

IBM Research Report

Toward a Systematic-Variation Aware Timing Methodology

Puneet Gupta
Department of ECE
University of California at San Diego
San Diego, CA

Fook-Luen Heng
IBM Research Division
Thomas J. Watson Research Center
P.O. Box 218
Yorktown Heights, NY 10598



Research Division
Almaden - Austin - Beijing - Haifa - India - T. J. Watson - Tokyo - Zurich

Toward a Systematic-Variation Aware Timing Methodology

Puneet Gupta [†] and Fook-Luen Heng [‡]

[†]Department of ECE, UC San Diego, CA, USA

[‡] USA IBM T.J. Watson Research Center, Yorktown Heights, NY, USA

{puneet@ucsd.edu, heng@us.ibm.com}

Abstract

Variability of circuit performance is becoming a very important issue for ultra-deep sub-micron technology. Gate length variation has the most direct impact on circuit performance. Since many factors contribute to the variability of gate length, recent studies have modeled the variability using Gaussian distribution. In reality, the through-pitch and through-focus variation of gate length are systematic. In this paper, we propose a timing methodology which takes these systematic variations into account and we show that such approach can reduce the timing uncertainty by up to 40%.

1 Introduction

The ability to more accurately model and manage variability of designs in ultra-deep sub-micron technology has become ever more critical in the success of technologies beyond 90nm CMOS process. Since critical dimensions are scaling faster than our ability to control them, e.g. effective gate length of a transistor, variability has become an increasingly more important design issue [1, 2]. It is recognized that traditional static timing approach is becoming too conservative to predict the actual performance of a design [3, 1, 4, 5]. Progress has been made to employ statistical techniques to model variability of circuit performance. A general probabilistic framework has been proposed to improve the accuracy of timing prediction [3]. Several approaches to address the correlations due to path re-convergence and proximity gates are studied [1, 4, 6].

Across Chip Linewidth Variation (ACLV) is a major contributor to timing variation in ultra-deep sub-micron technology. Other sources of variation includes metal thickness, temperature, voltage, oxide thickness etc. In this paper we focus on the systematic components of ACLV for the polysilicon level. There are many sources which contribute to ACLV: through-pitch variation, through-process variation, topography variation, mask variation, etching etc. Due to the complex interaction between these sources of variation, ACLV has been modeled as a random phenomena [5]. In reality, at least 50% of ACLV is systematic [9, 8]. The systematic through-pitch variation is the major contributor to variation at nominal process condition, and the systematic through-focus variation is the major contributor for through process condition. These systematic variations can be modelled very accurately once a physical layout is completed.

Static timing analysis based on worst-case timing is a

common sign-off process adopted in ASIC. In reality, the worst-case timing is never achieved in actual hardware. One reason is because the worst-case timing approach assumes ACLV of transistors is independent, which is never the case. This is addressed by various statistical timing approaches [1, 6, 3]. The other reason is because worst-case timing model does not consider the systematic components of ACLV which can be predicted accurately based on physical context of the gates. In this paper, we investigate the systematic components of ACLV, its magnitude and timing impact. We propose a systematic-variation aware timing methodology. We show that by taking into consideration the systematic variations, we can reduce the best-case to worst-case timing spread by up to 40% in a traditional static timing analysis. Similar impact of this type of systematic aware modeling in statistical timing analysis can also be expected but it is not covered in this paper.

In the next section, we discuss the systematic ACLV in more details. Section 3 describes our experiment to quantify the timing impact of the systematic variation in benchmark circuits. Section 4 shows our results. Then in section 5, we describes future improvement of a systematic variation aware timing methodology. Finally, in section 6, we summarize our finding and describe future work.

2 Systematic Variation: Magnitude and Impact

Through-pitch linewidth variation at nominal process condition, can best be demonstrated by a typical plot of linewidth versus pitch such as Figure 1. The plot shows printed linewidth systematically decreases as the pitch increases, up to the radius of influence. Optical proximity correction (OPC) is a technique used to correct this systematic effect. OPC is a necessary VLSI mask data processing step in today's technology [12]. It attempts to correct the distortion of printed image due to proximity environment of the designed shapes at nominal process condition.

