Quantified Impacts of Guardband Reduction on Design Process Outcomes

Kwangok Jeong, Andrew B. Kahng and Kambiz Samadi
Electrical and Computer Engineering Dept.
University of California, San Diego
La Jolla, CA 92093
abk@cs.ucsd.edu, {kambiz,kjeong}@vlsicad.ucsd.edu

Abstract

The value of guardband reduction is a critical open issue for the semiconductor industry. For example, due to competitive pressure, foundries have started to incent the design of manufacturing-friendly ICs through reduced model guardbands when designers adopt layout restrictions. The industry also continuously weighs the economic viability of relaxing process variation limits in the technology roadmap [2]. Our work gives the first-ever quantification of the impact of modeling guardband reduction on outcomes from the synthesis, place and route (SP&R) implementation flow. We assess the impact of model guardband reduction on various metrics of design cycle time and design quality, using open-source cores and production (specifically, ARM/TSMC) 90nm and 65nm technologies and libraries. Our experimental data clearly shows the potential design quality and turnaround time benefits of model guardband reduction. For example, we typically (i.e., on average) observe 13% standard-cell area reduction and 12% routed wirelength reduction as the consequence of a 40% reduction in library model guardband; 40% is the amount of guardband reduction reported by IBM for a variation-aware timing methodology [8]. We also assess the impact of guardband reduction on design yield. Our results suggest that there is justification for the design, EDA and process communities to enable guardband reduction as an economic incentive for manufacturing-friendly design practices.

1. Introduction

In sub-90nm process technologies, there has been increased interest in design for manufacturability (DFM) techniques that address mounting variability and leakage power challenges. For example, as we review below, several recent works attempt to 'close the loop' from systematic or deterministic variability sources (litho, etch, CMP) back to design analysis (SPICE models of devices and gates, RC extraction of interconnects, etc.). However, DFM tools and methodologies that bring process awareness into design analysis and optimization will be of limited interest to design teams unless the signoff design attributes (quality-of-result, or QOR), and/or the design cycle (turnaround time, or TAT) actually improve. In particular, design teams require clear financial returns to go through the extra tool adoption, flow integration, and design effort that lead to more manufacturable tapeouts to the foundry. The challenge today is for the foundry and EDA sectors to collaboratively deliver opportunities for design-side customers to realize such financial benefits in return for deploying DFM approaches. To this end, quantified ROI (return on investment) analyses are required.

Another motivation for our work comes from the semiconductor technology roadmapping (ITRS) [2] community, which spans lithography, process integration, front-end process, interconnect, etc. technologies. In the ITRS effort, it has never been clear 'how much variability can design tolerate?' For example, the 2005 edition of the ITRS increased the lithography critical dimension (CD) 3-sigma tolerance from its historical 10% value up to 12%. While this relaxation of the ITRS CD control requirement enables continuation of the foundry process roadmap, it was obtained without analysis of the net impact on design value extractable per wafer. Future balancing between process scaling and design technology 'equivalent scaling' on the Moore's Law roadmap must be guided by more quantitative analyses.

Today, in the 65nm and early 45nm nodes, particularly for high-performance process flavors, silicon providers are likely to consider providing variant guardbands at the level of device (SPICE) model or interconnect RCX models, corresponding to different regimes of manufacturing-friendliness or "DFM score" in the tapeout. A first example might be the reduction of worstcase-bestcase (WC-BC) guardband for RC extraction, which is enabled by the deployment of new golden models for chemical-mechanical planarization (CMP), and which lead to new process-aware extraction and timing analysis (as well as process-driven dummy fill) flows. A second example might be the application of a different (narrower) SPICE model guardband for, e.g., a multi-fingered device that is laid out with optimal pitch and poly dummy layout choices.

With respect to the preceding discussion and examples, significant overheads to the silicon provider are associated with this nascent paradigm shift in the foundry-designer business model. Among these overheads: commitment to additional model-to-silicon fidelity constraints, increased process technology characterization effort, opening up of another dimension of competition with other foundries, etc. Yet, the benefits to the foundry are clear: incentive for design customers and EDA partners to 'do the right thing' for the manufacturing process, and the opportunity to offer differentiated value to customers. Clearly, a missing element for the concept of layout-specific design guardbanding to go forward is a *framework* to quantify the impact of guardband change on design QOR and TAT. Our present work seeks to fill this gap.

In this paper, we develop an experimental framework and

then experimentally quantify the impact of model guardband reduction on outcomes of the synthesis, place and route (SP&R) implementation flow. We make the following contributions.

- We study small standard-cell cores in 90nm and 65nm foundry technologies (ARM/TSMC), and separately evaluate the impacts of guardband reductions in the FEOL (Liberty timing models) and in the BEOL (RCX in golden extraction such as with STAR-RCXT).
- We assess impact of guardband reduction with respect to a number of metrics of design productivity (iterations, CPU times in synthesis, CTS and P&R phases, total design flow TAT, etc.), design closure (final timing fixes, etc.), and design quality (standard-cell area, routed wirelength, critical-path delay, etc.).
- We observe that the value of guardband reduction can be very significant. For example, we find that the 40% guardband reduction obtained by [8] with a 'iso-dense' variational timing analysis methodology leads to typical reductions of 13% in standard-cell area, 12% in routed wirelength, and 28% in SP&R turnaround time for both 90nm and 65nm designs.
- We quantify the impact of the guardband reduction on design yield. Our analysis shows up to 4% increase in the number of good dice per wafer with 20% guardband reduction. However, we notice a reduction in the number of good dice per wafer after 40% guardband reduction.

