Why are Timing Estimates so Uncertain? What could we do about this?

Louis K. Scheffer*

For a specific digital gate in a specific modern technology, SPICE will give you a very specific timing number. But no one believes this number, and with good reason. First, there is a huge, and growing, dependence on the local design environment. This enviroment may not be known when the cell is designed, and may vary by instance. Next, even if the design is complete, there are manufacturing uncertainties. These range from statistical fluctuations, to process drifts, to intra-die and inter-die systematics, to the unknown characteristics of unbuilt fabs. Finally, other factors like supply voltage and temperature change in operations, and the circuits age.

Dealing with this uncertainty is one of the main problems of modern VLSI design, and currently has no satisfactory solution. The simple “worst case” approach is almost universally acknowledged to be far too conservative. A great deal of effort has gone into statistical timing, but many of the uncertainties are not statistical in nature.

What is needed is a flow that takes uncertainty into account as a first-class design object. At any point, the designer must be able to fix what is known, plug in assumptions about what is not known, and predict the resulting performance. Keeping sensitivities is a natural way to do this since by definition they predict what will happen when a parameter is changed, allowing the designer to account for what is known and bound the effects of the unknown. Sensitivities have been used for many years in statistical timing and analog design, where their main advantage is the ease of handling correlated random variables. However, sensitivities are perhaps even more useful in dealing with deterministic and systematic effects. Here we propose a flow that systematically addresses uncertainties by measuring, storing, and using sensitivities. Such a flow preserves the use of hierarchy (crucial for productivity), retains the isolation of design and usage, supports both worst case and statistical design, and supports optimization directly. The main disadvantage to such a flow, and the the increased use of sensitivities, is the need for format changes and new tool algorithms.

1. Introduction

Modern IC design is fraught with uncertainties. Smaller devices naturally lead to larger (relative) uncertainties. In addition, as process nodes shrink, all devices, intended and parasitic, show a larger dependence on their environment. Some of these uncertainties are:

- The exact size of each fabricated geometry depends on the local litho and etching environment.
- A cell may be included horizontally or vertically. This often causes a systematic L dependence.
- The thickness of the metal and dielectric layers depends on the local CMP environment.
- Device and interconnect sizes also depend on where in the optical field the device lies.
- Deposition and etching steps typically vary across the wafer surface - even real time control can only get the thickness exactly right at one spot.
- Performance of circuits depends on supply voltages and temperatures, which vary during operation of the chip.
- The performance may change further during the life of the chip due to slow material changes such as electromigration or NTBI.
- The fab in which the chip will be built may be unknown, and perhaps not yet constructed. Even if existing, the fab performance and statistics may be time varying.
- Finally, there may be statistical variations, not predicted by any of the above factors.

Some of these factors may be known at design time (litho environment), some are hard to predict until the design is complete (CMP environment), some vary during operation (temperature and voltage), some are not known until years

*Louis Scheffer is with Cadence Design Systems, 555 River Oaks Parkway, San Jose, California, USA. Email: lou@cadence.com

later (NTBI), and some are outside the designers control entirely (position of the die on the wafer).

How can designers cope with these uncertainties? What is needed is a flow that takes uncertainty into account as a first-class design object. At any point, the designer must be able to fix what is known, plug in assumptions about what is not known, and predict the resulting performance.

The common thread to keeping track of all the above uncertainties is keeping sensitivities. While the environment may be unknown, the design itself, that which the designer *can* control, predicts how each relevent parameter will vary based on environmental influences. Sensitivities have been used for dealing with statistical analysis. However, they are perhaps even more useful in dealing with deterministic and systematic effects. The key advantages are that they preserve the use of hierarchy (crucial for productivity), retain the isolation of design and usage, support both worst case and statistical design, and support optimization directly.

Many very practical questions asked by designers relate directly to uncertainties, and could be addressed by sensitivities. What on-chip gradient of any parameter can I have, and have the chip still work? Will my chip work with the possible layer variations I might get? Do I need a clock mesh, or will a tree suffice? How much margin must I leave for a given path to account for possible process variation? What's my parametric yield for this part?

A flow that systematically addresses uncertainty by keeping sensitivities can directly address all these problems. The main disadvantage is the need for format changes and new tool algorithms.

1.1 A motivational example

Here's an example that illustrates why a systematic view of uncertainty is helpful, by showing two cases where the appropriate treatment of uncertainty is very different. The example largely considers timing, but power and yield have similar considerations. Note that the absolute, and relative, importance of each of the uncertainties is constantly changing - OPC, for example, attempts to remove litho variations, and the transition from steppers to scanners will change the optical field distortions.

