PARADE: PARAmetric Delay Evaluation Under Process Variation*

Xiang Lu†, Zhuo Li†, Wangqi Qiu‡, D. M. H. Walker‡, Weiping Shi†

†*Dept. of Electrical Engineering*
Texas A&M University
College Station, TX 77843-3124
wshi@ee.tamu.edu

‡*Dept. of Computer Science*
Texas A&M University
College Station, TX 77843-3112
walker@cs.tamu.edu

Abstract

Under manufacturing process variation, the circuit delay varies with process parameters. For delay test and timing verification under process variation, it is necessary to model the variational delay as a function of process variables. However, conventional methods to generate such functions are either slow or inaccurate. In this paper, we present a number of new methods for fast parametric delay evaluation under process variation. Our methods are either based on explicit delay formulae or based on characterized lookup tables, and are significantly faster than conventional methods of comparable accuracy. Due to the efficiency of our method, we can accurately model any path delay as a function of multiple interconnect and device process variables in large circuits. Experimental results on ISCAS85 circuits show that the path delay error predicted by our methods is about 1% of that computed by the RSM using SPICE, where the path delay variation is within ±10%.

1. Introduction

With the shrinking feature size in VLSI technology, the impact of process variation is increasingly felt. To address the effect, great amount of research has been done recently, such as the clock skew analysis under process variation [1, 2, 3], statistical performance analysis [4, 5, 6], worst case performance analysis [7, 8], parametric yield estimation [10], impact analysis on micro architecture [10] and delay fault test under process variation [11, 12, 13, 14].

In all the above research, one important task is to compute variational path delay under process variation, either as functions of process variables [1, 2, 4, 8, 9, 14] or as random variables of certain distribution [3, 6, 12, 15]. However, the conventional methods to compute path delay are either slow or inaccurate. The response surface method (RSM), which performs multiple simulations and curve-fittings, is used in [4, 8, 9, 15]. To achieve high accuracy, the RSM method must perform multiple parasitic extractions under different process conditions. Due to the large number of metal layers in the modern technology, there are many interconnect process variables. For example, for a k-layer technology, there are 3k process variables, corresponding to the metal width, metal thickness and inter-layer dialectic thickness of each layer. As a result, the traditional RSM becomes prohibitive for large circuits. Orshansky *et al.* [7] derived delay sensitivity to gate length variation based on a simple model, and expressed delay as a function of gate length. Their method does not automatically apply to interconnect process variation due to the lack of a similar model for the interconnect. For statistical timing analysis, it is also necessary to compute the path delay under different process conditions [6, 12]. The previous methods simply perform multiple delay evaluations, which is obviously very time consuming.

In this paper, we present a new method PARADE for fast parametric delay evaluation using analytical formulae and pre-characterized lookup tables. The variational path delays are modeled as linear functions of process variables, and computed efficiently. No multiple parasitic extractions and multiple delay evaluations are needed, resulting in a significant speedup over the traditional RSM. Instead, we analyze a small sample of nets to compute the capacitance sensitivity for all process variations and use an efficient method to evaluate delay variation. The efficiency of our method makes it possible to comprehensively analyze circuit performance on all interconnect and device process variables for large circuits. Experiments on ISCAS85 circuits show that our methods achieve high accuracy and efficiency. Compared to the traditional RSM, the delay error is within ±7% using analytical methods, and is within ±1% using the table lookup method.

The paper is organized as follows. In Section 2, we present the analytical method and the table lookup method. In Section 3, we compare the performance of the new methods with RSM. The conclusion is given in Section 4.

*This research was supported in part by the SRC grant 000-TJ-844, NSF grants CCR-0098329, CCR-0113668, EIA-0223785, and ATP grant 512-0266-2001.

0-7695-2093-6/04 $20.00 © 2004 IEEE

IEEE COMPUTER SOCIETY

2. Parametric Delay Evaluation

In order to calculate the path delay under process variation, we first compute the *buffer-to-buffer* delay. The buffer-to-buffer delay, or net delay, is defined as the delay from the input pin of a cell to the input pin of a downstream cell. After all buffer-to-buffer delays in the circuit are computed, the delay of any path can be easily obtained by adding up buffer-to-buffer delays along the path.