The remainder of this paper is organized as follows. Section 2 reviews several related aspects of the literature. Section 3 describes our scaling methodology for both FEOL and BEOL guardband reduction. Section 4 describes the implementation flow, tools and testcases used in our experimental investigation. In Section 5 we present experimental data that assesses impact of guardband reduction on a number of design-related metrics. Finally, Section 6 gives conclusions.

2. Related literature

We are not aware of any previous literature that quantifies impact of guardband reduction in a modern IC implementation flow, as we do. However, we note two related literatures that respectively address (1) taxonomies of variation sources and guardbanding in the modeling and analysis chain, and (2) systematic process variation-aware design analyses.

Taxonomies of Variation Sources and Guardbanding. It is well-understood that variation can arise from environmental parameters (temperature, supply voltage, etc.), manufacturing processes that lead to device and interconnect changes, and reliability effects (hot-carrier degradation, NBTI, etc.). Scheffer [15, 16] gives a taxonomy of uncertainty and variation sources, with emphasis on the back end of the line (BEOL), i.e., the interconnect stack. This is in a similar spirit to the work of Nassif [13], which reviews sources and impacts of parameter variability across inter-die and intra-die sources. While such works as these taxonomize and quantify individual variation sources, they do not make connections back to quantified impacts within the chip implementation flow.

Systematic Process Variation-Aware Design Analyses. Prediction and compensation of systematic variations has traditionally been done by the manufacturing process, with only simple guardbanded abstractions (e.g. design rules) being passed on to the designers. However, the increasing magnitude and 2-D pattern dependence of these variations, their impact on design metrics, and the inability of manufacturing equipment and process techniques to fully mitigate them, are cause for serious concern in sub-100nm technologies. If modeling and design guardbands used for timing and power signoff include compensatable systematic variations, the result is overdesign and a more difficult design closure task. With this in mind, a number of recent works have proposed systematic process variation-aware design analyses to 'close the loop' from manufacturing simulation back to the design flow.

Balasinski et al. [4] propose a methodology of manufacturability qualification for ultra-deep submicron circuits, based on optical simulation of the layout, integrated with device simulation; see also [17]. Pack et al. [14] propose to incorporate advanced models of lithographic printing effects into the design flow to improve yield and performance verification accuracy. Gupta et al. [9] observe that lithography simulation permits post-OPC (optical proximity correction) estimation of on-silicon feature sizes at different process conditions. Yang et al. [20] address post-lithography based analysis and optimization, proposing a timing analysis flow based on residual OPC errors (equivalent to lithography simulation output) for timing-critical cells and their layout neighborhoods. Cao et al. [6] propose a methodology for standard-cell characterization considering litho-induced systematic variations. In [6], the objective is to enable efficient post-litho analysis by running litho-aware characterization. Furthermore, to minimize the difference between isolated and actual placement contexts of a given standard cell, vertical dummy poly patterns are inserted at the cell boundary. Finally, it is noteworthy that Gupta and Heng [8] perform "iso-dense aware" timing analysis (based on modeling of systematic through-focus Leff variation) to achieve up to 40% reduction of the BC-WC guardband in static timing analysis. Also, Sylvester et al. [23] observe that up to 60% of BEOL guardband can be eliminated by use of the realistic BEOL variation model.

Despite such vigorous research activity in this arena, a fundamental question remains open: What is the impact of the guardband on design quality? And, what is the specific return that we can expect to be realized by the design team from availability of, e.g., iso-dense aware timing analysis [8], post-lithography based analysis and optimization [20], or any other potential path to reduced guardband? The following sections describe our efforts toward a quantified answer to this question.

3. Model and guardband scaling

3.1. Liberty model scaling

Static timing analysis operates independently of characterization, reading both a Verilog netlist and multiple timing libraries in Liberty format (.lib). The data available in the Liberty format include capacitance, thresholds/switching points, rise time, fall time, timing arcs and power arcs of each cell in the library. In corner-based design and signoff methodologies, there are best-case and worst-case design behaviors for which cells are characterized, and which are captured in respective Liberty libraries.

To capture the impact of guardband reduction on the design process, we require the ability to scale production .lib files accordingly. In our experiments, we run through a traditional timing-driven SP&R flow; hence, we scale only the input pin capacitances and timing tables, and we do not modify the power tables of the .lib files. In the Liberty format, each standard cell master has several attributes, such as pin type, loads, stimuli and lookup-table indices.

It is well-known that one can specify "PVT" scaling factors in the technology library environment, using so-called k-factors. These k-factors (so-called because they are attributes with names starting with $k_$) are multipliers that scale defined library values, allowing consideration of the effects of changes in process, voltage and temperature [1]. However, in our methodology we do not use k-factors since they cannot correctly capture guardband reduction. Instead, we apply an entry-by-entry library scaling methodology in which (1) the difference between values of a certain table entry in two libraries (e.g., worst-case and best-case) is computed, and (2) then, the amount of required guardband reduction is applied to this difference and the corresponding (e.g., best- and worst-case) table values are modified accordingly.

Figure 1 illustrates the steps required to scale timing tables within the Liberty files.

