Defocus-Aware Leakage Estimation and Control

Andrew B. Kahng
CSE and ECE Departments
UC San Diego
abk@cs.ucsd.edu

Swamy Muddu
ECE Department
UC San Diego
smuddu@ucsd.edu

Puneet Sharma
ECE Department
UC San Diego
sharma@ucsd.edu

ABSTRACT

Leakage power is one of the most critical issues for ultra-deep submicron technology. Subthreshold leakage depends exponentially on linewidth, and consequently variation in linewidth translates to a large leakage variation. A significant fraction of variation in linewidth occurs due to systematic variations involving focus and pitch. In this paper we propose a new leakage estimation methodology that accounts for focus-dependent variation in linewidth. The ideas presented in this paper significantly improve leakage estimation and can be used in existing leakage reduction techniques to improve their efficacy. We modify the previously proposed gate length biasing technique of [9] to consider systematic variations in linewidth and further reduce leakage power. Our method reduces the leakage spread between worst and best process corners by up to 62%. Defocus awareness improves leakage reduction from gate length biasing by up to 7%.

Categories and Subject Descriptors: B.8.2 Performance Analysis and Design Aids

General Terms: Design, Performance

Keywords: Lithography, Leakage, ACLV, Yield

1. INTRODUCTION

Leakage power is one of the most critical design concerns in sub-100*nm* technology nodes. Decreased supply voltage (and consequently threshold voltage) combined with aggressive clock gating reduces dynamic power, but increases leakage power causing the leakage share of total power to increase. Leakage is composed of three major components: (1) subthreshold leakage, (2) gate leakage, and (3) reverse biased drain substrate and source-substrate junction band-to-band-tunneling leakage [2]. At room temperature, subthreshold leakage is the dominant contributor to total leakage at the 90*nm* technology node and will continue to contribute significantly at least through the 65*nm* node (25*nm* L_{eff}) [2].

Runtime leakage reduction techniques explore design tradeoffs within performance constraints by identifying candidate devices for optimization using leakage power estimates. Inaccurate estimation of leakage power can degrade the results of leakage reduction, and hence accurate estimation of leakage is important. Leakage power increases exponentially with decrease in linewidth. For example, with 90*nm* BPTM devices models [1, 4], we observe over a $5\times$ and $2.5\times$ increase in leakage for PMOS and NMOS devices respectively when the drawn gate length reduces from 100*nm* to 90*nm*.

Traditional leakage optimization techniques are either oblivious to across-chip linewidth variation (ACLV) or model it as a random variable resulting in very pessimistic guardbanding and consequently over-design. In reality, ACLV is partly systematic and can be modeled. A significant fraction of ACLV occurs due to systematic interactions between focus and pitch. Line pitch is dependent on the layout topology and focus depends primarily on optical column parameters in the wafer stepper. Pitch-dependent linewidth variation is accounted for at nominal focus conditions during optical proximity correction (OPC). At focus conditions other than nominal, however, linewidth variation can be large, i.e., variation in focus leads to pitch-dependent linewidth variation. Sources of focus variation, also referred to as *defocus*, include wafer topography variation, lens aberrations, tilt of wafer stage during processing, etc. [5]. Among different sources of defocus, topography and lens aberrations are systematic and can be modeled.

In this paper, we first assess the improvements in leakage estimation that can be obtained by modeling of systematic variations in linewidth. Toward this end we simulate pitches by test patterns of parallel gate polys with varying spacing and analyze their linewidth variation after performing optical proximity correction (OPC) and lithography simulation at different defocus levels. To predict the leakage of a design, we analyze its layout to compute the pitches of all devices in it and take the defocus map of the design as an input. We use the defocus and pitch information along with our observations from the lithography simulation of test patterns to predict linewidths of all devices in the design. The predicted linewidths are then used to determine device leakages and the circuit leakage.

We also add defocus awareness to enhance a recently proposed leakage reduction technique, gate length biasing [9] Gate length biasing selectively increases the gate length of devices (which has an effect of making the device slower but less leaky) in cells that are not on timing-critical paths. Defocus awareness inclines gate length biasing to bias cells, devices of which are likely to print with a smaller gate length and be extremely leaky. With our modifications, gate length biasing achieves improved leakage reduction. In summary, the main contributions of our work are:

1. Modeling of pitch- and defocus- dependent systematic components of linewidth variation to better predict leakage.
2. Defocus aware gate length biasing that models systematic linewidth variation for improved leakage reduction.