While the correction reproduces the intended design shapes on wafer as best as possible, it is not perfect. We demonstrate even with OPC, there is systematic linewidth variation at nominal process condition. This is done by applying standard OPC to parallel poly lines with different spacing, and then measure the average linewidth of the simulated wafer image of the corrected poly lines. Our result indicate a systematic decrease in linewidth as the pitch increases from 300nm to 600nm. The magnitude of the variation is about 10% of the target linewidth. This implies that the nominal timing model can have as much as 10% dis-

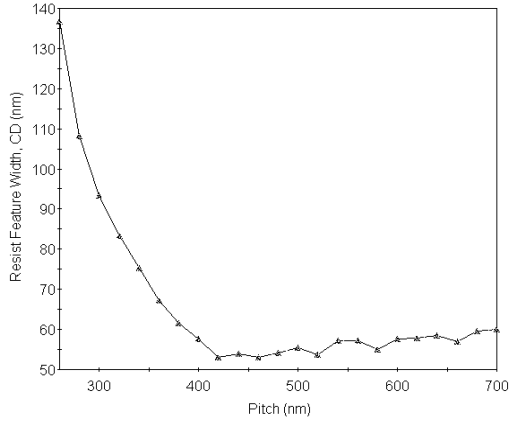


Figure 1: An example of through-pitch variation for an annular illumination system with $\lambda=193\text{nm}$ and $\text{NA}=0.7$ calculated using *Prolith* [13]. The drawn CD is 130nm. Notice the “radius of influence” of less than 600nm.

crepancy from the target linewidth, assuming delay varies linearly with gate length.

Through-focus linewidth variation is illustrated by a standard Bossung plot (e.g. Figure 2) of linewidth vs defocus condition. For a binary mask technology, the Bossung plot depicts opposite behavior of dense lines and isolated lines. For a dense line, the linewidth increases as the process go out of focus, the “smiling” part of the plot. For an isolated line, the linewidth decreases as the process go out of focus, the “frowning” part of the plot. This systematic effect is somewhat mitigated by insertion of assist features [11] but never completely. The through-focus variation can account for up to 30% of the total ACLV budget.

3 Overview of the Systematic Variation Aware Static Timing Methodology

Timing model for a standard-cell is characterized with very intensive simulation process. It is reduced to a set of formulas which predict delay of input to output paths based on parameters such as gate length, temperature, voltage, oxide thickness etc. The corners of the model assume worst-case condition for each parameter. In particular, worst-case gate length is assumed to be the maximum possible gate length variation. In reality, as described above, gate length variation can be predicted more accurately based on the spatial environment of each gate. The accurate prediction will remove at least half of the best-case to worst-case spread of the gate length. In this section, we describe a timing methodology which takes into consideration the systematic variation of gate length. We also quantify the pessimism caused by using the worst-case assumption.

3.1 Accounting for Pattern Dependent Variation

Traditional timing methodology assumes perfect printing of the gates and hence computes timing of a design based on the target gate length. Model-based OPC tries to achieve the target gate length but is never able to correct the design perfectly. The reasons may include OPC-unfriendly layout