- *Goal: Entry-by-entry BC-WC guardband reduction.* Figure 1(a) shows an example of timing tables within best- and worst-case Liberty files.[1] Our goal is to apply a uniform percentage of guardband reduction to each entry-by-entry difference (i.e., the amount of guardband associated with each delay value) between best-case and worst-case delay values, which are characterized under the corresponding PVT conditions.[2] Note that we cannot simply reduce values of worst-case delays, and increase values of best-case delays, by fixed percentages; this will not result in a uniform guardband reduction.
- *Index matching step.* In a production timing library, it is common for, e.g., the input slew time indices of the best-case library to be different from the indices of the worst-case library. Hence, before we can scale entry-by-entry guardband values, we must first match up the indices of corresponding tables in the best-case and worst-case libraries. We achieve this by interpolation/extrapolation from the original index values of both tables, as illustrated by the "index-matched best-case" table in Figure 1(b).
- *Calculation of entry-by-entry guardband reduction.* After unifying the library table indices, we can compute the entry-by-entry difference (i.e., original amount of guardband) and apply the necessary guardband reduction. For example, in Figure 1(b), we see that for input slew time = 2 and capacitive load = 1, the best-case and worst-case delay values are 2 and 4, respectively. To reduce the guardband by 10%, we first find the difference between corresponding values (i.e., 4 - 2 = 2). Then, we add 5% of this difference to the best-case value, and subtract 5% of this difference from the worst-case value. The resulting guardband-reduced BC/WC values are seen in Figure 1(c). We more formally describe our index-matching and guardband reduction procedures in Figures 2 and 3.[3]
- *Scaling of pin capacitance guardband.* Note that input pin capacitance values can be considered as 1×1 tables. Hence, the same guardband reduction methods are applied to them as well.

[1]The tables shown in Figure 1 are for illustrative purposes. Neither their indices nor their entries represent realistic values.

[2]PVT condition for best (worst) case is fast (slow) transistors, high (low) supply voltage and low (high) temperature.

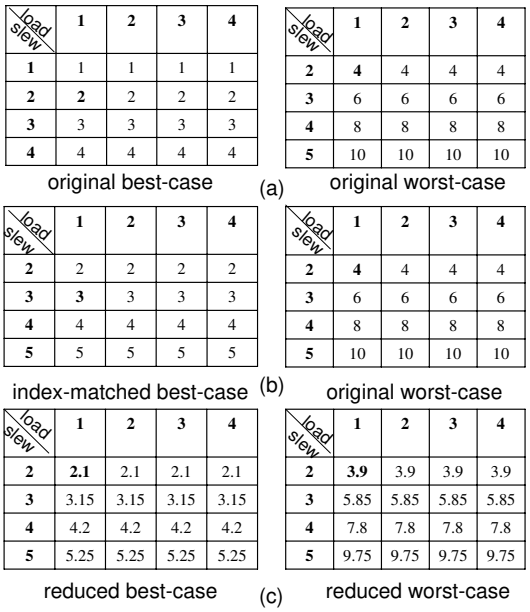

Figure 1. Illustration of steps in guardband reduction for timing tables of the Liberty (.lib) files. (a) Original best/worst-case tables. (b) New best-case table with input slew time indices matched up with those of the worst-case table. (c) 10% guardband reduction, computed on an entry-by-entry basis, across all the table entries.

Input: best/worst-case libraries.
Output: index-matched best-case library.

for all the cells in the best-case library:

find the corresponding cell in the worst-case library.

interpolate/extrapolate the new best-case timing table entries using the best/worst-case values.

copy the slew rate index of the worst-case table on to that of the best-case table.

Figure 2. Index matching procedure.

3.2. Interconnect model scaling

It is commonly accepted that interconnect has become a dominant factor in determining circuit performance. In sub-100nm processes, litho- and CMP-induced variations in conductor width, conductor thickness, and inter-layer dielectric (ILD) height within the BEOL stack can cause significant variation of interconnect parasitics.

In the corner-based design methodology, extreme values of resistance and capacitance are used to obtain worst-case

[3]In Figure 3, the factor 1/200 arises because half of the x% guardband reduction is applied to each of the best-case and worst-case values.

Input: index-matched best/worst-case libraries and $x\%$ guardband reduction.
Output: guardband reduced best/worst-case libraries.

for all the common cells in the best/worst-case libraries:
  for each entry in a best-case table ($value_{best}$):
    $value_{best} = value_{best} + \frac{x}{200}(value_{worst} - value_{best})$.
  for each entry in a worst-case table ($value_{worst}$):
    $value_{worst} = value_{worst} - \frac{x}{200}(value_{worst} - value_{best})$.

Figure 3. Guardband reduction procedure.

and best-case corners in timing analysis. For example, in best-case analysis we use the smallest capacitance value, and in worst-case analysis we use the largest capacitance value. Resistance behaves inversely to capacitance, hence minimum resistance is used in worst-case analysis and maximum resistance is used in best-case analysis. In addition to process variations, operating conditions such as temperature affect resistance and capacitance values. In 90nm copper technology, large temperature variation (e.g., from -40^{o}C to 125^{o}C) can lead to 50% increases in resistance. From Table 1, including the process and temperature effects, we see that at the worst interconnect corner, the values of capacitance and resistance are greater than those at the best interconnect corner by 17% and 13%, respectively.

We implement model guardband reduction for interconnect resistance and capacitance as follows.

- We first extract resistance and capacitance from a sample design for best and worst corners using a signoff extractor (Synopsys Star-RCXT).
- We compare the mean of the worst-corner values with that of the best-corner values.
- Finally, for a given percentage reduction in guardband, we find proper scaling factors for each corner by a method similar to that described above for Liberty scaling. The scaling equations and the relative values of interconnect capacitance and resistance for 90nm technology are summarized in the Table 1.[4]

4. Implementation flow and testcases

4.1. Timing-driven implementation flow

Figure 4 shows the traditional SP&R flow that we have scripted for "push-button" use in our experiments. The steps in Figure 4 represent the major physical design steps. At each step, we require that the design must meet the timing requirements before it can pass on to the next step. (This is standard practice, since the later in the design flow, the harder it is to fix a given timing violation.) In other words, in the event of any timing violation, our implementation flow goes back to the previous step through a return path and fixes the violation.

