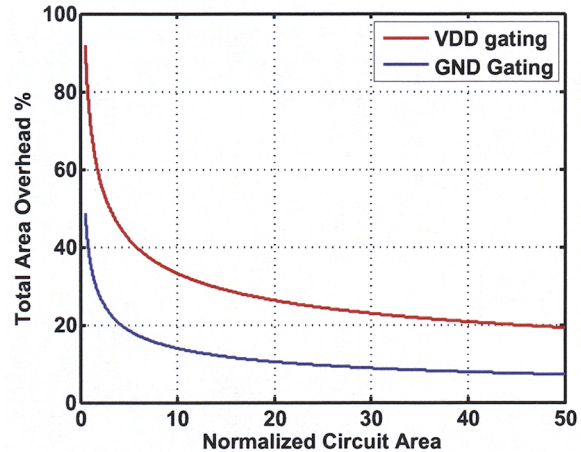


Fig. 5. Relationship between circuit size and area of power switches for a worst case VDD droop or GND blip of 5%.

erwise occupied by filler cells in a place and route design flow. This approach necessitates a small modification to the place and route flow. The global supply nets are distributed in a manner typical of an ordinary design. However, the tool must connect the local supply nets of each substage to the global supply nets through the power gating devices. The power MUX cell consists of a pair of GND-gating devices of predetermined width, which also implies a minimum quantized area cost even for the smallest circuit block sizes that may require smaller GND-gating transistors. This minimum cost further motivates us to closely consider the size of the blocks being gated, and the number of power switching cells required. Once all of the overheads have been calculated, the final comparison of different strategies only considers solutions that allows all of the power switches to fit within empty spaces.

C. Place and Route Overhead

There are two additional sources of overhead resulting from the place and route flow for a design with VI—delay and energy.

C.1 Delay Overhead

If any constraints or limitations are imposed upon the place and route flow, there is a potential for an increase in critical path delays compared to a layout without constraints. This delay overhead results from the decrease in flexibility that the placement tool has to find optimal locations for each cell and related groups of cells, leading to increases in wire length and routing congestion. We can quantify this delay overhead by comparing worst-case delays reported by Encounter after place and route.

C.2 Energy Overhead

If during place and route, cells of a particular substage cannot all be placed together, there is an increased likelihood of exacerbating the static power loss by unintentionally introducing additional VLOW to VHIGH boundaries. As depicted in Figure 6, all of the cells of each substage are

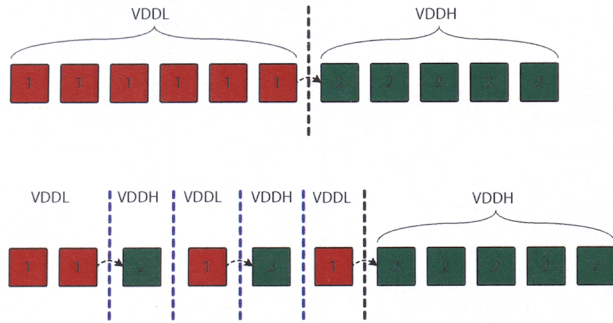


Fig. 6. Static power overhead resulting from cell reassignment.