The remainder of this paper is organized as follows. In Section 2 we present background on ACLV, summarize the role of

ISLPED'05, August 8–10, 2005, San Diego, California, USA

topography in defocus, and discuss topography simulation. Section 3 describes our defocus-aware leakage estimation methodology and gives experimental results. Defocus-aware gate length biasing methodology and results are presented in Section 4. Section 5 concludes with a brief description of ongoing research.

2. BACKGROUND

Across-chip linewidth variation is one of the most significant contributors to leakage power variability and timing uncertainty. Sources of ACLV include variations due to pitch, focus, exposure and random fluctuations (e.g., dopant density variation and gate oxide thickness variation).

OPC enables control of variations due to pitch and focus and is mandatory before mask manufacturing in VLSI design flows at present technology nodes. OPC controls printed shapes on wafer by applying corrections to design features based on proximity effects at nominal defocus conditions. Though OPC is extremely effective in linewidth control at nominal defocus conditions, it can result in significant linewidth variation at other defocus conditions. This variation, caused by defocus and pitch is systematic and can be modeled and the predicted and compensated.

2.1 Systematic ACLV

The *Bossung plot* in Figure 1 shows the variation of post-OPC printed linewidth at different pitch and defocus conditions.

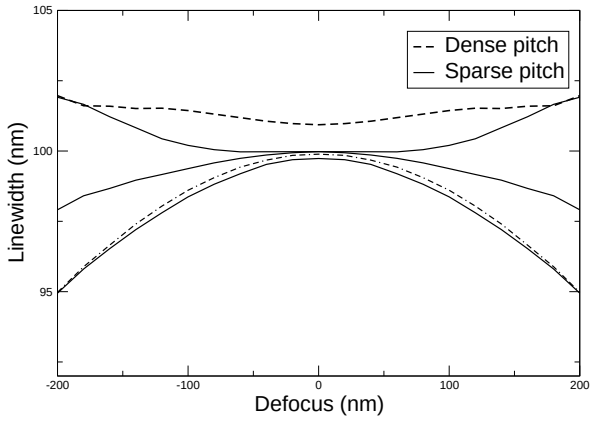

Figure 1: Linewidth variation with defocus for patterns with different pitches. Dense pitch corresponds to the minimum pitch of $250nm$ between poly lines in a 5-bar test pattern and sparse pitch corresponds to a pitch of $650nm$ between poly lines in a 3-bar test pattern. Linewidth increases with defocus for dense patterns, and decreases for isolated patterns.

We observe that dense lines tend to "smile" with defocus, while isolated lines "frown". The decrease in linewidth for isolated patterns is greater than the increase in linewidth for dense patterns over the same range of defocus despite the use of scattering bars (assist features) in our OPC recipes. Hence, isolated lines tend to become more leaky than their dense counterparts. Defocus is caused due to several sources, such as variation in shallow trench isolation (STI) layer thickness during chemical mechanical planarization (CMP), lens aberrations, wafer stage misalignment and resist thickness variation. Linewidth variation caused by defocus due to thickness variation can be modeled systematically by layout density analysis. For a particular value of defocus, linewidth increases for dense pitches and decreases for isolated pitches respectively, due to the presence or absence of proximity effect.

2.2 Topography Simulation

A schematic of topography-dependent defocus during lithography is shown in Figure 2. If the image plane of the reticle and lens system coincides with the wafer plane, the image prints with high resolution. However, in the regime of topography variation, caused predominantly by erosion and dishing effects during CMP, image prints out of focus leading to topography-dependent linewidth. Other optical and mechanical effects such as wafer stage misalignment, substrate flatness, and field tilt variation result in additional variations that are random or difficult to model.