In the flow, we first synthesize RTL codes with worst-corner libraries. This synthesis step, when different reduced-guardband libraries are used, produces initial netlists with different total standard-cell area. We fix the utilization ratio in all testcases at the floorplan stage. We optimize timing inside the P&R tool using its embedded RCX and delay calculation engines. Since the designer's concern is generally to obtain the best performance within given environments and constraints, we concentrate on fixing setup violations at this stage of the implementation flow. Once all setup violations are cleared, it is necessary to fix hold violations using the best-case library. While attempting to fix hold violations, sometimes new setup violations are created, and iteration over the above steps is required until all violations are cleared at both the best and worst timing corners.

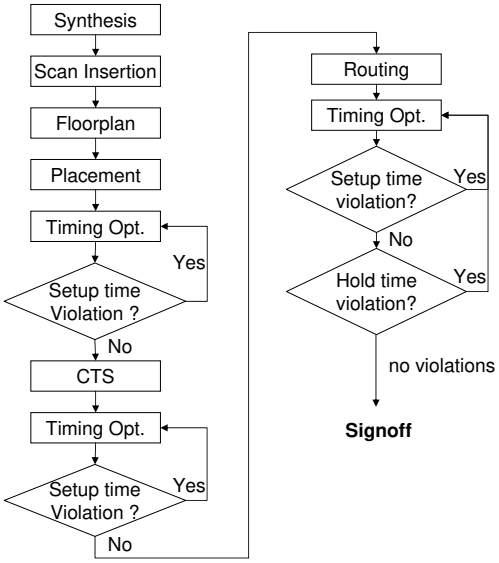

Figure 4. Implementation (Synthesis, place and route) flow.

4.2. Testcases and tools

We use three benchmark designs in our experiments. The first two are the aes and $jpeg$ cores, obtained as RTL from the open-source site $opencores.org$ [3]. The third testcase is $5Xjpeg$, which is composed of 5 copies of the $jpeg$ core. We perform our experiments using front-end libraries in TSMC 90nm and 65nm technologies. The aes core typically synthesizes to approximately 16K instances; target clock frequency is 400 MHz in 90nm and 600 MHz in 65nm. The $jpeg$ (resp. $5Xjpeg$) core typically synthesizes to approximately 64K (resp. 320K) instances; target clock frequency is 300 MHz in 90nm and 500 MHz in 65nm. We use *Cadence RTL Compiler v05.20-s009_1* to synthesize the designs. We use *Cadence SOC Encounter v5.2* to execute the P&R flow. Initial row utilizations are 40%, 60% and 60% for the aes, $jpeg$ and $5Xjpeg$ designs, respectively. Note that final row utilizations may change depending on timing optimization steps (e.g., buffering, sizing, etc.) that are executed during the P&R flow. We use *Synopsys Design Compiler v2006.06-SP3* for scan insertion and *Synopsys STAR-RCXT v2006.06-SP1* for RCX. Finally, *Synopsys PrimeTime v2005.12-SP3* is used for static timing analysis.

[4]Note that since the P&R tool (Cadence SOC Encounter) and the signoff extraction tool (Synopsys Star-RCXT) have discrepancies in their computed interconnect resistance and capacitance values, we compute separate scaling factors for each. (Analogous scaling factors are separately computed for P&R and signoff extraction in the 65nm technology.)

Table 1. R and C comparison and scaling method for 90nm interconnect.

		Best corner	Worst corner
P&R	Resistance ratio	1	1.11
	Resistance scaling factor for x% of guardband reduction	$1 + \frac{x}{200} \cdot (1.11 - 1)$	$1 - (1 - \frac{1}{1.11}) \cdot \frac{x}{200}$
	Capacitance ratio	1	1.15
	Capacitance scaling factor for x% of guardband reduction	$1 + \frac{x}{200} \cdot (1.15 - 1)$	$1 - (1 - \frac{1}{1.15}) \cdot \frac{x}{200}$
Signoff	Resistance ratio	1	1.13
	Resistance scaling factor for x% of guardband reduction	$1 + \frac{x}{200} \cdot (1.13 - 1)$	$1 - (1 - \frac{1}{1.13}) \cdot \frac{x}{200}$
	Capacitance ratio	1	1.17
	Capacitance scaling factor for x% of guardband reduction	$1 + \frac{x}{200} \cdot (1.17 - 1)$	$1 - (1 - \frac{1}{1.17}) \cdot \frac{x}{200}$

5. Experimental results and discussion

In our experiments, we run the entire implementation flow with six sets of libraries corresponding to model guardband reductions of 0%, 10%, 20%, 30%, 40% and 50%. We do this for each of three scenarios: (1) only back-end-of-line (BEOL) guardband reduction, (2) only front-end-of-line (FEOL) guardband reduction, and (3) both BEOL and FEOL guardband reduction - in order to separately observe the impact of FEOL and BEOL guardband reduction. Last, we do this for each of 90nm and 65nm technologies. As a result, each testcase is implemented with the scripted flow of Figure 4 a total of 6 x 3 x 2 = 36 separate times, 18 times in each technology.

In the following, we use "F" or "FE" as shorthand for FEOL; "B" and "BE" are shorthand for BEOL. We also give detailed tables of numerical data for the 90nm $jpeg$ core implemented with 300 MHz target frequency. Other results are presented more compactly in graphical form.