Here we follow the case of a two small cells through the design process - a standard cell NAND gate, and an SRAM bit cell. In Figure 1(a), we show a simple cell, and in it a transistor. Suppose the transistors have a nominal L of 45 nm. In a 45nm process, however, such small cells will less than 1000 nm on a side, and so some transistors will be at most a few hundred nm from the edge. These transistors will be strongly affected by their optical surroundings. How might the two designers compensate for this uncertainty? The standard cell designer is forced to reply on the final OPC, since the surroundings are unknown. The exact OPC recipe, and the residual errors, are also uncertain, so the designer needs to guardband for this. The post-OPC process window is very uncertain as well, so yield optimization is difficult. Conversely, the SRAM designer may chose to do their own OPC. They may also spec dummy rows and columns in the array, to make the environment more uniform.

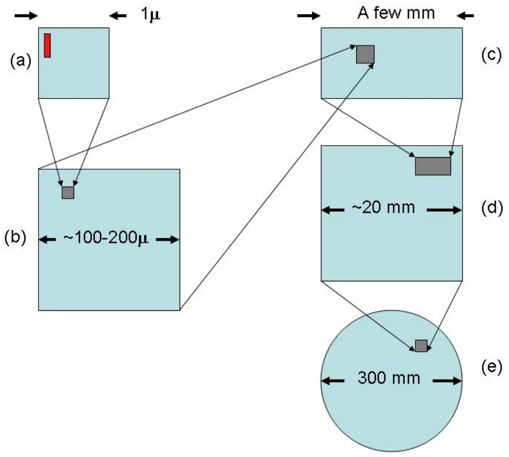


Figure 1. Figure illustrating uncertainties in the design. (a) shows the position of transistor in a cell, (b) and (c) the position of a cell in parent cells, (d) the final position on the chip, and (e) the position of a chip on the wafer.

The etch environment is also uncertain. Since this is not compensated for it must be guardbanded. The two designers may use different guardbands, however. In the SRAM, it is probably not practical to include enough dummy rows and columns to control this completely. The dummy cells will reduce the magnitude of any possible effect, though.

In Fig 1(b) the cell is used in a larger cell. For the SRAM this containing cell is always the same, and no additional uncertainty is introduced. For the NAND gate the containing cell is typically different. Typically, at this level, the litho environment is now known, but CMP environment is not. The radius of this effect is too big for either designer to control completely, and will probably need to be guardbanded. Customers of either cell might like to know how performance and yield is affected by the local density they can achieve, a metric hard to get now.

Then the cells are included in a yet larger cell, Fig 1(c). Now the CMP environment is known. The SRAM designer, as the author of an included block, may also need to watch their metal density. This is not because it will make their own block malfunction, but because it may cause problems for the blocks around it. Here the point is that designers need to worry not only about their own response to uncertainty, but how much uncertainty their designs cause to others who are similarly try to cope.

Next, the cell is included in a chip, as shown in Figure 1(d). At this point the corrections for location within the optical field can be added. It might be possible for the SRAM designer to state in their manual how performance varies as a function of location in the field (since their are only a few hundred SRAMS used in a given design, this could in theory be used manually or by an automatic floorplanner). For NAND gates, which are used by the 10s of thousands, and not assigned by the floorplanner, location dependence would need to be fully automatic, and incorporated into synthesis tools to be of any use.

Finally, each chip is located somewhere on the wafer, as shown in Fig. 1(e). This leads to gradients and other gentle curves across the chip surface. Once again the response of the two designers to uncertainty differs. Since the NAND gate may be used in a clock tree, the designer may be asked to supply a spec of how different two otherwise identical NAND gates might be depending on their placements. For the SRAM designer, the customer is probably not interested in the difference between 2 SRAMS on the same chip - they want both SRAMs to work and meet their spec. Each must work in the presence of gradients and other cross-chip variation.

Though for these two cells the detailed treatment of uncertainty is very different, there are common threads. Designers must:

- Consider what sources of uncertainty are present, and how to deal with them.
- Describe to their users how the uncertainties can be resolved, if indeed they can be. This may involve exporting various cell internals. Exactly what must be in such an 'abstract' depends on the effect being considered and the hierarchical level.
- Describe how their metrics will be affected by this resolution. This might be as simple as guardbanding (the final result will be between X and Y), or as complex as a full simulation model. The rest of this paper argues that sensitivities are a good compromise here.
- Tell their users of design decisions that may affect their (the customer's) uncertainties.