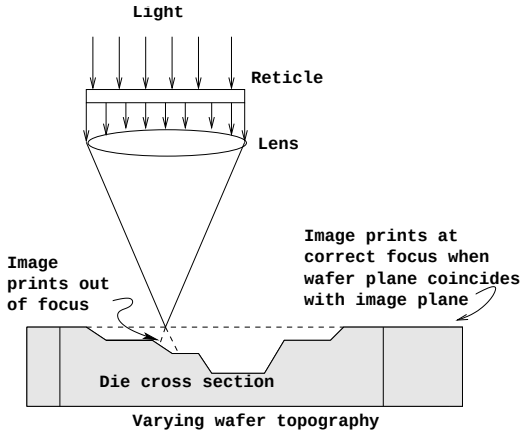

Figure 2: Focus variation due to non-planar wafer topography. Substrate thickness due to CMP effects (dishing and erosion) varies according to the density of STI (active) layer. Lens aberrations, misalignment of wafer plane axis and lens axis, and variation in distance between lens plane and wafer plane (stage error) add to the defocus.

Topography simulation has been the focus of several recent works [12, 11]. An analytical model of post-CMP oxide thickness based on layout density is described in [12]. The paper defines a parameter called "planarization length" that characterizes a CMP process. Planarization length is modeled as a function of long-range CMP pad deformation and pressure distribution during CMP. A pattern-density dependent analytical model is used to predict time-dependent oxide thickness evolution over different regions of the chip. CMP simulators based on the methods described in this paper can be used to analyze topographies of complex layouts. Since CMP simulation is a detailed procedure in itself, we assume that chip-level topography map is given as input for use in leakage estimation.

3. DEFOCUS-AWARE LEAKAGE ESTIMATION

In this section we describe our defocus-aware leakage estimation methodology, explain our experimental setup, and present results.

3.1 Methodology

Figure 3 illustrates our leakage estimation methodology. We use the following two flows depending on the availability of die topography information.

- *Defocus-aware, topography-oblivious flow.* We do not rely on a CMP simulator and assume the defocus (due to topography and other sources) to be random. In this flow, we use the fact that linewidth variation is more only for devices with dense or sparse pitches. Devices that have medium pitch, or high pitch on one side and sparse pitch on another print with lesser linewidth variation.

- *Defocus-aware, topography-aware flow.* In this flow, we consider a topography map from a CMP simulator. Since topography is a significant contributor to defocus variation, improved topography prediction leads to improved defocus prediction and consequently better leakage estimation.

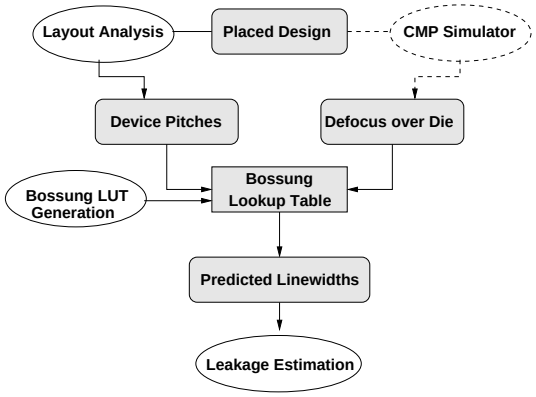

Figure 3: The proposed defocus-aware leakage estimation flow.

Both our flows find the pitch of each device in the given design and use it with the defocus level (assumed completely or partly random depending on the flow) at that device location to predict its linewidth. The predicted linewidths are then used for leakage estimation of each cell and then of the entire design. The main components of our flow are: (1) Bossung lookup table (LUT) generation, (2) layout analysis for pitch calculation, and (3) cell leakage estimation.

3.1.1 Bossung LUT Creation

The Bossung LUT captures systematic variations in linewidth due to pitch and defocus. We use it to predict linewidth at a certain pitch and defocus. Bossung LUT creation is an offline process that needs to be redone only when OPC or lithography simulation recipes change. To create the LUT, we construct line-and-space patterns of gate poly with different spacings to simulate different pitches. The linewidth of gate poly in each pattern is fixed at 100*nm* which corresponds to the gate length of TSMC 90*nm* technology. Line pitch is varied from 250*nm* (the minimum spacing at this technology node) to 850*nm* in steps of 100*nm*. In each pattern, there is one gate poly feature that we call the poly of interest with two identical neighbors on each side at various spacings, to get a total of five features in each pattern. Next, for each pattern, neighbors that are away from the poly of interest by more than 800*nm* are removed. It is safe to discard distant neighbors because the 193*nm* steppers used for patterning features in the 90*nm* technology node have an *optical radius* (radius of influence) of approximately 600*nm* (i.e., features separated by more than 600*nm* have negligible impact on each others printing). We conservatively use 800*nm* as the optical radius for all our experiments. We utilize symmetry of patterns to significantly cut down their number to a total of 153. After the creation of line-and-space patterns, we perform OPC of the patterns with zero defocus using Calibre OPC. To measure linewidth variation due to defocus, we then perform lithography simulation at different defocus levels for all the patterns. We choose defocus values in the range of $(-200nm, 200nm)$ in steps of $20nm$. Poly linewidth values are then extracted from all simulated printed images at each defocus level. Resist models for each defocus level are generated with a numerical aperture (NA) of 0.7 in Calibre WorkBench and we set the resist threshold to 0.38; both values fall in their standard ranges.