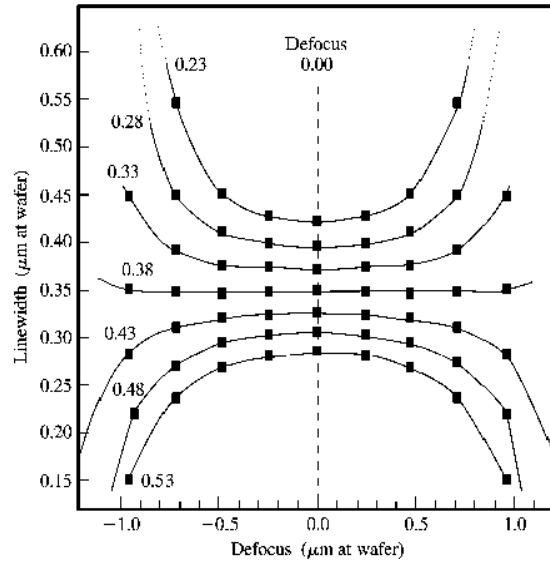


Figure 2: Linewidth vs. defocus for $0.35\ \mu\text{m}$ width with varying spacing for a given exposure setting [10]. Notice the systematically different behavior of isolated and nested lines.

patterns and limitations of the OPC algorithm as well as constraints on runtime. As a result there always is some iso-dense bias in printing of polysilicon shapes. Isolated lines tend to print smaller (or larger depending on the process) than nested or dense shapes. This pitch dependent variation of printed gate length is systematic and hence can be predicted. After placement spacing between all gate shapes is known and hence printed shapes can be predicted accurately.

OPC can be performed on the layout and lithography simulations can be done to predict the printed shape on the final wafer. The critical dimension or gate length can then be measured from this simulated print-image of the layout for each device. This more accurate gate length can then be used to predict the timing of the device, cell and hence the entire design more accurately. The problems with such an elaborate approach are as follows.

- *OPC is computation intensive.* Model-based OPC is very computation intensive. Typical numbers range from about 1100 seconds for a small 5900 gate design (see Table 1) to several CPU days for modern multi-million gate designs. Moreover, image simulation of the entire design is also very time consuming and hence not suitable for use during the design process which may involve many synthesis, P&R iterations.
- *Library characterization is an involved process.* Characterizing a standard cell for continuously varying gate lengths (or *Critical Dimension, CD*) of all the devices within it is a herculean task if not an impossible one. Performing circuit-level timing on the entire design with accurate gate-lengths is also not feasible due to runtime and scalability constraints.

Our method of accounting for through-pitch variation in static timing has three major components namely: accurate

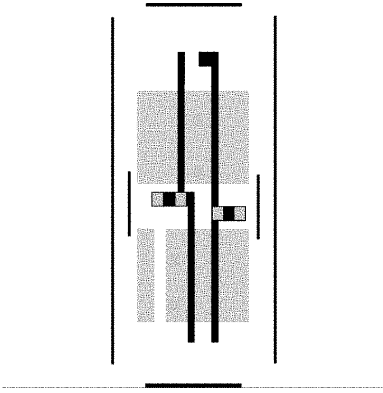


Figure 3: A library-based OPC environment setup for a simple NAND gate. Note the dummy poly geometries inserted to emulate the impact of neighboring cells on the cell under consideration.

CD measurement, constructing timing libraries and contextual timing analysis. We describe these parts of our flow next.

3.1.1 CD measurement

To circumvent the problems of full-chip OPC and elaborate characterization, we adopt a library based OPC approach similar to one described in [7]. Individual library cells are corrected conservatively in a typical placement environment. The placement environment is emulated using a set of dummy geometries. For example, see Figure 3. Further details can be found in [7]. The average gate length¹ is then measured for all devices in the gate. These “printed” gate lengths are then used to predict timing for the devices.

This library-based OPC approach is accurate enough because the radius of influence for 193nm steppers is about 600nm. I.e., features beyond 600nm of any given device have negligible impact on its printing. As a result, the devices which are not at the periphery of the cell have an environment which is almost identical to their actual placement environment. Therefore, the CD predicted for them after library-based OPC is very close to the CD predicted for them after full-chip OPC.

Devices which lie at the boundary of the cell are not as accurately predictable by the library-OPC approach. For these devices, we use a through-pitch CD simulation approach. We construct a look-up table which matches pitch to printed CD for the given process. The CD measurements are again done post-OPC. The empirical model is constructed for a number of spacings up to 600nm. The placement of the cell in layout determines the CD to be used for these border devices. An example is shown in Figure 4.