1.2 Structure of this work

Section 1 (this section) is an introduction and motivation. Section 2 looks at previous work. Section 3 discusses how analysis might be done using sensitivities. Section 4 looks at how this could be extended to design, as opposed to analysis. Section 5 looks at some of the practical considerations, and finally section 6 is a summary and conclusion. Appendix A looks specifically at cross-chip variation problems.

Three portions of this paper are novel. First, it looks at flow issues as opposed to tool issues. The suggested treatment of intra-chip variability is new, and finally the discussion of constructive tools (as opposed to analysis) is includes some new suggestions.

2. Previous work

Coping with uncertainties is second only to logical correctness in importance. Many approaches have been tried.

The sensitivities of cells to the local environment has mostly been handled by designing and analyzing the cells in a nominal environment, then adding guardbands to account for variation due to the actual environments. This approach leaves a lot of performance on the table, since (in essence) every cell is assumed to be operating in its worst case environment. This is the approach used for standard cell libraries today.

Designers of circuits such as DRAMs and SRAMs have the advantage the environment is known, since the cell under design is embedded in a large array of identical cells. This community can therefore use tools such as litho simulation to account for the enviroment. The extra effort (per cell designed) is very large, but tolerable since just a few cell designs account for most of the area of the chip. Also, these designs can use strategies (such as dummy cells at the edges of arrays) to make the environment more uniform[18].

Analog (and some digital) designers have used flows where the design is complete, then subject to influences (such as litho sim), then extracted and simulated [16, 12]. This approach has two limitations - first, the design must be complete before this analysis can be performed, and second it generates a flat design as output, since every transistor is potentially unique after this process. Digital designers have tried the same general approach, but by creating location specific timing models[20].

Orshansky, et. al. [10, 11] have looked at variability based upon position in the optical field. They analyze the consequences by altering the circuit parameters, resulting in a flat, transistor level layout which they then simulate at the transistor level.

Quite a bit of work on handling sensitivities has been done in the context of statistical timing, such as [19]. This has been done mainly to address the parametric yield question. While this is important, it is by no means all that we should consider. There are many other practical uses for sensitivities, most of which have little or no statistical components.

3. Analysis

Designers have many possible ways of coping with uncertainties. They may try to try to control the uncertainty (as RAM designers do by including dummy rows and columms), or guardband against it, or change the design to be less sensitive. To make this decision correctly, they must be able to view and manipulate their metrics (typically power, yield, and area) and see how these are affected by the different strategies they might employ. Note that when a cell is designed, only the designer knows how it may be used. Will it be included more than once? Will it be included in multiple orientations? Is the final position in the optical field known, even approximately? It must be possible to specify these uncertainties by hand since no automatic method has the required information.

3.1 Dealing with uncertainties

There are two basic ways of computing the effects of uncertainties - waiting until the relevent data is known, or computing a sensitivity to the missing data, then adjusting the value later.

The first approach involves waiting until the relevent data is known, then redoing the analysis. This is the form of most of the previous work above - if your litho environment is unknown, for example, then wait until it is known, calculate the effect, then re-do your analysis. This is the most accurate method, and can correctly account for complex non-linear effects and interactions.

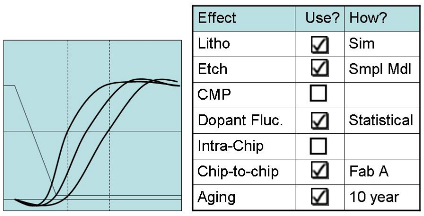
However, after-the-fact analysis is not very helpful during design. Here, sensitivities are more useful, even if they are less accurate. This is because during design, when things can be easily changed, the environment is usually unknown, and often unknowable. And while the environment may be unknown, the design itself, where the designer has control, predicts how each relevent parameter will vary based on environmental influences. Sensitivities have long been used in analog design and statistical analysis. However, they are perhaps even more useful in dealing with deterministic and systematic effects. The key advantages are

- They preserve the use of hierarchy (crucial for productivity)
- Retain the isolation of design and usage.
- Support both worst case and statistical design, and
- Support optimization directly.

It may take some creativity to express some environmental influences as sensitivities. Litho and etching, for example, show very complex environmental dependencies. Even in this case, though, the environment determines the Ls, and then the Ls determine the timing. So information about the possible variation in L can still give the designer very useful feedback about the performance in different environments.