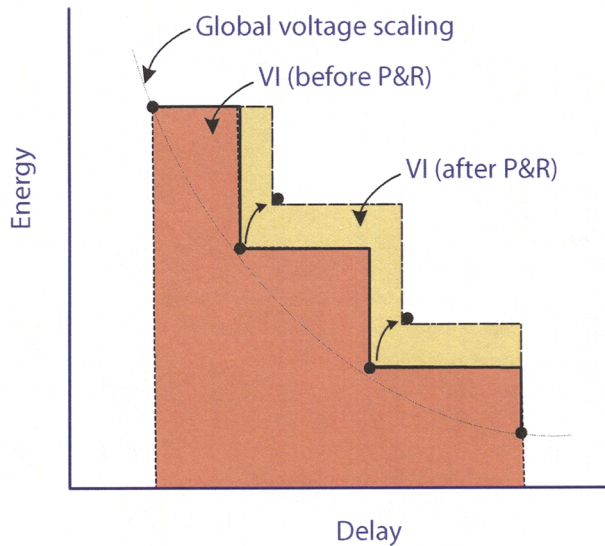


Fig. 7. Calculation of energy overhead

contiguous and there in a single boundary, before placement (top). However, depending on the place and route strategy, some cells might get separated from their original grouping (group 1) and be reassigned to the local supply node of a different group (group 2), after placement (bottom). We can evaluate this potential energy overhead by analyzing the energy vs. delay curve associated with VI. For any given VI design, there are a number of different possible tuning points, depending on the number of VI substage cuts. Figure 7 illustrates the stepwise relationship between energy and delay due to quantized tuning points. If more substages connect to VHIGH, delay is lower at the expense of higher energy. The smooth curve represents the energy vs. delay relationship if local voltage scaling were possible. Cell reassignment can cause a shift in the stair steps and increase energy for the same delay. Throughout the remainder of this paper, we represent energy overhead as the average increase in energy across the delay tuning range, calculated via detailed power analysis.

III. Place and Route Strategies

In order to realize an implementation of a design instrumented with VI, each substage needs to have its own local supply net. This motivates the need for regular well-

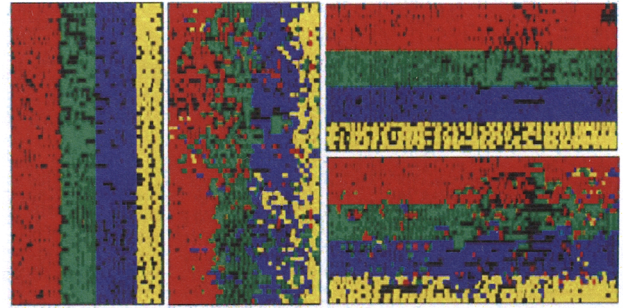


Fig. 8. Examples of forced placements with strict (left and right-upper) and relaxed (middle and right-lower) boundaries for two aspect ratios ($h/w=2$ and $h/w=0.5$).

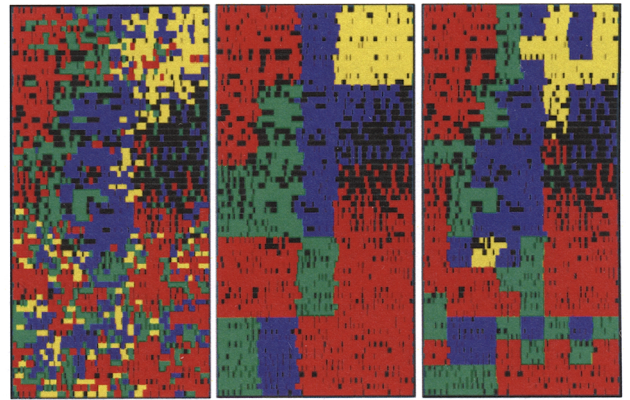


Fig. 9. Examples of cluster boxing with unconstrained (left), 5x5 (middle), and 8x15 (right) grids.

defined stage layout regions, such as “checkerboard” grids. We explore three layout placement strategies to achieve such regular layout regions using the built-in capabilities of Encounter: forced placement, cluster boxing, and a hybridization of the two.

In this work, we use two blocks from the floating point unit of the UltraSPARC T1 core, the second (FADD2) and third (FADD3) adder stages, chosen because they are likely to lie on a critical path of the core. These blocks were instrumented for one, two and three cut VI using an approach similar to the method used in [6]. To start, we use Encounter to perform unconstrained placement on the FADD2 and FADD3 blocks. This provides the baseline critical paths to which to compare the results of different strategies. In addition, these unforced placements serve as a basis for cell reassignment in the cluster boxing strategy, described below.

We assume a placement density of 70% and allow four layers of metal for signal routing. High effort timing driven placement is employed in addition to high effort timing driven routing. We choose a core density of 70% to allow fair comparisons among the layouts resulting from the various placement strategies. If a higher placement density is used, a large number of the design points fail to route and hamper our evaluation of the relative effectiveness of the different strategies. We assume that the empty space that would otherwise go to filler cells can be used to hold unit power MUX cells. Since we chose 70% core density, this leaves 30% of the area for power switches.