5.1. Impact on quality of results

We assess impact of guardband reduction with respect to design quality metrics of area, routed wirelength, and critical-path delay. Table 2 shows the impact of guardband reduction on the area (i.e., the sum of all standard cell areas within the design) for the 90nm $jpeg$ core implemented with 300 MHz target frequency. Table 3 shows the impact of guardband reduction on total wirelength. Also, Figures 5 and 6 show the impact of guardband reduction on area and routed wirelength for aes, $jpeg$ and $5Xjpeg$ designs using 90nm and 65nm technologies, respectively. We see that both area and wirelength metrics are "well-behaved"; they improve (decrease) as the percentage guardband reduction increases. At the 40% guardband reduction achieved by the variational timing approach from IBM [8], nearly 18% area reduction and over 21% wirelength reduction is achieved.[5] Somewhat surprisingly, guardband reduction for interconnect parasitics (BEOL) has much less impact on design quality than guardband reduction for FEOL models.

Analysis of an Example Critical Path. It is instructive to look more closely at the effect of guardband reduction on timing modeling and analysis. Table 4 shows the average cell delays in a critical path of the 90nm $jpeg$ implementation, for both best-case and worst-case corners, across different guardband reductions. We see that a 10% reduction in guardband increases (decreases) the best (worst) average

[5] In [12], Kahng and Mantik observed the existence of 'inherent noise' in IC implementation tools, and documented up to 12% change in quality of result (e.g., total post-route wirelength) due to the tools' sensitivity to such noise sources as input renaming, randomization, scaling, etc. We note this previous work because it implies a limit to cleanliness of experimental data as we trace impact of guardband reduction through the tool flow. Also, inherent tool noise may swamp any benefits of guardband reduction in certain design regimes (e.g., with respect to tightness or looseness of timing and area constraints).

Table 2. Guardband reduction vs. area for 90nm $jpeg$ design at 300 MHz.

Area	0%		10%		40%		50%	
	mm^2	%	mm^2	%	mm^2	%	mm^2	%
F	0.367	100	0.356	97.0	0.339	92.3	0.331	90.3
B	0.367	100	0.367	100.1	0.357	97.5	0.355	96.7
F+B	0.367	100	0.355	96.9	0.339	92.4	0.331	90.3

Table 3. Guardband reduction vs. total wirelength for 90nm $jpeg$ design at 300 MHz.

WL	0%		10%		40%		50%	
	mm	%	mm	%	mm	%	mm	%
F	1609.2	100	1608.6	99.9	1544.3	96.0	1512.2	94.0
B	1609.2	100	1617.2	100.5	1586.6	98.6	1590.1	98.8
F+B	1609.2	100	1593.3	99.0	1539.0	95.6	1514.8	94.1

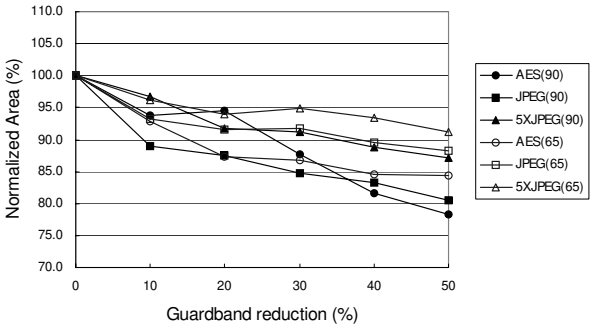

Figure 5. Guardband reduction versus area.

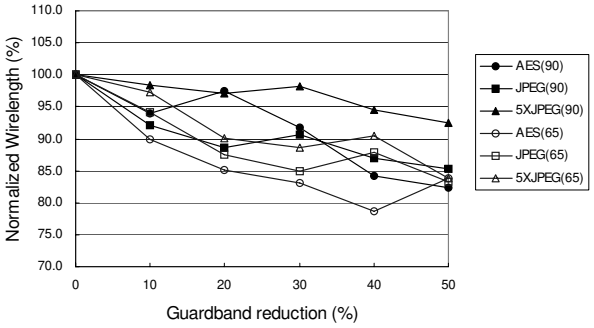

Figure 6. Guardband reduction versus total wirelength.

stage delay by only 4ps (3% of the average stage delay). Also, the delay differences across different guardband reductions in the BEOL are very small compared to the differences in the FEOL. Possibly, the impact of BEOL guardband reduction (despite being expected and evident from our data) will not always be visible due to inherent noise in EDA implementation tools [12]. The results of Table 4 are in alignment with the trends we observed for area and wirelength versus guardband reduction above.

Table 4. Critical path delay variations across different guardband reductions.

Cases	GB reduction	Timing corner	Total path delay (ns)	Average stage delay (ns)
	0%	WC	3.520	0.147
		BC	1.435	0.060
F	10%	WC	3.406	0.142
		BC	1.525	0.064
	40%	WC	3.069	0.128
		BC	1.813	0.076
	50%	WC	2.960	0.123
		BC	1.910	0.080
B	10%	WC	3.515	0.146
		BC	1.437	0.060
	40%	WC	3.502	0.146
		BC	1.443	0.060
	50%	WC	3.497	0.146
		BC	1.445	0.060
F+B	10%	WC	3.410	0.142
		BC	1.523	0.063
	40%	WC	3.085	0.129
		BC	1.804	0.075
	50%	WC	2.979	0.124
		BC	1.899	0.079

5.2. Impact on design cycle time

Table 5 shows the substantial impact of guardband reduction on total SP&R flow runtime for the 90nm $jpeg$ testcase. Also, Figure 7 shows the impact of guardband reduction on total SP&R flow runtime for aes, $jpeg$ and $5Xjpeg$ designs using 90nm and 65nm technologies. The data show up to 41% reduction in SP&R flow runtime with a 40% guardband reduction. In real-world design contexts, such a substantial reduction in SP&R runtime can be enabling: at a minimum, it reduces tapeout schedule risk, and permits additional optimization iterations and design space explorations. A substantial reduction in SP&R flow runtime can also reduce time to market for an IC product.