One of the distinctions between a flow and a tool is the flow must consider issues that at least some of the tools are completely unaware of. So if we want designers to consider all sources of uncertainty at all times (even if just to decide to ignore it), then it helps to have a master list of the all relevent variables, their range of variation, their radius of spacial correlation, and so on. Table 1 shows how this might be organized, in a very informal way.

The master table will contain all the possible uncertainties to be considered. The user will normally only want to consider a subset of these, since the effects to be considered will depend on the mitigation techniques used, the eventual use of the cell, and other designer preferences. Furthermore, estimation of each uncertainty will probably come with options as well - fast but approximate methods, detailed simulation, and do on. The user interface must make it clear exactly which uncertainties are included, which are not, and any options related to the computation. Figure 2 show how this might be accomplished from a user interface point of view. In this figure the user is looking at a circuit level timing analysis while modifying the uncertainties included. Obviously, this is only a simple cartoon of how this might be done. Making these ideas work well in a practical flow will require considerable software engineering, in addition to the research required.



Effect	Use?	How?
Litho	☑	Sim
Etch	☑	Smpl Mdl
CMP	□	
Dopant Fluc.	☑	Statistical
Intra-Chip	□	
Chip-to-chip	☑	Fab A
Aging	☑	10 year

Figure 2. Figure illustrating a possible user interface for dealing with uncertainty. The user picks which effects are to be included, and how they should be handled, and sees the results in terms of their preferred metric - in this case timing. There will need to several hierarchical ways of organizing this data, since the user might want to include/exclude whole sets of related effects, or all effects for a given layer, or other subsets.

Next, the computations such as timing must be reasonably fast when the uncertainties are varied. Sensisitivities help in two ways during analysis. In statistical analysis, they allow determination of correlation. Statistical timing is the poster child for this usage. However, sensitivities can help when dealing with completely deterministic effects as well. For example, suppose you have a tool that computes local layer thicknesses as function of the CMP enviromnment. If you have done your extraction with sensitivities, you can immediately combine the two data sets to generate parasitics corrected for CMP variations (a deterministic effect).

A technical distinction that must be considered is the difference between partial and total derivatives. For example, consider the delay of a cell. This would have sensitivities to Vt and L. However, a change in L will also cause a change in Vt. So do we measure the partial derivative (change caused by L alone), or total derivative (change caused by also considering the other effects of a change in L). At the very least this must be carefully specified. The total derivative is more useful to the designer, but the partials can be combined via the chain rule to get the totals. In this particular case it probably makes sense to measure the sensitivities with respect to $Vth0$, the hypothetical threshold of a device with very large L. Then the measured sensitivity to L will include the effect of threshold variation, and the application should not include this, or it will be double counted.

Timing, from a physical designer's perspective, always starts with extraction. Then analog simulation may be used, either directly to measure some timing properties, or to characterize cells. In the digital environment, the characterized cells are combined with the extracted parasitics in delay calculation. Then timing analysis computes the expected circuit performance. The needed modifications to each step are discussed in turn.

3.2 Extraction

Table 1. Master table of uncertainties

Effect	Amount	Range	Distribution	Default Environment
Litho	[-5,+5]nm	1000 nm	N/A	50% Fill
Etch	[-5,+5]nm	10 um	N/A	50% Fill
Loc. in Field	[-4,+6]nm	N/A	Worst Case	$(4 \pm 1, 7 \pm 1)$
M1 thickness	200-300nm	N/A	Gaussian	250 nm
M1 X gradient	[-10,+10]%	N/A	Uniform	0%
...	...	.....	...	

First, extraction should take into account any environmental influences that are known. This implies some new variation-aware options, such as extract-in-context, specification of location on an optical field, and use of an "average" context if no context is known.

For the remaining variation(s), extraction needs to keep track of sensitivities, and write them into its output format. There will be at least 4 parameters for each metal layer. These include, at a minimum, width, thickness, dielectric thickness, and via resistance (spacing does not need to be considered separately since the pitch is extremely well controlled). A few more variables per layer may be required - perhaps one for individual dielectric thicknesses, when multiple dielectrics are used between layers, and perhaps a variable for cladding thickness. For 10 metal layers, this impies at least 40 variables, and maybe as many as 60. At least this problem is close to linear over the range of anticipated values[14]. However, the correct model for via variability is very unclear.

Extractors can already deal with changes in width, as part of their task to extract the parameters for lines of different (deliberate) widths. Therefore derivatives with respect to W require no new characterization. Thickness of metal layers affect both capacitance and resistance. The R changes are relatively easy to predict as a function of thickness and width only, since they do not interact with other parameters. Capacitance is harder and will require characterization changes with both metal thickness and dielectric thicknesses. Modern processes may require another variable as well, the thickness of a conformal or other partial dielectric layer[9].