3.1.2 Constructing Timing Libraries

In a placement, a cell’s environment will depend on the neighboring cells (left and right in a horizontal cell placement row)² and the whitespace between the cell and its

¹The gate length varies along the width of the device. We do a simple averaging of the CD. We believe this to be a reasonable approximation as device delay varies almost linearly as gate length.

²We do not consider “vertical” neighbors as they have negligible impact on gate CD.

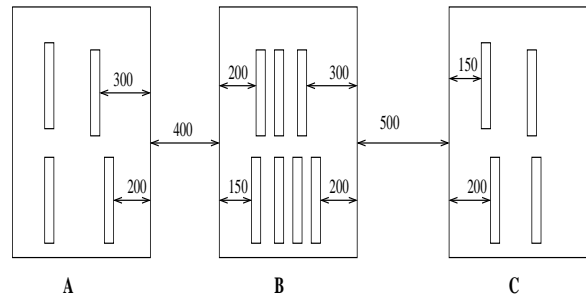


Figure 4: An example placement of cells A, B and C. For cell B, $nps_B^{LT} = 900$, $nps_B^{RT} = 950$, $nps_B^{LB} = 750$, $nps_B^{RB} = 900$.

neighbors.

In a placement for a cell C_i , its environment is described by a set of four spacings nps_i^{LT} (distance of the device on the “left-top” to the nearest poly feature on the left in the neighboring cell), nps_i^{RB} (distance of the device on the “right-bottom” to the nearest poly feature on the right), nps_i^{LB} and nps_i^{RT} .³ These four space parameters enable us to determine the printed CD for the border poly features in the cell in the placement context using the through-pitch CD simulation results. Since continuous variation of these parameters makes a library difficult to characterize, we use *three* different values each of these parameters. This gives rise to 81 different versions of the same cells.⁴

For our current experiments, we assume delay of any timing arc from an input pin to an output pin in a cell to be linearly proportional to the gate lengths of the devices involved in the transition. The devices involved are fixed for the worst-case transition. Though we use this linear approximation for simplicity, more accurate circuit simulation based analysis is also feasible. We construct timing look up tables (with varying load capacitance and input slews) for these 81 versions of the library cell master. As a result, we obtain a .lib which has 81 versions of each cell in the original library.

3.1.3 In-Context Timing Analysis

After the library generation, the next step is to identify correct canonical environment for every cell instance in the layout and perform a contextual static timing analysis. We define four parameters for a cell C_i : s_i^{LT} (the distance of cell outline from the closest device on the “left-top” corner of the cell), s_i^{LB} (spacing between left-bottom device and the cell outline), s_i^{RT} and s_i^{RB} . Analyzing the placement (i.e., whitespace around the cell and the four s parameters for the given cell and its immediate neighbors) puts the given cell in the given layout into one of the 81 categories.

After annotating each cell instance with its correct version, we run static timing analysis with the expanded library. The result of this timing analysis takes into account iso-dense effects and the resulting through-pitch variation at the nominal focus and exposure.

³Note that the top and bottom spacings can be different as they correspond to p and n devices respectively which may not be aligned in the cell layout.

⁴81 is arrived as a compromise between accuracy and ease of implementation.

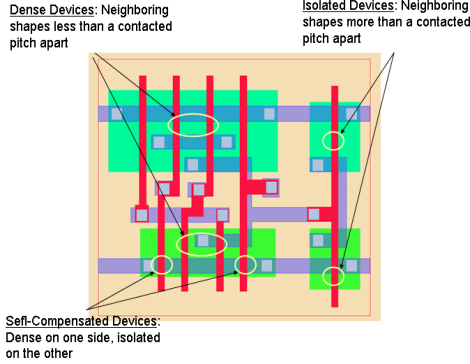


Figure 5: An example cell layout depicting isolated, dense and self-compensated devices.