A. Forced-Placement Strategy

Forced placement specifies well-defined regions assigned to different substages prior to placement. Then, Encounter places cells into regions that match their substage number. Since Encounter limits only one rectangular region for a given set of instances to be placed, each substage is only able to be placed in one contiguous region. However, these regions can either be relaxed or strict. A strict region only allows cells assigned to it to be placed within the region. A relaxed region allows some flexibility at the boundaries, such that some cells that are not originally assigned to it may still be placed within the region. Figure 8 shows forced placement applied in four ways, using two different aspect ratios, each of which uses strict or relaxed regions. The advantage of the forced-placement strategy is that substage assignments originally made for the cells (during synthesis) can be maintained and avoid energy overheads. Unfortunately, this method is susceptible to delay overheads when compared to an unconstrained place-and-route of the netlist due to routing congestion. Relaxed forced placement can alleviate routing congestion, but incurs additional power mux area overhead due to a larger number of non-contiguous substage regions in the layout.

B. Cluster-Boxing Strategy

Cluster boxing is a placement strategy that starts with the baseline, unconstrained placement of cells and overlays a predetermined grid on the layout, where each rectangle within the grid is identical in size. Then, each grid rectangle is reassigned to be a substage with respect to the substage that the majority of the cells belong to. Figure 9 shows two example results of cluster boxing applied to the original placement (left) with two different grid sizes. The advantage of this method is that it does not incur a delay overhead as it retains the original unconstrained placement. On the other hand, arbitrarily changing the substage assignments of cells can result in an increased number of substage boundary crossings along any given path, introducing energy overheads.

C. Hybrid Strategy

The forced-placement and cluster-boxing strategies represent two ends of a delay-energy tradeoff continuum. The former enjoys no energy overhead, but potentially high delay overhead, while the latter incurs a high energy overhead, but no delay overhead. This motivates the consideration of a hybrid strategy that starts with a relaxed forced placement and applies cluster boxing to it. Since relaxed forced placement offers looser constraints there is a smaller impact on the critical path; additionally, there is a smaller power penalty since fewer cells are reassigned during the cluster boxing phase.

IV. Analysis

In order to evaluate the different placement strategies, we consider the tradeoffs involved in balancing among the following variables: original stage assignments for the cells, which corresponds to tunability of the system; overall critical path; power implications, particularly at the stage

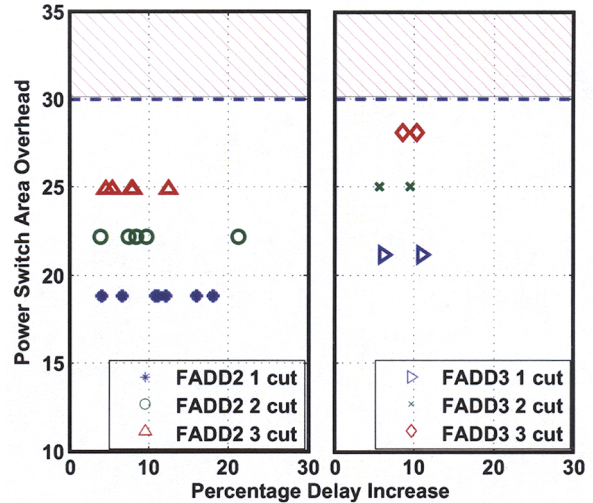


Fig. 11. Delay overhead vs. area overhead for forced placement with strict regions.

boundaries; area of power control MUXes; and the simplicity of the place-and-route schemes. The number of VI substages for each block can vary as function of the number of cuts implemented, where one cut results in two substages, and we consider up to three cuts. While more cuts typically improve circuit tunability (finer-grain effective voltages), they also increase static power overhead. Prior work in [6] concludes that three cuts is optimal, but it does not consider the additional overheads associated with place and route, which are thoroughly explored in this section.

A. Forced Placement Results

There are two types of overhead associated with the forced-placement strategy, delay and area. The energy for each layout is unchanged from the post-synthesis average energy. The delay overhead is reported as negative slack that results from place and route compared to the original delay target during synthesis. Figure 10 presents the negative timing slack (delay overhead) for a variety of aspect ratios, cut directions, and number of cuts, for FADD2 and FADD3 with strict and relaxed forced placement. The baseline results correspond to unforced placement with no cuts. While there should be a total of 10 points per column (except baseline), not all configurations routed successfully. The plot shows that in general, relaxed regions result in less negative slack than strict regions, as expected. Although a large number of designs imposing strict regions failed to route (and omitted), there are a significant number of design points that suffer less than 10% negative slack. It is important to note that since place and route is not deterministic, some results even come out better than the baseline. Moreover, imposing placement regions may provide the place and route tool with a better starting point.