Our Bossung LUT contains rows corresponding to patterns and columns corresponding to defocus levels. Entries in the table give printed linewidth values for the feature of interest in the pattern. Linewidth variation with defocus for different patterns is illustrated in Figure 1. For dense patterns, we observe the linewidth to increase by up to $2nm$. For sparse or isolated patterns, on the other hand, we observe a reduction in linewidth of up to $6nm$. These observations are in line with previously reported trends [8].

3.1.2 Pitch Calculation

We use the design's standard-cell placement along with the position of devices within each cell instance to compute pitches. We use three measurements: (1) spacing between the cell and its neighbors (inter-cell spacing), (2) orientation of the cell and its neighbors, and (3) spacings between each device and the boundaries of the cell that contains the device (device-to-boundary spacing). Figure 4 illustrates three neighboring cells with inter-cell and device-to-boundary distances shown. Spacing between devices of a cell can be easily computed by taking the difference between their respective device-to-boundary spacings for a given boundary. We note that spacings between devices that belong to the same cell need to computed only once for each standard cell master (cell type). Spacing computation between devices of different cells involves adding the inter-cell distance between the two cells and the distance of the two devices from their corresponding cell boundaries, with careful consideration of the cell orientations.

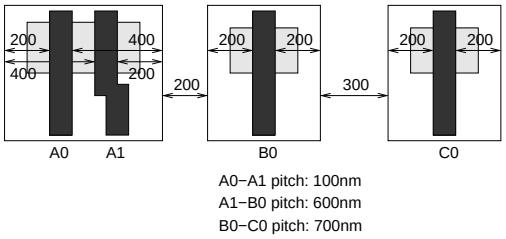

Figure 4: Pitch computation from a design layout.

3.1.3 Cell Leakage Estimation

Our cell leakage estimation methodology is identical to [13]. We pre-characterize leakage values using SPICE simulations for PMOS and NMOS devices of unit-width for all linewidths we are likely to encounter ($60nm$ to $140nm$). To find the cell leakage, we first compute the *average leakage* of each device in it over all input patterns of the cell, and then add the average leakages of all devices. The average leakage of a device is the pre-characterized leakage of a unit width device with the same gate length, multiplied by the fraction of patterns in which the device leaks. The fraction of patterns in which a device leaks is found by propagation of logic values inside the cell. Patterns for which a device is

turned off and has different logic values on its drain and source terminals are assumed to cause leakage in that device. We ignore the leakage of stacked devices since it is orders of magnitude less than non-stacked devices due to self reverse-biasing of stacked devices [10]. If state-dependent leakage estimation is desired, our cell leakage estimation approach be can easily modified to weight the device leakages for different input patterns by state probabilities. To compute the design leakage, we sum up leakages of all cells.

3.2 Experimental Setup

We validate our leakage estimation flow on c5315 (2,077 cells), c6288 (4,776 cells), and c7752 (3,155 cells) from the ISCAS'85 test suite, as well as alu128 (11,724 cells) from opencores.org. The designs were synthesized using Synopsys Design Compiler $v2003.06 - SP1$ using a small standard cell library of 20 cells under tight delay constraints. Our library is composed of the 20 most frequently used cells in our test cases[1]. To create the Bossung LUT, we use Mentor Calibre $v9.3\_5.9$ for OPC and lithography simulation. Our industry-strength OPC and lithography simulation recipes are for 100*nm* linewidths using 193*nm* stepper. We insert scattering bars (assist features) to improve the process window. We use Synopsys HSPICE $vU2003.09$ for all our SPICE simulations and Cadence SignalStorm $v4.1$ for library characterization with BPTM BSIM3 SPICE models [1, 4]. We place the designs with Cadence SOC Encounter $v3.2$.