Extractors often contain 'ad hoc' reducers that compress the output netlist. These will need to be modified, as well. Perhaps a C++ overloaded library would help, following the lines of [8].

Note that the extractor will presumably be slower, and generate bigger output files, if complete variation information is requested. However, the overall flow efficiency may well be improved, since there will be no need for corners. Even if traditional extraction corners are desired, these can be derived by plugging in extremal variations in layer properties.

From the point of view of an extractor, components are point sources. They need to specify the root variable, and the XY, but are not concerned with correlation (except maybe in the reducer, and a practical solution might be to not reduce parasitics that are far apart physically).

The output of extractors are written in ASCII formats such as SPEF, proprietary formats, and databases such as OpenAccess. Each of these must be modified to support sensitivities. Companies [4] are working on extensions to SPEF that support sensitivities.

3.2.1 Hierarchical Extraction with Sensitivities

If models of optical or CMP effects are available, they can be employed in at least two ways. First, the geometry itself can be modified to reflect local conditions before extraction. This is most accurate but may make the use of hierarchy difficult. Alternatively, extraction can incorporate sensitivities, and then the environment effects can be used to modify the extracted values. In this case the extractor must compute the correct environment for the child cells. If a child cell has sensitivities based (for example) on layer density, the extractor must promote the appropriate cell info (perhaps a layer density map for the child), perform the higher level density calculation, and then write out the result as the local density for the child instance.

It may also fall to the extractor to build an abstract version of the cell that can be used for calculating sensitivities. (Conceptually, this is just the same as computing an input C for a cell to be used in timing analysis.) The information that needs to be included depends on the effect. In the case of optical and litho effects, this might be the polygons near the edge and a set of measurement points. In the case of a CMP model, it might be a density map of the cell.

3.3 Analog simulation and cell characterization

After extraction, the user will want to see the impact of possible variations of the uncertain parameters. Analog simulation is the most accurate tool for this, and may be used in two ways - directly, by an analog designer or a digital designer wishing to examine a particular path, or to characterize cells.

As a simple example, imagine you have a NAND gate with 4 transistors, then with a given load and input slope you can simulate to get a delay D and output slope S, using a conventional simulator. But the L of the 4 transistors are really uncertain, so you also need

$$\frac{\partial D}{\partial L_1}, \frac{\partial D}{\partial Vt_1}, \frac{\partial S}{\partial L_1}, \frac{\partial S}{\partial Vt_1}, \quad \frac{\partial D}{\partial L_2}, \frac{\partial D}{\partial Vt_2}, \frac{\partial S}{\partial L_2}, \frac{\partial S}{\partial Vt_2}, \quad \text{etc.}$$

Normally, this would take N simulations, where N is the number of underlying uncertainties you want sensitivities for. (Here N is 8, and you might vary each of the 8 underlying variables and re-simulate.) SPICE and SPICE-like simulators can compute sensitivities in much less than N times as long for certain analysis types (AC, DC, and RF). Extending this to transient analysis is not obvious, but may well be possible[13].

If the user is trying to do cell characterization, at least 3 problems come to mind. First, the sensitivities must be computed as above. Next, they must be written out so someone else can use them. This requires a new delay format is needed, since none of the stock formats (.lib, ECSM, etc.) can handle this now. Finally, enough data about the cell must be written so the parent can compute appropriate instance specific values. This is covered in the extraction section above.

These all have analogs in the existing characterization flows - the current sensitivities (to load C and input slope) are represented as tables or equations. There are some standard formats that can hold this data (.lib for timing, LEF for physical data). And some information is promoted to the parent for parameter calculation - for example, the lumped input C is stored in the format, so the parent can compute the input slope, and then use this as a parameter to the cell.

3.4 Delay Calculation

Delay calculation with sensitivities has been examined by a number of authors[2, 5, 7, 8], with several possible solutions. The biggest remaining question is on-chip correlation, as paths, and even single nets, often extend over large portions of a chip. See Appendix A for a more detailed discussion of some possible approaches.

3.5 Timing Analysis

Statistical timing is an area where sensitivities are already used (See [19], and the references therein, for examples). When compared to propogating PDFs, they have the advantage that correlation, caused by either circuit topology or geographic proximity, can be handled easily.