Figure 11 plots the corresponding area overhead introduced by the power MUX cells vs. given delay overhead for forced placement with strict regions. Using the results of Section 2-B, the area overhead required by the two power

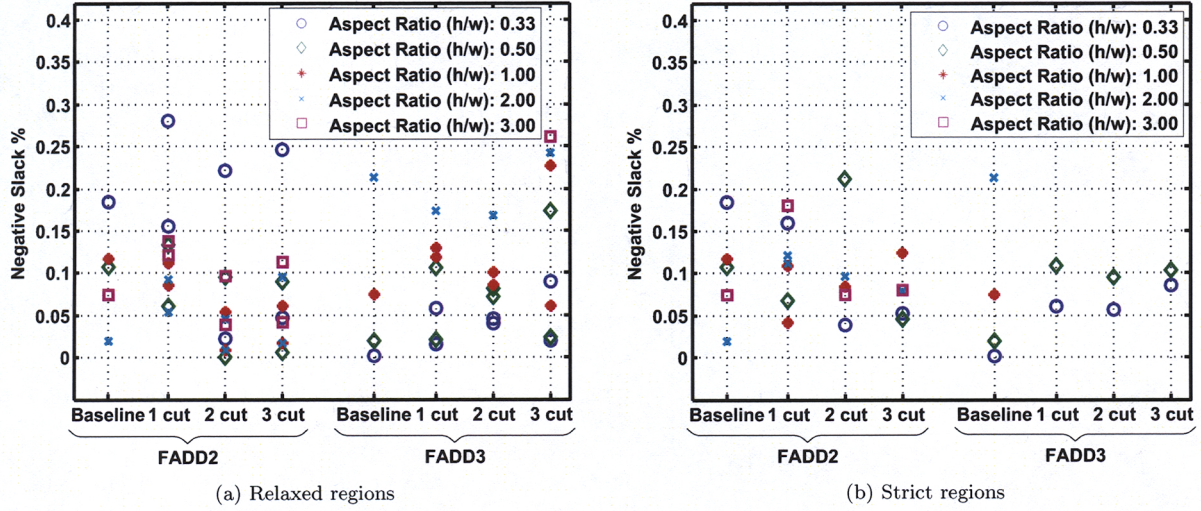


Fig. 10. Worst case delay overheads resulting from forced placement.

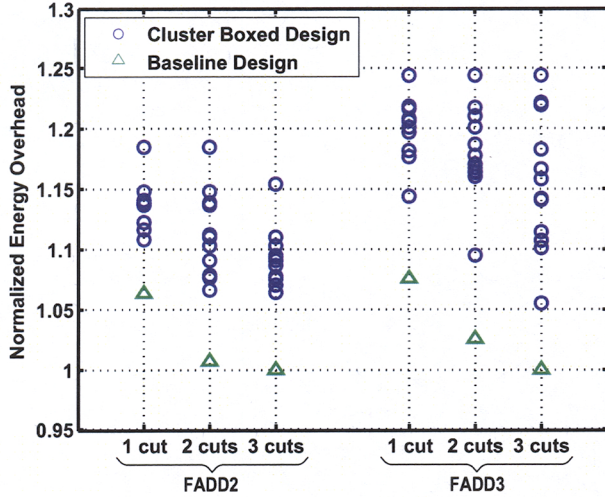


Fig. 12. Energy overhead with cluster boxing.