For the defocus-aware, topography-aware flow, we assume the topography of Figure 5 as an input. The topography height is 100*nm* at the center of the die and quadratically reduces with distance from the center to become $-100nm$ at the die corners. A topography variation of $\pm 100nm$ is well within the defocus tolerance and expected to exist. In practice, the topography should be predicted by a CMP simulator that models STI layer planarization (STI-CMP simulator).

In addition to the observed variation in linewidth of $-6nm$ to $2nm$ due to defocus, we assume an additional random variation of $\pm 8nm$ to get a linewidth range from 86*nm* to 110*nm*.[2] Our assumptions are in line with the findings of [7]. In the defocus-aware, topography-aware leakage estimation flow, we assume that only half of the variation in defocus is due to topography and that the other half is Gaussian-random. For our defocus-aware, topography-oblivious leakage estimation flow, we assume the defocus variation to be completely Gaussian-random.

3.3 Results

Table 1 shows the leakage estimation at worst, nominal and best process corners using: (1) traditional, (2) the proposed defocus-aware, topography-oblivious, and (3) the proposed defocus-aware, topography-aware leakage estimation flows. Traditional leakage estimation assumes all devices to have a linewidth of 86*nm* for the worst, 100*nm* for the nominal, and 110*nm* for the best process corners. In defocus-aware, topography-oblivious leakage estimation, we assume random defocus between $-200nm$ and $200nm$. The defocus value is used with pitch information for the devices in the standard cell to predict linewidths at worst, nominal and best process corners for all devices. For defocus-aware, topography-aware leakage estimation, we use the topography height as the nominal defocus and add $-100nm$ to $100nm$ random variation in defocus to compute the corner-case leakages. A variation of $-8nm$, $0nm$, or $8nm$ due to random sources is then added to the linewidth for worst case, nominal and best case leakage estimation respectively.

[1]To identify the most frequently used cells, we first synthesize our test cases using the entire TSMC 90*nm* standard cell library.

[2]It is not appropriate to find the standard deviation in linewidth due to the two sources by the "square root of sum of squares" method because contribution due to defocus is partly modeled by our approach and the remainder is not close to Gaussian.

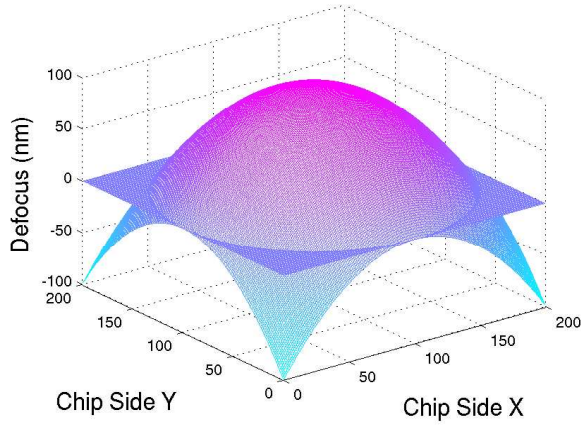

Figure 5: Die topography used in our experiments. Maximum height is $100nm$ higher than nominal (illustrated by the plane) at the center and decreases quadratically with distance from the center to become $100nm$ below nominal at the die corners.

In addition to accurate design leakage estimation, our methodology predicts the individual cell (or device) leakages for each cell (or device) more accurately. Figure 6 shows the distribution of the difference between cell leakage predicted by traditional estimation and our methodology at nominal process conditions for test case *c6288* when the topography map of Figure 5 is used. We observe cell leakage estimation errors in the range of -29% to 124%. While there is large error in cell leakage estimation at nominal process conditions, we note that the error in overall *circuit* leakage estimation is only -1.86%.

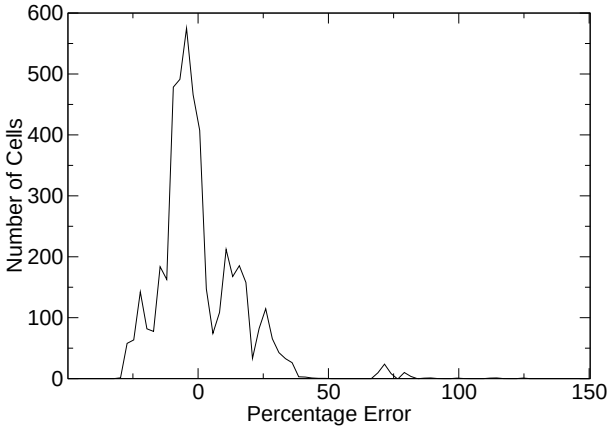

Figure 6: Distribution of leakage estimation error between traditional and defocus-aware flows for test case *c6288* at the nominal process corner. The error in overall *circuit* leakage is -1.86% (traditional is higher).