Another very clear benefit from guardband reduction can be seen from analysis of violations in signoff analysis. Recall that if there are violations at the signoff stage, then it is necessary to go back to the P&R stage and fix them. The number of design iterations needed to fix violations is reflected by a variety of 'figure of merit' parameters that are often tracked by designers, e.g., total number of violations, worst negative slack (WNS), and total negative slack (TNS). These three metrics represent different views of the design timing characteristics:

- The total number of violations represents how many violating points the designer needs to worry about.
- WNS represents the largest timing violation.
- TNS indicates how difficult fixing all the current violations in a design can be.

From these numbers, we can estimate the difficulty of meeting timing constraints, and how many iterations will be required. For example, from the total number of violations and TNS of hold time analysis, the designer can estimate how many buffers are needed to fix the violations, and indirectly estimate how much the standard-cell area will increase as a result. Or, the designer can use the WNS value to see how close a design is to becoming feasible with respect to timing constraints.

Table 6 shows various figures of merit for the 90nm $jpeg$ post-P&R result obtained with a 0% guardband reduction - and then evaluated using other (10%, 40%, 50%) guardband reductions. The table gives number of violations, worst negative slack, and total negative slack, with respect to both setup and hold constraints using signoff flow. Here, we can see very substantial benefits from guardband reduction. E.g., with a 40% guardband reduction, the vast majority of timing violations are erased, and the WNS and TNS metrics are also reduced substantially (by up to 100%). This will clearly improve timing convergence by reducing design iterations.

Table 5. Guardband reduction vs. total SP&R flow runtime for 90nm $jpeg$ design at 300 MHz.

Runtime	0%		10%		40%		50%	
	sec	%	sec	%	sec	%	sec	%
F	7129	100	5653	79.3	4068	57.1	4061	57.0
B	7208	100	7327	101.7	7507	104.1	5755	79.8
F+B	6950	100	5729	82.4	4366	62.8	4061	58.4

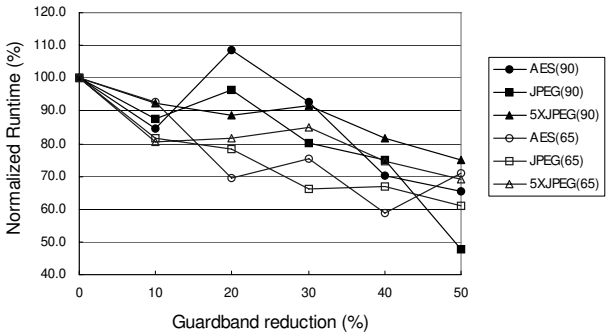

Figure 7. Guardband reduction versus total SP&R flow runtime.

Table 6. Guardband reduction versus number of violations, worst negative slack (WNS) and total negative slack (TNS).

			Guardband reduction			
			0%	10%	40%	50%
F	Setup	# of viols	235	3	0	0
		WNS (ns)	-0.126	-0.016	0	0
		TNS (ns)	-9.95	-0.03	0	0
	Hold	# of viols	4414	675	526	287
		WNS (ns)	-0.116	-0.045	-0.030	-0.028
		TNS (ns)	-259.68	-15.19	-4.20	-1.06
B	Setup	# of viols	235	231	203	198
		WNS (ns)	-0.126	-0.121	-0.11	-0.10
		TNS (ns)	-9.95	-8.97	-6.29	-5.43
	Hold	# of viols	4414	4410	4404	4400
		WNS (ns)	-0.116	-0.116	-0.116	-0.116
		TNS (ns)	-259.68	-259.39	-259.59	-258.34
F+B	Setup	# of viols	235	3	0	0
		WNS (ns)	-0.13	-0.011	0	0
		TNS (ns)	-9.95	-0.02	0	0
	Hold	# of viols	4414	676	524	298
		WNS (ns)	-0.116	-0.045	-0.030	-0.034
		TNS (ns)	-259.68	-15.24	-4.30	-1.11

5.3. Impact of guardband reduction on design yield

Guardbanding exists in today's design methodologies to help guarantee high yield in spite of process variability. In this subsection, we quantify the impact of guardband reduction on design yield. We believe that such quantification will be an essential part of manufacturing-aware design methodologies in the future.

Modeling. Overall yield may be modeled as the product of *random* yield, which depends on die area, and *systematic* yield, which is independent from die area.

$$Y = Y_r \cdot Y_s \tag{1}$$

A variety of models exist for the spatial distribution of random electrical faults across a wafer, and random yield Y_r. Commonly, random defects are characterized by defect density parameter d, and clustering parameter α. The average number of defects on a chip of area A is Ad. The number of defects x in a random chip is an integer-valued random variable, and the observed phenomenon of defect clustering is effectively modeled by assuming a negative binomial probability density function for x [5]:

$$\begin{aligned} p(x) &= Prob(\text{number of defects on chip} = x) \\ &= \frac{\Gamma(\alpha + x)}{x!\Gamma(\alpha)} \frac{(Ad/\alpha)^x}{(1 + Ad/\alpha)^{\alpha+x}} \end{aligned} \tag{2}$$

where $\Gamma(x)$ is the Gamma function. The yield with respect to random defects is obtained as the probability $p(0)$ of having no defect on a chip. Substituting $x = 0$ in (2):

$$Y_r = (1 + Ad/\alpha)^{-\alpha} \tag{3}$$

If we use $\alpha = \infty$, which corresponds to the case of unclustered defects, Eq. (3) gives a *Poisson* density function with mean Ad, and the yield with respect to random defects is pessimistically estimated as:

$$Y_r = e^{-Ad} \tag{4}$$

Yield with respect to systematic variation, Y_s, can be estimated by considering a normal distribution with best-case and worst-case corners being set at -3σ and 3σ, respectively. The 3σ window can be taken to define the original guardband (i.e., 0% guardband reduction, with range 6σ).[6] Then, assuming no change in manufacturing variability, a K% design guardband reduction would result in a reduced range of (6σ)(100-K)/100. To calculate the systematic yield impact of design guardband reduction with no change of manufacturing variability, we may use the error function (erf, i.e., cumulative distribution of the normal distribution) for the appropriate range. For example, $Y_s(RGB\%)$ with respect to 0% guardband reduction can be computed as

$$Y_s(0) = \frac{1}{2}(1 + erf(\frac{3}{\sqrt{2}})) - \frac{1}{2}(1 + erf(\frac{-3}{\sqrt{2}})) = 0.9973$$