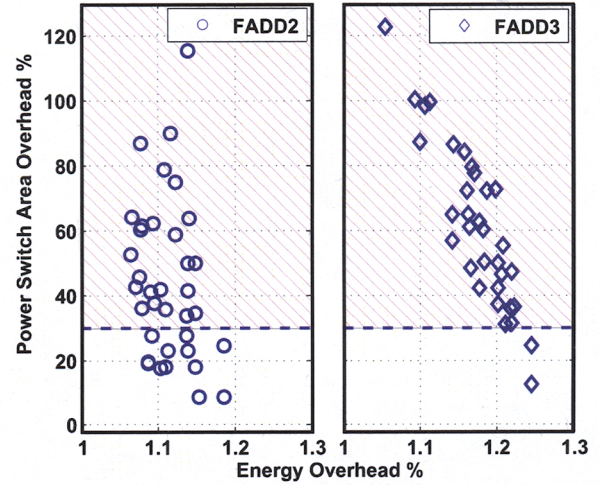


Fig. 13. Energy overhead vs. area overhead for cluster boxing.

switches can be calculated for a worst-case voltage droop of 5%. Due to the large block sizes resulting from forced placement, area overhead is modest. The fewer the number of cuts, the larger the size of the blocks, and therefore the lower the area impact. All of the design points for this strategy incur less than 30% area overhead, which may not always be the case. For example, forced placement with relaxed regions would incur much larger area overhead due to more isolated substages that each need power MUX cells, which motivates the hybrid scheme analyzed below.

B. Cluster Boxing Results

Since unforced placements are used as the basis for this strategy, there is no increase in critical path over the baseline designs. The grid size was varied from 1x5 boxes to 15x15 boxes. Figure 12 presents the energy overhead incurred by applying this strategy. The baseline design represents the average energy without considering any effects

of cell reassignment. In general, any cell reassignment results in an increase in average energy due to an increase in static power. Nevertheless, there are several cluster boxed design points which lie between 5% and 10% energy overhead.

Figure 13 presents the corresponding area overhead with respect to the set of energy overhead design points. While some points have relatively low energy and area impacts, a significant number have unacceptably high area overhead for a given energy overhead. FADD3 exhibits relatively worse energy and area overheads than FADD2. This is because the FADD3 block has unbalanced substage sizes. Since cluster boxing will reassign cells to the majority substage in a given grid block, if a substage is overrepresented, the resulting cluster boxed layout will make the previously large substages larger, and the previously small substages smaller. This creates an imbalance in the energy/delay tuning points of the design, leading to higher average energy. Note that any design points lying above 30% (or any

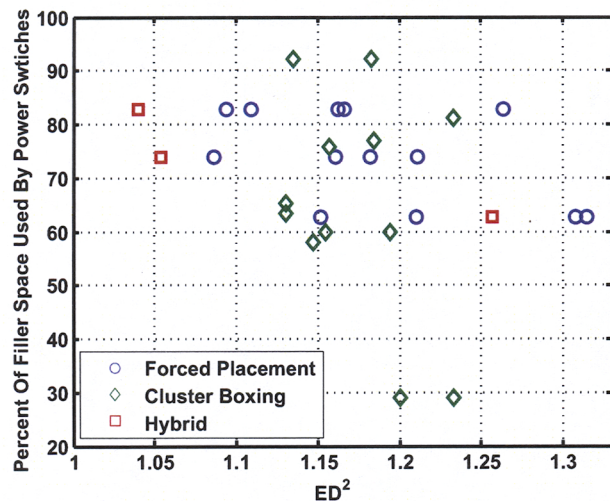


Fig. 14. Comparison of all three placement strategies for the FADD2 block.

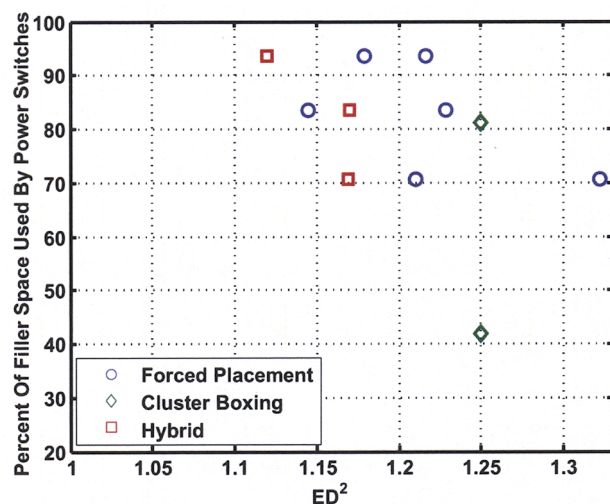


Fig. 15. Comparison of all three placement strategies for the FADD3 block.

design points where individual grid blocks had area overhead larger than the grid block size) will not be able to be implemented due to lack of room for the power switches.