Our improved cell leakage prediction can be used to improve the quality of leakage reduction techniques that selectively optimize the cells (or devices) with high leakage, such as input-vector control, V_{th} assignment and gate length biasing.

Table 1: Estimated leakage at worst, nominal and best process corners using (1) traditional, (2) topography-oblivious, defocus-aware, and (3) topography-aware (assuming the topography of Figure 5), defocus-aware leakage estimation flows. Leakage values when the entire circuit uses only low V_{th} devices and when it uses only nominal V_{th} devices are shown.

Circuit	V_{th}	Traditional			Defocus-Aware, Topography-Oblivious				Defocus-Aware, Topography-Aware			
		WC (mW)	Nom (mW)	BC (mW)	WC (mW)	Nom (mW)	BC (mW)	Spread Reduction	WC (mW)	Nom (mW)	BC (mW)	Spread Reduction
c5315	Low	8.006	0.956	0.304	5.269	0.853	0.337	35.96%	4.119	0.889	0.337	50.90%
	Nom	1.481	0.125	0.036	0.931	0.111	0.040	38.34%	0.675	0.116	0.040	56.06%
c6288	Low	19.540	2.308	0.726	15.298	2.158	0.838	23.14%	11.256	2.265	0.838	44.63%
	Nom	3.625	0.302	0.086	2.827	0.282	0.101	22.97%	1.897	0.297	0.101	49.25%
c7552	Low	12.327	1.469	0.465	9.541	1.360	0.533	24.06%	7.126	1.433	0.533	44.42%
	Nom	2.281	0.192	0.055	1.757	0.177	0.064	23.94%	1.203	0.188	0.064	48.83%
alu128	Low	48.499	5.771	1.826	27.264	4.985	1.987	45.84%	22.442	5.153	1.987	56.17%
	Nom	8.978	0.754	0.217	4.574	0.644	0.238	50.51%	3.577	0.668	0.238	61.89%

4. DEFOCUS-AWARE GATE LENGTH BIASING

In this section, we begin with a primer on the previously proposed technique of gate length biasing for runtime leakage and its variability reduction [9]. We then describe our methodology to add defocus awareness to gate length biasing and present results.

4.1 Traditional Gate Length Biasing

Gate-length biasing exploits the fact that leakage reduces exponentially while delay increases only linearly with increase in gate length. To have minimal impact on circuit delay, the technique selectively biases only the devices that belong to cells that are not on timing-critical paths. Biasing a cell increases its delay and may cause some non-critical paths to become critical and consequently prevent other cells on the new critical paths from getting biased. The authors use a *sensitivity*-based greedy solution in which cells are iteratively biased in the order of their decreasing sensitivity. The sensitivity is defined as the ratio of leakage reduction and delay increase of a cell caused by biasing. If biasing a cell causes a timing violation, the cell is unbiased (i.e., its gate length is set back to nominal). The algorithm continues until no more cells can be biased. Sensitivity-based algorithms have also been used for V_{th} assignment [14] and gate width sizing [6]. We add defocus awareness by using the leakage estimated by our defocus-aware leakage estimation methodology in the sensitivity computation done by gate length biasing.

4.2 Methodology

We use the following terminology to explain our modifications to the sensitivity function for gate length biasing.

- L_p represents the leakage of cell instance p, L_p^n its leakage at the nominal process corner, and $\langle L_p \rangle$ its expected leakage.
- L_{pt} represents the leakage of the t^{th} device of cell instance p, and L_{pt}^n and $\langle L_{pt}^n \rangle$ are its nominal process corner and expected leakages, respectively. ($L_p = \sum_i L_{pt}$, where the summation is taken over all devices of the cell.)
- ΔL_p^n and $\Delta\langle L_p \rangle$ represent the change in nominal and expected leakages due to biasing cell instance p (i.e., biasing all devices in cell instance p).
- Δd_p is the change in delay of cell instance p after biasing it at the nominal process corner.