However, sensitivities can also be very useful for deterministic effects. Take, for example, analysis with temperature variation. During operation, the temperature across the die will not be uniform, and the designer wants to know if the circuit will still operate properly. Treating these statistically seems wrong, since they are not random, and the patterns of correlation will vary depending on the chip's operating modes. Doing a corner case analysis is even worse, since it is scarcely likely that one transistor on the chip is at -55° C while another is at $+125^\circ$ C, even though these are individually possible. Instead, if the designer knows how the delay of each path varies with temperature, they can apply models of many different degrees of sophistication to this problem.

Sensitivity based analysis can help reduce the additional work needed when different analyses, under different conditions, must be run (though the goal of statistical analysis is to avoid this sort of corner analysis, sometimes it is unavoidable). Suppose, for example, that a product has to work at 3 different fabs. If the distributions are different, then the extraction and delay calculation can be re-used, though the timing analysis must be re-done (since MAX and MIN may give different results). If the correlations are different, then extraction can be re-used but both delay calculation and timing analysis must be re-done.

4. Variation aware constructive tools

Of course, once analysis is available, the next question from users will be "Why don't you optimize for this?" Here are some possibilities:

4.1 Placement

There are systematic variations of parameters such as ΔL across the optical field of steppers and scanners. This makes the cell performance a strong function of location - see [10] and [11] for a much fuller discussion of this effect. A placer could optimize for this. The change from steppers to scanners may make this effect much less important[15].

4.2 Clock trees

Clock tree design could benefit in several ways from sensitivities. Analysis will be the first step, where sensitivities allow a detailed slack analysis for each clocked element pair connected by logic. The next step is to optimize the clock networks against variation.

(a) balance trees with respect to layer sensitivities. Designers of high performance circuits have done this for years, using equal amounts of M1, M2, and so on on each branch. With sensitivities, a program can do this automatically and more accurately. A program, for example, can account for the differing delay amounts induced by the same amount of metal, depending on the distance from the root. The manual schemes today just try to equalize the raw amounts.

(b) Smoothly convert between trees, trees with links, and meshes. With a sensitivity driven flow, a clock tree tool can check the sensitivities between each pair of connected flip-flops for a tree, a mesh, and anything in between. A hypothetical algorithm, for example, could add links one at a time between nodes with excessive sensitivities. By this means it could compute a Pareto-like curve that goes between a minimal tree (with maximum sensitivity) and a full mesh (minimal sensitivity but maximum expense). Intuitively, a tree like structure with just a few critical links might do almost as well as a full mesh, for much lower cost.

(c) A tool could present the user with a plot of needed slacks vs. on chip gradients. Then the user can see the tradeoff between allowance for larger skews, and performance.

4.3 Signal Routing

Signal routing can look at the sensitivities when doing layer assignment. It might well make sense, for example, to only route nets with adequate slack on layers with high variability. Conversely, routing nets with low slacks in ways that have low sensitivity (either from layer properties, or by making the routing contribution small), will help the overall yield. Also, the ‘more duplicate vias’ vs ‘more wiring’ tradeoff can be examined in detail.

4.4 Testing

Delay fault testing could target those nets that have high sensitivities. Production test could look at failures, look at the sensitivities of the failing paths, and speculate upon the fabrication problems that might have lead to them.

4.5 Chip final timing

Tools might substitute in cell variants at the very end, when all the environments and slacks are known. This is in practice what happens now with OPC (each instance become unique) but a wider variety of effects could be addressed - etching effects and position in the optical field are examples.

5. Conclusions

Building a flow with explicit consideration of what is known, and what is unknown, offers many advantages. It seems likely this should be based on sensitivities, which allow hierarchical design and can account for both deterministic and statistical effects, and lead naturally to optimization. However, many practical problems remain, as new format and new tools would be required.

Appendix A: Modelling on-chip variation

How should on-chip variation be modelled? This is a key question. It will also determine when variables can be combined during network reduction and other operations, and for data compression will determine the size of the numbers needed for representing coordinates. This will also determine the number of statistical variables per underlying physical variable.

The correlation between spatially separate values of the same parameter (such as metal thickness) could be done in many ways.

- A variable per location could be used, with a correlation vs distance specified directly. This is a direct and accurate model, but mathematically cumbersome because of the many (millions) of partially correlated variables.

- A hierarchical structure of independent variables can be used, where the coefficients of the larger squares provide the correlation [1]. This also uses many variables, but at least they are independent. It should probably be enhanced with overlapping squares, as proposed by Sylvester[17].