3.2 Taming Focus Variation

The next systematic component of variation that we account for in our proposed timing analysis methodology is the CD variation arising out of focus variation. Isolated and dense lines behave differently with defocus. Isolated lines tend to get thinner with defocus while dense lines get thicker. As a result, isolated devices get faster with focus variation while dense devices tend to get slower than nominal.

An important component of “process” corner for timing is gate length variation. A very important component of gate length variation is focus variation. The systematic “smile-frown” behavior of focus-based variation of CD implies that depending on whether a certain timing arc involves isolated devices or dense ones, the worst-casing in one of its corners can be reduced. Moreover, there is some “self-compensation” of focus variation for timing arcs which involve both isolated and dense devices.

As before, we analyze the devices in the layout and label them as isolated, dense or self-compensated depending on the spacing to the nearest poly line on the left and the right.⁵ For example, a standard-cell layout with the three kinds of devices labeled is shown in Figure 5. Next we label each timing arc (input pin to output pin transition) as “smile”, “frown” or “self-compensated” depending on whether the devices involved in the transition are isolated, dense or self-compensated.⁶

We assume given a certain percentage contribution of focus variation to CD variation. For smiling timing arcs, we trim off that portion from the best-case gate length. For frowning timing arcs, the worst-case gate-length is reduced while for self-compensated timing arcs worst-case as well as best-case gate lengths are impacted. As a result, timing uncertainty arising out of focus variation is reduced for *all* timing arcs in the design.

⁵We assume “dense” spacing to be less than the contacted-pitch and anything larger to be “isolated”.

⁶For purpose of this work, we assume the majority determines the nature. For example, if a timing arc involves two isolated and one dense device, then it is labeled as frowning. Better focus-sensitivity based characterization is possible but we limit ourselves for want of an accurate defocus print-image simulator.

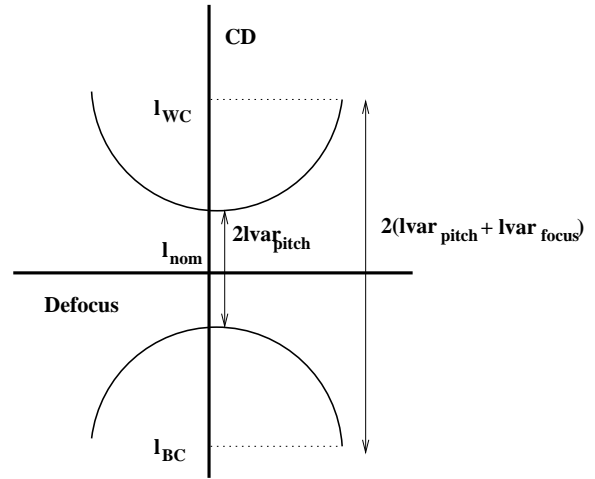


Figure 6: An artificial Bossung curve at some given nominal exposure. The smile denotes the “most dense” feature in the technology while the frown denotes the “most isolated” one. It should be clear that the total span of CD variation ($= 2(lvar_{pitch} + lvar_{focus})$) is too pessimistic.

3.3 Computing the Corners

Traditional corner-based timing analysis uses slow, nominal and fast corners for process. The systematic variation aware static timing analysis flow proposed in this work reduces the pessimism and uncertainty caused by these variations.

To compute the impact of through-pitch variation, we draw test layouts consisting of parallel poly lines with fixed width and length but varying spacing. These test layouts are then corrected with the standard OPC flow and CD is measured to construct the lookup table described in section 3.1. Denote the total range of CD variation after OPC by $\pm lvar_{pitch}$. We calculate (similarly defined) $\pm lvar_{focus}$ using the FEM (Focus Exposure Matrix) curves built from fabrication of test structures. We measure the CD variation with defocus (focus variation range is taken to be $\pm 300\text{nm}$) for a number of pitches (ranging from minimum pitch to a pitch slightly larger than the contacted pitch). These variations are shown in the artificial Bossung plot in Figure 6.