Yield Impact Calculation. To assess the impact of guardband reduction on design yield, we track the change in the number good dice per wafer as we reduce the design guardband. There are two main scenarios.

[6] We understand that these assumptions are appropriate to current practice. Our discussion can be easily modified to use a different kσ window.

1. We are able to control or improve the process so as to reduce the amount of guardbanding. This scenario corresponds to performing "iso-dense" timing analysis [8].
2. We simply apply a reduced guardband during the design process, even though the actual variability of the manufacturing process remains the same. This scenario corresponds to the $Y_s(RGB\%)$ calculation above.

Scenario (1) implies that Y_s remains at 0.9973, while overall yield increases because we benefit from decreased random defect yield loss due to decreased die area. Scenario (2), which is the focus of our discussion henceforth, changes $Y_s(RGB\%)$ as described above and is more pessimistic because no process improvement is assumed: the design guardband reduction increases random defect yield Y_r due to reduced die area, but this trades off against decreased Y_s.[7] To calculate the number of good dice per wafer, we first compute the gross number of dice per wafer as described in [19]:

$$N_{gross} = \pi(\frac{r^2}{A} - \frac{2r}{\sqrt{2A}}) \tag{5}$$

where A represents the die area which is fabricated on a wafer with radius r. In the above equation the second term accounts for wasted area around the edges of a circular wafer. Using Equations 1 and 5, then the number of good dice per wafer is:

$$N_{good} = Y \cdot N_{gross} \tag{6}$$

Figure 8 shows the change in the number of good dice per wafer over the guardband reduction for different defect clustering. Figure 9 shows the change in number of good dice per wafer over the space of all the percent guardband and percent area reductions. These plots reflect a typical SOC in 90nm and 65nm, with die area = 1 cm^2 and only 50% of the die being logic that is affected by the guardband reduction (i.e., half the die is memory, analog, etc.). Area reduction due to the guardband reduction is taken from average area reduction over all our test cases at each guardband. Our use of an average area reduction is justified since all testcases across 90nm and 65nm show results that are monotone in guardband reduction value, and that have standard deviation (for any given guardband reduction value) of less than 5% (see Figure 5).

From these plots, we can see that the number of good dice per wafer is maximized at around 20% of guardband reduction. This trend will not be changed by the clustering of defects. Figure 9 shows level curves of the number of good dices per wafer, plotted against guardband reduction (y-axis) and area reduction (x-axis). The dashed trace shows (area reduction, guardband reduction) points that we have realized experimentally. We see that the number of good dice increases by up to 4%, then starts to decrease,

[7] There is a third scenario, where the design floorplan is fixed so that standard-cell area reduction (due to reduced design guardbanding) does not result in any die area reduction. In this third scenario, wirelength reduction in the standard-cell blocks will result in lower metal density, which will reduce particle defect yield loss (since critical area is a function of wire density [11]). Hence, even when there is no change in die area with guardband reduction (e.g., with fixed-floorplan or pad-limited designs), we can expect a certain amount of Y_r improvement which increases the number of good dice per wafer. However, we do not currently have the tool infrastructure or foundry critical-area analysis decks needed to study this scenario.

until the onset of yield degradation beyond 40% reduction in guardband.[8]

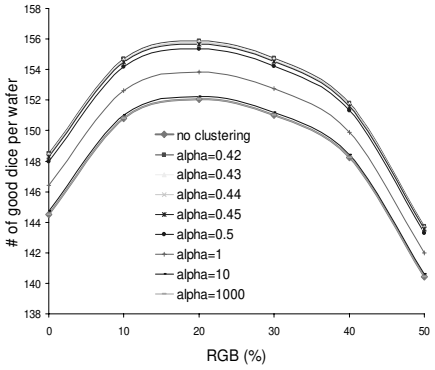

Figure 8. Change in number of good dice per wafer, versus guardband reduction (%) and defect clustering.

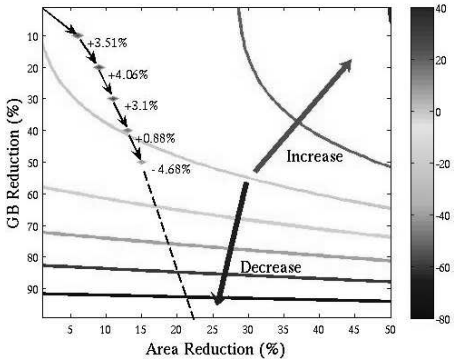

Figure 9. Change (%) in number of good dice per wafer, versus guardband reduction (%) and area reduction (%).