- A set of basis functions that describe the on-chip variation can be specified. This can be mathematically complete (such as fourier or wavelet transforms), and so can describe any possible on-chip variation. For example, replacing the hierarchical structure of [1] with a 2D wavelet transform might make the coefficients more intutitive (first is variation of the mean, second is the difference between halves of the chip, and so on). Or perhaps a power spectrum of spatial variations could be used. This has not been tried, to the author’s knowledge.

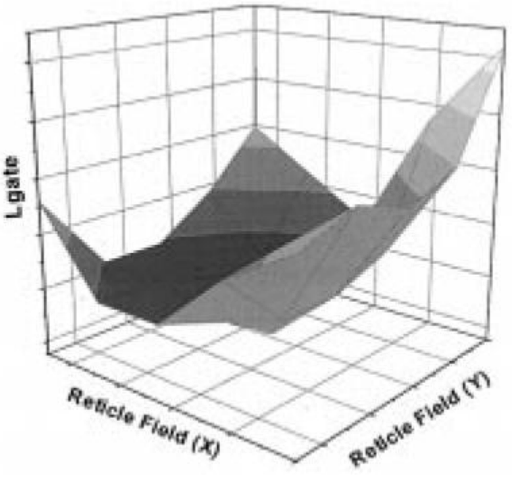


Figure 3. Figure illustrating smooth variation across the field of a stepper.

- A set of basis function based on expected mechanisms can be used. For example, we might expect gradients (due to chip position on the wafer, see [6]), and perhaps focus terms that look like $O(x^2)$. A small set of basis functions cannot support all possible on-chip variations, but model the ones they do support very efficiently. This is expanded below.

5.1 Polynomial bases

Variations of parameters across a wafer would be expected to look quite smooth across a chip. This is true on both a theoretical (what would cause uncorrelated random variations?) and experimental grounds[3], where it is found that once all deterministic variations are removed, what is left has very little correlation. See also [6], which looks a linear cross-chip gradient model. Therefore it might make sense to expand variations in a polynomial basis, which also corresponds to the way a designer might think. For example, the overall thickness of a layer might vary by ±30% from chip to chip, but the variation across a chip might be at most 10%, and close to linear. The next term describes the maximum departure from linearity, and so on. This has a number of advantages

- Few coefficients - 1 for value, 3 for gradients, 6 if $O(x^2)$ is included, 10 if $O(x^3)$.

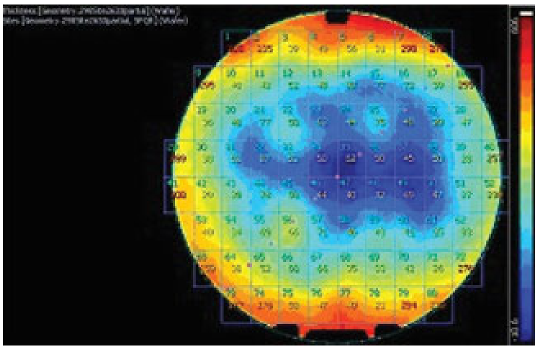


Figure 4. Figure illustrating smooth variation across a wafer. No die has more than a single minimum or maximum. From http://www.micromagazine.com/archive/03/10/maleville.html

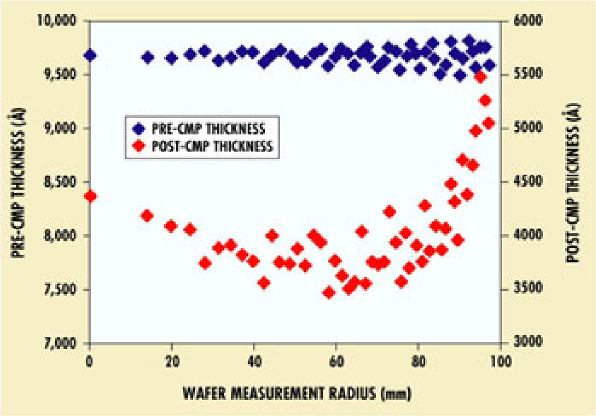


Figure 5. Figure illustrating smooth variation across a wafer as a function of radius. Note the correlation between values and gradients - when the value is low (at 60 mm) the gradient would be expected to be small. From http://www.micromagazine.com/archive/02/01/Lawing.html

- Sensible correlation behavior. There are no edges in the corrleation function, unlike the proposal.Items close to each other will always be highly correlated. An $O(x^2)$ behavior can cause items far apart to be more highly correlated than items at smaller distances[3]. This method can capture that behavior.

- Higher order correlation. For example, in figure 5, we can see that when the metal is thinest, it will also have very small gradients.