Let l_{new}^{nom} and l_{pitch}^{nom} denote the traditional nominal gate length (independent of the cell layout and placement) and the iso-dense aware gate length respectively. Define l_{pitch}^{WC} and l_{pitch}^{BC} to be the worst-case and best-case gate lengths after accounting for through-pitch variation in CD. Similarly, l^{WC} and l^{BC} be the corresponding numbers in the conventional flow. Then

$$\begin{aligned} l_{pitch}^{WC} &= l_{new}^{nom} + (l^{WC} - l^{nom} - lvar_{pitch}) \\ l_{pitch}^{BC} &= l_{new}^{nom} - (l^{nom} - l^{BC} - lvar_{pitch}) \end{aligned} \quad (1)$$

There are many factors affecting the best and worst case gate length, we are removing the variation due to pitch. In reality, there are dependency between the pitch and the non-pitch factors. For the purpose of quantifying the potential impact of taking into consideration the systematic variation, this is a very good first order assumption. We will discuss what can be done to improve the accuracy in an actual systematic variation aware timing methodology in section 5.

Focus variation does not affect the nominal process corner. Moreover it may affect worst-case and best-case corners differently depending on whether the timing arc under consideration is smiling, frowning or self-compensated.⁷ For smiling timing arcs, the values are

$$l_{smile}^{WC} = l_{pitch}^{WC} \quad (2)$$

$$l_{smile}^{BC} = l_{pitch}^{BC} + lvar_{focus}$$

Here, we are removing the variation due to focus from the best case, since it is not a factor for dense lines. Similarly for frowning timing arcs,

$$l_{frown}^{WC} = l_{pitch}^{WC} - lvar_{focus} \quad (3)$$

$$l_{frown}^{BC} = l_{pitch}^{BC}$$

For self-compensated arcs, both worst-case and best-case timing is modified.

$$l_{selfcomp}^{WC} = l_{pitch}^{WC} - lvar_{focus} \quad (4)$$

$$l_{selfcomp}^{BC} = l_{pitch}^{BC} + lvar_{focus} \quad (5)$$

4 Experiments and Results

To quantify the magnitude of the pessimism of traditional STA, we take 10 most frequency used cells in a 90nm standard-cell library, synthesize ISCAS85 benchmark circuits with the 10 cells, and then time the synthesized and placed circuits for best-case, nominal and worst-case. The corner case libraries are constructed with just the process corners while the voltage and temperature are kept the same across all the libraries. We do this to evaluate the benefit of the proposed timing methodology independent of any orthogonal effects.

We apply OPC to these 10 cell masters as described in section 3.1.1 using *Mentor Graphics Calibre*. Model-based OPC is performed using IBM 90nm pre-production process models. To verify that through-pitch variation is sizeable even after model-based OPC, we measure CDs of simulated full-chip standard model-based OPC and compare it with simulated nominal gate length. The distribution of error is given for an example circuit in Figure 7. We see up to 20% variation in printed gate length even after model-based OPC.

To evaluate effectiveness of the library-based OPC approach we compare the printed CD of library-based OPC with traditional full-chip OPC approach. The results are given in Table 1. The table shows that about 50% of all devices corrected in a library-based OPC fashion fall within 1% error while nearly all devices have a printed gate length within $\pm 6\%$ of full-chip OPC. Moreover, most of the error-prone devices are likely to lie on the periphery of the cell which are accounted for in a “rule-based” fashion in our timing methodology.

We perform in-context timing analysis for the synthesized and placed circuits with the in-context timing model described in section 3, by substituting the correct version of the timing model for each cell based on its placement. We generating the 81 versions of each cell as described in section 3.1 with values of nps^{LT} , nps^{RT} , nps^{LB} and nps^{RB}

⁷In this work we do not consider “degree” of compensation for the lack of supporting data.