C. Hybrid Results

For the hybrid strategy, we apply cluster boxing to the relaxed region forced placements that have the shortest critical paths among the one, two and three voltage interpolation cuts for each block, giving us a total of six different layouts. We note that there is a low average energy increase after cell reassignment. To evaluate this hybrid strategy, we compare it to the forced-placement and cluster-boxing strategies by plotting the area overhead vs. the energy delay squared product (ED^2). This metric is used because it is a measure of circuit performance independent of any voltage or frequency scaling techniques. Figures 14 and 15 plot all of the design points which had

low enough area overhead, such that all of their power switches could be placed within what would otherwise be used for filler cells. For both FADD2 and FADD3, the hybrid strategy with 3 cuts leads to a layout with the lowest ED^2 . In general, the hybrid solutions have lower ED^2 than most, but not all, of the forced placement and cluster boxing designs for a given block.

V. Conclusion

Voltage interpolation is an interesting approach to deal with process variations. Existing CAD tools can be used to implement VI with only slight modifications. Previous work has examined the benefits of VI [5, 6]. Our work examined the costs introduced by place and route within the context of three different placement strategies. Of them, the hybrid placement strategy explored here produces the best balance of delay and energy overhead, given a certain amount of filler space used for the requisite power switches mandated by voltage interpolation.

VI. Acknowledgements

This work is supported by NSF grant CCF-0429782, NSF grant CCF-0702344, and a gift from Intel Corp. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation or Intel.

REFERENCES

- [1] K. Usami and M. Horowitz, "Clustered voltage scaling technique for low-power design," in *International Workshop on Low Power Design*, April 1995.
- [2] K. Agarwal and K. Nowka, "Dynamic power management by combination of dual static supply voltages," in *International Symposium on Quality Electronic Design*, March 2007.
- [3] C. Tran et al., "95% leakage-reduced fpga using zigzag power-gating, dual-vth/vdd and micro-vdd-hopping," in *Asian Solid-State Circuits Conference*, November 2005.
- [4] Jui-Ming Chang and Massoud Pedram, "Energy minimization using multiple supply voltages," *IEEE Transactions on VLSI Systems*, vol. 5, no. 4, December 1997.
- [5] X. Liang, D. Brooks, and G.-Y. Wei, "A process-variation-tolerant floating-point unit with voltage interpolation and variable latency," in *International Solid-State Circuits Conference*, Feb. 2008.
- [6] K. Brownell, D. Brooks, and G.-Y. Wei, "Evaluation of voltage interpolation to address process variations," in *International Conference On Computer-Aided Design*, Nov. 2008.
- [7] Synopsys, Synopsys Design Compiler, www.synopsys.com.
- [8] Sun Microsystems Inc., "OpenSPARC T1 Chip Design," <http://www.opensparc.net/>.
- [9] Cadence, Cadence SoC Encounter, www.cadence.com.
- [10] Farzan Fallah and Massoud Pedram, "Standby and active leakage current control and minimization in cmos vlsi circuits," *IEICE Transactions*, vol. 88-C, no. 4, pp. 509–519, 2005.
- [11] James Kao, Siva Narendra, and Anantha Chandrakasan, "Mtc-mos hierarchical sizing based on mutual exclusive discharge patterns," in *DAC '98: Proceedings of the 35th annual conference on Design automation*, New York, NY, USA, 1998, pp. 495–500, ACM.
- [12] James Kao, Anantha Chandrakasan, and Dimitri Antoniadis, "Transistor sizing issues and tool for multi-threshold cmos technology," in *DAC '97: Proceedings of the 34th annual conference on Design automation*, New York, NY, USA, 1997, pp. 409–414, ACM.
- [13] Anand Ramalingam, Bin Zhang, Anirudh Devgan, and David Z. Pan, "Sleep transistor sizing using timing criticality and temporal currents," in *ASP-DAC '05: Proceedings of the 2005 conference on Asia South Pacific design automation*, New York, NY, USA, 2005, pp. 1094–1097, ACM.