6. REFERENCES

[1] A. Agarwal, D. Blaauw, and V. Zolotov. Statistical timing analysis for intra-die process variations with spatial correlations. In *ICCAD '03: Proceedings of the 2003 IEEE/ACM international conference on Computer-aided design*, page 900, Washington, DC, USA, 2003. IEEE Computer Society.

[2] K. Agarwal, D. Sylvester, D. Blaauw, F. Liu, S. Nassif, and S. Vrudhula. Variational delay metrics for interconnect timing analysis. In *DAC '04: Proceedings of the 41st annual conference on Design automation*, pages 381–384, New York, NY, USA, 2004. ACM Press.

[3] P. Friedberg, Y. Cao, J. Cain, R. Wang, J. Rabaey, and C. Spanos. Modeling within-die spatial correlation effects for process-design co-optimization. In *Sixth International Symposium on Quality of Electronic Design (ISQED'05)*, pages 516–521, 2005.

[4] V. Gerousis, 2005. Personal Communication.

[5] P. Li, F. Liu, X. Li, L. T. Pileggi, and S. R. Nassif. Modeling interconnect variability using efficient parametric model order reduction. In *DATE '05: Proceedings of the conference on Design, Automation and Test in Europe*, pages 958–963, Washington, DC, USA, 2005. IEEE Computer Society.

[6] Y. Liu, S. R. Nassif, L. T. Pileggi, and A. J. Strojwas. Impact of interconnect variations on the clock skew of a gigahertz microprocessor. In *DAC '00: Proceedings of the 37th conference on Design automation*, pages 168–171, New York, NY, USA, 2000. ACM Press.

[7] Y. Liu, L. T. Pileggi, and A. J. Strojwas. Model order-reduction of rc(l) interconnect including variational analysis. In *DAC '99: Proceedings of the 36th ACM/IEEE conference on Design automation*, pages 201–206, New York, NY, USA, 1999. ACM Press.

[8] J. D. Ma and R. A. Rutenbar. Fast interval-valued statistical interconnect modeling and reduction. In *ISPD '05: Proceedings of the 2005 international symposium on physical design*, pages 159–166, New York, NY, USA, 2005. ACM Press.

[9] N. S. Nagaraj, 2005. DAC 2005 DFM Tutorial.

[10] M. Orshansky, L. Milnor, P. Chen, K. Keutzer, and C. Hu. Impact of spatial intrachip gate length variability on the performance of high-speed digital circuits. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 21(5):544–553, May 2002.

[11] M. Orshansky, L. Milnor, and C. Hu. Characterization of spatial intrafield gate CD variability, its impact on circuit performance, and spatial mask-level correction. *Semiconductor Manufacturing, IEEE Transactions on*, 17(1):2–11, Feb 2004.

[12] M. Orshansky, L. Milor, P. Chen, K. Keutzer, and C. Hu. Impact of spatial intrachip gate length variability on the performance of high-speed digital circuits. *IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems*, pages 544–553, 2002.

[13] J. Phillips, 2005. Personal Communication.

[14] L. Scheffer. Explicit computation of performance as a function of process variation. In *TAU '02: Proceedings of the 8th ACM/IEEE international workshop on Timing issues in the specification and synthesis of digital systems*, pages 1–8, New York, NY, USA, 2002. ACM Press.

[15] W. Staud, 2005. Personal Communication.

[16] B. Stine, D. Boning, J. Chung, D. Ciplickas, , and J. Kibarian. Simulating the impact of poly-cd wafer-level and die-level variation on circuit performance. In *Proc. Second International Workshop on Statistical Metrology*, pages 24–27, 1997.

[17] D. Sylvester. Personal Communication.

[18] R. Venkatraman, R. Castagnetti, . Kobozeva, F. Duan, A. Kamath, S. Sabbagh, M. Vilchis-Cruz, J. Liaw, J.-C. You, and S. Ramesh. The design, analysis, and development of highly manufacturable 6-t sram bitcells for soc applications. *Electron Devices, IEEE Transactions on*, 52:218 – 226, Feb 2005.

[19] C. Visweswariah, K. Ravindran, K. Kalafala, S. G. Walker, and S. Narayan. First-order incremental block-based statistical timing analysis. In *DAC '04: Proceedings of the 41st annual conference on Design automation*, pages 331–336, New York, NY, USA, 2004. ACM Press.

[20] J. Yang, L. Capodieci, and D. Sylvester. Advanced timing analysis based on post-opc extraction of critical dimensions. In *DAC '05: Proceedings of the 42nd annual conference on Design automation*, pages 359–364, New York, NY, USA, 2005. ACM Press.