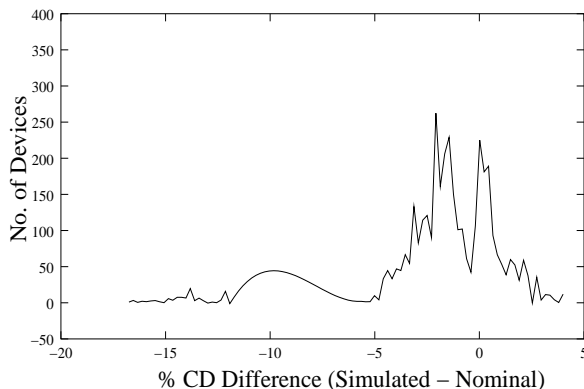


Figure 7: Distribution of error for model-based OPC for C3540 ISCAS85 benchmark.

Testcase	N-1%	N-3%	N-6%	Runtime (s)
C1355	58	83	97	477
C2670	45	78	96	747
C3540	40	77	96	1131
C432	35	76	97	185
C499	54	79	96	495

Table 1: Comparison of Library-based OPC and full-chip OPC. N-i% denotes % of devices with less than i% error compared to full-chip OPC. Library OPC Runtime is 90 seconds for 10 masters.

each being put into one of the three bins: {400-500nm, 500-600nm, ≥ 600 nm}. Since the radius of influence of 193nm steppers is about 600nm, any spacing larger than 600nm is isolated spacing and prints almost the same as a 600nm spacing. Since dense geometries print larger in the process, we use lower of the bin extremes (e.g., 400nm for 400-500nm bin) to be pessimistic in our timing estimates.

We compare the best-case, nominal and worst-case timing with the standard timing as described above. Assuming $lvar_{focus}$ and $lvar_{pitch}$ each to be 30% of the total gate length variation [8], the results of systematic-variation aware STA are shown in Table 2. Our results show that the best-case to worst-case timing spread is reduced by 28% to 40% in the systematic variation aware approach. Since majority of the devices in the layout are isolated (due to the whitespace distribution or the cell layout itself), the nominal timing improves when through-pitch variation is accounted for.

5 Practical Systematic Variation Aware Timing

Our experiment demonstrates that there is substantial pessimism in the traditional static timing analysis by not considering the systematic components of ACLV. In this section, we propose a practical systematic variation aware timing methodology.

In order to produce more accurate in-context timing model for each standard cell, each cell will need to be “corrected” by the OPC process before it is characterized. This can be done by the library based OPC methodology proposed in [7], in which, gates in the cell are corrected by standard OPC processed on a per cell definition basis as opposed to be corrected in a per instance basis. Gates on the boundary can have several versions of correction based

Testcase	#Gates	Traditional Timing (ns)			New "Accurate" Timing (ns)			% Reduction in Uncertainty
		Nom	BC	WC	Nom	BC	WC	
C1355	2058	2.15	1.57	2.88	2.15	1.70	2.62	29
C2670	3655	5.07	3.74	6.64	5.05	4.04	5.96	33
C3540	5903	6.32	4.72	8.34	6.26	5.20	7.35	40
C432	968	5.77	4.21	7.70	5.70	4.53	6.88	32
C499	1728	2.30	1.66	3.10	2.29	1.79	2.82	28

Table 2: Comparison of traditional worst-case timing with systematic variation aware timing methodology. Nom, BC, WC denote nominal, best-case and worst-case corners of the library respectively.

on context. In such an OPC methodology, the timing characterization of a cell can be performed based on the actual wafer image of the corrected gates in the cell.

Furthermore, we need to develop a parameterized gate length model for each gate on the cell boundary. The model will predict the actual gate length and its variation based on the proximity spatial information, i.e. distance of the neighboring gate. From our discussion in section 3, the nominal gate length can be predicted by through-pitch gate length simulation, and the through-focus gate length variation can be predicted by a Focus Exposure Matrix (FEM) plot.