6. Conclusions

In this paper, we have established an experimental framework and then experimentally quantify the impact of model guardband reduction on outcomes of the synthesis, place and route (SP&R) implementation flow. We assess the impact of model guardband reduction on various metrics of design cycle time and design quality, using open-source cores and production 90nm and 65nm technologies and libraries. We observe a typical (i.e., on average) outcome of 13% and 12% reductions in standard-cell area and total routed wirelength metrics from a 40% reduction in library model guardband. We also observe up to 100% reduction in number of timing violations for a netlist that is synthesized with original library and extraction guardbands; this improvement can prove to be a significant factor in timing closure and design cycle turnaround time. Last, we quantify the impact of the guardband reduction on design yield. Our (Scenario 2) analysis shows up to 4% increase in the number of good dice per wafer with 20% guardband reduction. Interestingly, this increase in the number of good dice comes without any assumption of improved manufacturing capability (i.e., variability reduction). In addition, statistical analysis and optimization methodologies may not provide, by themselves, sufficient improvement of circuit metrics (e.g., [21] cites a 2% power reduction from statistical optimization; see also [22]). Therefore, our results suggest that there is justification for the design, EDA and process communities to enable guardband reduction as an economic incentive for manufacturing-friendly design practices.[9]

[8] 4% increase in the number of good dice is significant. For example, if a design needs 50K wafers to produce 30M good units, and the cost per wafer is $3K, the 4% represents a reduction of 2K wafers for the same number of good units, and the cost saving is $6M.

[9] As we have noted above: Although there exist clear decreasing trends in area and wirelength with respect to guardband reduction, due to the noise in the commercial tools, small guardband reductions (e.g., by $<$ 10%) may not always change flow outcomes as noticeably or consistently.

References

[1] *Liberty User Guide*, Vol. 1, Version 2006.06.

[2] *International Technology Roadmap for Semiconductors*, http://public.itrs.net/.

[3] *OPENCORES.ORG*, http://www.opencores.org/.

[4] A. P. Balasinski, L. Karklin and V. Axelrad, "Impact of Subwavelength CD Tolerance on Device Performance", *Proc. SPIE*, 361 (2002), pp. 361–368.

[5] M. L. Bushnell and V. D. Agrawal, *Essentials of Electronic Testing: for Digital, Memory and Mixed-Signal VLSI Circuits*, Kluwer Academic Publishers, 2000.

[6] K. Cao, S. Dobre and J. Hu, "Standard Cell Characterization Considering Lithography Induced Variations", *Proc. DAC*, 2006, pp. 801–804.

[7] M. Garg, A. Kumar, J. van Wingerden and L. Le Cam, "Litho-Driven Layouts for Reducing Performance Variability", *Proc. ISCAS*, 2005, pp. 3551-3554.

[8] P. Gupta and F.-L. Heng, "Toward a Systematic-Variation Aware Timing Methodology", *Proc. DAC*, 2004, pp. 321–326.

[9] P. Gupta, A. B. Kahng, S. Nakagawa, S. Shah and P. Sharma, "Lithography Simulation-Based Full-Chip Design Analyses", *Proc. SPIE*, vol. 6156 (2006), pp. 61560T1–61560T8.

[10] P. Gupta, A. B. Kahng, Y. Kim, S. Shah and D. Sylvester, "Modeling of Non-Uniform Device Geometries for Post-Lithography Circuit Analysis", *Proc. SPIE*, Vol. 6156 (2006), pp. 61560U1–61560U10.

[11] H. T. Heineken and W. Maly, "Interconnect Yield Model for Manufacturability Prediction in Synthesis of Standard Cell Based Designs", *Proc. ICCAD*, 1996, pp. 368–373.

[12] A. B. Kahng and S. Mantik, "Measurement of Inherent Noise in EDA Tools", *Proc. ISQED*, 2002, pp. 206-211.

[13] S. Nassif, "Modeling and Forecasting of Manufacturing Variations", *Proc. IWSM*, 2000, pp. 2-10.

[14] R. C. Pack, V. Axelrad, A. Shibkov, V. V. Boksha, J. A. Huckabay, R. Salik, W. Staud, R. Wang and W. D. Grobman, "Physical and Timing Verification of Subwavelength-Scale Designs: I. Lithography Impact on MOSFETs", *Proc. SPIE*, 51 (2003), pp. 51–62.

[15] L. Scheffer, "Why Are Timing Estimates So Uncertain? What Could We Do About This?", *Workshop Notes*, TAU-2002. Available at http://www.lscheffer.com/Uncertain.pdf .

[16] L. Scheffer, "An Overview of On-Chip Interconnect Variation", *Proc. SLIP*, 2006, pp. 27-28.

[17] A. Shibkov and V. Axelrad, "Integrated Simulation Flow for Self-Consistent Manufacturability and Circuit Performance Evaluation", *Proc. SISPAD*, 2005, pp. 127-130.

[18] D. Tsien, C.K. Wang, Y. Ran, P. Hurat and N. Verghese, "Context-Specific Leakage and Delay Analysis of a 65nm Standard Cell Library for Lithography-Induced Variability", *Proc. SPIE*, Vol. 6521, 2007, pp. 65210F

[19] N. H. E. Weste and D. Harris, *CMOS VLSI Design: A Circuits and Systems Perspective*, Addison Wesley, 2005.

[20] J. Yang, L. Capodieci and D. Sylvester, "Advanced Timing Analysis Based on Post-OPC Extraction of Critical Dimensions", *Proc. DAC*, 2005, pp. 359–364.

[21] S. M. Burns, M. Ketkar, N. Menezes, K. A. Bowman, J. W. Tschanz and V. De, "Comparative Analysis of Conventional and Statistical Design Techniques", *Proc. DAC*, 2007, pp. 238–243.

[22] F. N. Najm, "On the Need for Statistical Timing Analysis", *Proc. DAC*, 2005, pp. 764–765.

[23] D. Sylvester, O. S. Nakagawa and C. Hu, "Modeling the Impact of Back-End Process Variation on Circuit Performance", *Proc. VLSITSA*, 1999, pp. 58–61.