The sensitivity S_p in traditional gate length biasing is the ratio between leakage reduction and delay increase of cell p upon biasing, and is given by:

$$S_p = \frac{\Delta L_p^n}{\Delta d_p} \tag{1}$$

The sensitivity in defocus-aware leakage estimation is given by:

$$S_p = \frac{\Delta \langle L_p \rangle}{\Delta d_p} \tag{2}$$

To compute the expected leakage, we have two flows that are similar to the flows used for defocus-aware leakage estimation and depend on the availability of topography simulation. For the defocus-aware, topography-aware flow, we assume defocus to be a Gaussian random variable centered at the topography height given as an input from the STI-CMP simulator and with a 3σ of $100nm$ (50% of our defocus variation budget). For the defocus-aware, topography-oblivious flow, we consider defocus variation to be completely Gaussian random with a mean of $0nm$ and 3σ of $200nm$. We model leakage as a function of linewidth, which in turn is a function of pitch and defocus. Therefore,

$$L_{pt} = \mathcal{L}(\ell(D_{pt}, P_{pt})) \tag{3}$$

where, D_{pt} and P_{pt} are respectively the defocus and pitch for device t of cell p, and $\ell(D_{pt}, P_{pt})$ represents its linewidth. We may now write the expected leakage as:

$$\langle L_p \rangle = \sum_t \langle L_{pt} \rangle \tag{4}$$

$$\langle L_{pt} \rangle = \sum_t \sum_{D_{pt}} \mathcal{L}(\ell(D_{pt}, P_{pt})).\mathcal{P}(D_{pt}) \tag{5}$$

where $\mathcal{P}(D_{pt})$ is the probability that D_{pt} is the defocus value.

4.3 Results

A comparison between traditional and defocus-aware (topography-aware) gate length biasing is presented in Table 2. While we assume only defocus to be random during optimization (to exploit the systematic dependence of linewidth on defocus and pitch), we present results for the three process corners as described in Section 3. The delay penalty for gate length biasing is set to 0% (i.e., circuit delay does not increase after biasing). The runtime penalty due to defocus awareness is under 10% for all our test cases.

Table 2: Leakage after traditional and defocus-aware gate length biasing. Leakage optimization is done for nominal process-corner and the topography of Figure 5.

Circuit	Traditional Gate-Length Biasing			Defocus-Aware Gate-Length Biasing			Leakage Reduction		
	WC (mW)	Nom (mW)	BC (mW)	WC (mW)	Nom (mW)	BC (mW)	WC (%)	Nom (%)	BC (%)
c5315	3.948	0.855	0.326	3.838	0.838	0.321	2.78	2.01	1.63
c6288	9.363	1.923	0.730	8.958	1.861	0.712	4.33	3.23	2.56
c7552	6.678	1.350	0.507	6.212	1.280	0.485	6.98	5.17	4.21
alu128	21.258	4.908	1.907	19.968	4.663	1.827	6.07	4.99	4.19

Our results show modest leakage reductions for all three process corners from 1.63% to 6.98%. However, given that we have made only minor changes to only the sensitivity function of gate length biasing, we consider these results encouraging. Our approach may be used with several other leakage optimization approaches that rely on identifying candidate cells or devices to make tradeoffs. Significantly larger leakage reductions are expected when the impact of systematic linewidth variations on gate delays is also considered during optimization.

5. CONCLUSIONS

Due to the exponential dependence of leakage on linewidth, pessimism in linewidth translates to large leakage pessimism and overdesign. There is a need to model systematic components of linewidth variation for improved leakage estimation.

Our leakage estimation methodology models the pitch- and defocus-dependent systematic components of linewidth variation. We analyze a layout to calculate device pitches and use them with defocus and a pre-characterized Bossung lookup table to predict printed linewidths and to estimate leakage with increased accuracy. Our defocus-aware, topography-oblivious flow does not rely on an STI-CMP simulator and assumes defocus variations to be random. It considers device pitches to predict linewidth and consequently leakage with improved accuracy. The defocus-aware, topography-aware flow uses STI-CMP simulation to better predict defocus variation to further improve leakage estimation. Our methodology reduces the spread between leakage estimation at worst and best process corners by over half, and can estimate leakages of individual devices with improved accuracy.