A timing model which includes the proximity spatial information as a parameter for input to output path delay will need to be constructed. More specifically, the input to output delay is parameterized by s_i^{LT} , s_i^{LB} , s_i^{RT} , s_i^{RB} as described in section 3.1.3. One naive way to construct such a model will be to perform extensive input to output delay path simulation for each value of the boundary gate length. A more efficient construction of such a model is a topic which will require separate investigation.

With such a timing model parameterized by proximity spatial information, the systematic variation aware static timing analysis can be performed after placement.

6 Conclusions and Future Work

In this paper, we have proposed a novel static timing methodology which accounts for systematic variation arising due to proximity effects and focus variation. The methodology brings process and design closer and has elements of RET, library characterization as well as conventional static timing analysis. We quantify the magnitude of the pessimism of traditional static timing analysis which neglects systematic components of ACLV. This can amount to as much as 40% tightening of the best-case to worst-case timing spread. In practice, ASIC hardware always performs better than traditional STA predicts. Even though, different compensating mechanisms has been built into traditional STA, e.g. IBM EinsTimer [1], systematic variation could be one key component which contributes to the discrepancy as suggested by our results.

We are refining our experiment for process technology which includes other RET such as Sub-Resolution Assist Features. We also plan to further quantify such pessimism by using statistical timing methodology with more realistic gate length distribution based on iso-dense attributes and proximity spatial information, as opposed to the simplistic Gaussian distribution of gate length variation. Another process phenomenon not accounted for in our current experiments is exposure dose variation. Exposure variation can alter the nature of devices (i.e. dense or isolated).

Our current work also investigates the implications of exposure variation on the proposed timing methodology. Sys-

tematic nature of focus dependent CD variation suggests potential implications for compensating for such focus variation.

References

- [1] C. Visweswariah, "Death, Taxes and Failing Chips", *Proc. Design Automation Conference*, 2003, pp.343-347.
- [2] 2001 International Technology Roadmap for Semiconductors (ITRS), <http://public.itrs.net>.
- [3] M. Orshansky and K. Keutzer, "A General Probabilistic Framework for Worst Case Timing Analysis", *Proc. Design Automation Conference*, 2002, pp.556-561.
- [4] J.A.G. Hess, K. Kalafala, S.R. Naidu, R.H.J.M. Otten, and C. Visweswariah, "Statistical Timing for Parametric Yield Prediction of Digital Integrated Circuits", *Proc. Design Automation Conference*, 2003, pp. 932-937.
- [5] D. Blaauw, S. Nassif, L. Scheffer and A. Strojwas, "Design for Manufacturing in the Sub-100nm Era", *Design Automation Conference*, Tutorial.
- [6] A.B. Agrawal, D. Blaauw, V. Zolotov, and S. Vrudhula, "Statistical timing analysis using bounds and selective enumeration", *Proc. Design Automation Conference*, 2003, pp. 348-353.
- [7] P. Gupta, F.-L. Heng and M. Lavin, "Merits of Cellwise Model-Based OPC", *Proc. SPIE Conf. on Design and Process Integration for Microelectronic Manufacturing*, Feb. 2004, to appear.
- [8] William Chu, IBM Corp., *Personal Communication*, July 2003.
- [9] S. Postnikov and S. Hector, "ITRS CD Error Budgets: Proposed Simulation Study Methodology", May 2003.
- [10] R.A. Budd, D.B. Dove, J.L. Staples, R.M. Martino, R.A. Ferguson, and J.T. Weed, "Development and application of a new tool for lithographic mask evaluation, the stepper equivalent Aerial Image Measurement System, AIMS", *IBM Journal of R&D*, 41(12), pp. 119-129.
- [11] ASML MaskTools Inc., http://www.masktools.com/content/scat_bars.pdf
- [12] A.B. Kahng and Y.C. Pati, "Subwavelength Lithography and its Potential Impact on Design and EDA", *Proc. ACM/IEEE Design Automation Conf.* 1999, pp. 799-804.
- [13] Prolith version 8.0, <http://www.kla-tencor.com>