Leakage optimization techniques that rely on leakage estimation of individual cells or devices can benefit from the defocus-aware leakage estimation flow. We enhance the previously proposed gate length biasing methodology that relies on leakage estimation of individual cells to determine the order in which cells are biased. Defocus-aware gate length biasing has larger leakage reductions than traditional gate length biasing by up to 7%.

Our ongoing work explores several ways to improve leakage estimation accuracy and to apply improved leakage estimation to enhance other leakage reduction techniques. For example, defocus variation has a significant systematic component arising from lens aberrations during wafer processing stage[7]. Lens aberrations are machine-dependent and modeling them requires details of lens parameters. Depending on the size of the die, lens aberrations can lead to inter-die or intra-die variations. Chips that are larger than $2.0cm$ on a side (e.g., processors) span lengths that are greater than the field length of wafer stepper, resulting in inter-die variations. For smaller chips, lens aberrations lead to inter-die variations and modeling them will reduce pessimism in leakage estimation.

6. REFERENCES

[1] Berkeley predictive technology model. http://www-device.eecs.berkeley.edu/∼ptm.

[2] A. Agarwal, C. H. Kim, S. Mukhopadhyay and K. Roy. Leakage in Nano-Scale Technologies: Mechanisms, Impact and Design Considerations. In *Proc. ACM/IEEE Design Automation Conference*, pp. 6-11, 2004.

[3] S. Borkar, T. Karnik, S. Narendra, J. Tschanz, A. Keshavarzi and V. De. Parameter Variations and Impact on Circuits and Microarchitecture. In *Proc. ACM/IEEE Design Automation Conference*, pp. 338-342, 2003.

[4] Y. Cao, T. Sato, M. Orshansky, D. Sylvester and C. Hu. New Paradigm of Predictive MOSFET and Interconnect Modeling for Early circuit Design. In *Proc. Custom Integrated Circuits Conference*, pp. 201-204, 2000.

[5] R. Choudhury. Handbook of Microlithography and Micromachining Volume:1. In *SPIE Press Monograph*, 2002.

[6] J. Fishburn and A. Dunlop. Tilos: A posynomial programming approach to transistor-sizing. In *Proc. IEEE Intl. Conf. on Computer Aided Design*, pp. 326-328, 1985.

[7] D. G. Flagello, H. van der Laan, J. van Schoot, I. Bouchoms and B. Geh. Understanding systematic and random cd variations using predictive modelling techniques. In *SPIE Conf. Optical Microlithography XII*, pp. 162-175, 1999.

[8] P. Gupta and F.-L. Heng. Toward a Systematic-Variation Aware Timing Methodology. In *Proc. ACM/IEEE Design Automation Conference*, pp. 321-326, 2004.

[9] P. Gupta, A. B. Kahng, P. Sharma and D. Sylvester. Selective Gate-Length Biasing for Cost-Effective Runtime Leakage Control. In *Proc. ACM/IEEE Design Automation Conference*, pp. 327-330, 2004.

[10] W. Jiang, V. Tiwari, E. Iglesia and A. Sinha. Topological Analysis for Leakage Prediction of Digital Circuits. In *Proc. IEEE Asia and South Pacific Design Automation Conference*, pp. 39-44, 2002.

[11] B. Lee. *Modeling of Chemical Mechanical Polishing for Shallow Trench Isolation.* PhD thesis, Massachusetts Institute of Technology, 2002.

[12] D. O. Ouma, D. S. Boning, J. E. Chung, W. G. Easter, V. Saxena, S. Misra and A. Crevasse. Characterization and Modeling of Oxide Chemical Mechanical Polishing Using Planarization Length and Pattern Density. In *IEEE Tran. on Semiconductor Manufacturing*, Vol. 15(2), pp. 232-244, 2002.

[13] R. M. Rao, J. L. Burns, A. Devgan and R. B. Brown. Efficient Techniques for Gate Leakage Estimation. In *Intl. Symp. on Low Power Electronics and Design*, pp. 100-103, 2003.

[14] S. Sirichotiyakul, T. Edwards, C. Oh, R. Panda and D. Blaauw. Duet: An Accurate Leakage Estimation and Optimization Tool for Dual-V_{th} Circuits. *IEEE Tran. on VLSI Systems*, vol. 10, pp. 79-90, April 2002.