We approximate the buffer-to-buffer delay as a linear function of process variables:

$$d(\mathbf{x}, s) \approx d_0(s) + b_1 x_1 + b_2 x_2 + \cdots + b_p x_p , \qquad (1)$$

where $d_0(s)$ is the nominal delay, $\mathbf{x}=(x_1, x_2, \ldots, x_p)$ is the vector of process variables, each representing the deviation percentage from the nominal value, s is the input signal slope (slew time), and $b_i=\partial d/\partial x_i$ is the delay sensitivity to process variable x_i.

There are many forms of process variation, see for example Stine *et al*. [16] and Nassif [17]. In this paper, we consider the systematic process variation, such as the variation on gate length, and the variation of metal width, metal thickness, and inter-layer-dielectric (ILD) thickness related to each interconnect layer. Our methods can be extended to include other process variation such as the threshold voltage, the supply voltage and the temperature, as long as the approximated delay can be expressed as a linear function of the process variables within their variation ranges.

The effect of signal slope has been studied in previous research, for example, in static timing analysis [18] and in variational delay evaluation [17]. In this paper, we consider its effect in computing the nominal delay d_0. At the same time, the slope of the signal at the input pin of the downstream cell is computed for the next buffer-to-buffer delay evaluation on the path. The computation of nominal delay and signal slope can be done by any commercial tool, and is not the focus of this paper. The key issue is to efficiently compute delay sensitivities b_1, b_2, …, b_p.

2.1. Analytical Method

There are many models for buffer-to-buffer delay calculation, such as lumped C, Elmore, D2M [19], and effective capacitance [20]. In these methods, the delay $d = d(R, C)$ is a function of parasitic RCs, though d may not be a closed form expression of R and C. Nevertheless, the delay sensitivity can be defined as

$$\frac{\partial d}{\partial x_i} = \frac{\partial d}{\partial R} \cdot \frac{\partial R}{\partial x_i} + \frac{\partial d}{\partial C} \cdot \frac{\partial C}{\partial x_i} . \qquad (2)$$

Our analytical method is based on the lumped C model. In Fig. 1, we show the Thevenin equivalent circuit of a buffer-to-buffer segment, where the driving cell is represented by a voltage source V_s and a driving resistance R_d, the interconnect and the downstream cell are represented by a simple lumped C_L.

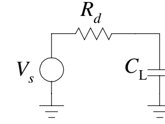

Figure 1. A buffer-to-buffer segment represented by a lumped C model.

The delay function of the lump C model is as follows:
Lumped C: $d = R_d C_L$.

The delay sensitivities with respect to R_d, and C_L can be derived as follows:

Lumped C: $\frac{\partial d}{\partial R_d} = C_L$, and $\frac{\partial d}{\partial C_L} = R_d$.

To complete the calculation of delay sensitivity, we need to compute $\partial R_d/\partial x_i$ and $\partial C_L/\partial x_i$.

The value of R_d varies with the input signal slope and output load, and can be pre-computed by simulation. Sensitivities of R_d to device parameters, such as the gate length and the threshold voltage, can be pre-determined by RSM and stored in a 2-dimensional table.

To make our method widely applicable to different design flows, the computation of $\partial C_L/\partial x_i$ must be independent of any particular parasitic extraction tool. However, it is more difficult to compute $\partial C_L/\partial x_i$. This is because the parasitic capacitance of a metal wire depends not only on the wire itself, but also on the neighboring condition. Traditional formula based methods are no longer used and are replaced by more accurate 2.5D/3D tools. For these tools, there is no explicitly capacitance formula we can use. To get $\partial C_L/\partial x_i$ for any process variable x_j efficiently and accurately under any complex neighboring condition, we define the concept of *unit capacitance sensitivity* sc_i, which is equal to the average of $\partial C_j/\partial x_i/C_j$ on n sample parasitic capacitances C_j

$$sc_i = \frac{1}{n} \sum_j \frac{\partial C_j / \partial x_i}{C_j} , \qquad (3)$$

For a given process technology, the parasitic capacitances are randomly selected. Therefore, the unit capacitance sensitivity reflects the average parasitic capacitance sensitivity under different neighboring environments of interconnect. In Fig. 2 we show capacitance sensitivity due to the wire width variation on metal 2 in ISCAS85 circuit c432 on 406 sample nets. The circuit layout generation and parasitic extraction is done by Cadence Silicon Ensemble™ in TSMC 180nm 1.8V 5-metal layer technology. From the figure, we can see that for most nets, the value of $(\partial C_j/\partial x_i)/C_j$ is about 0.61. The same process is repeated for every process variable and the corresponding sc_i is computed. Note that, since the sampled nets are

0-7695-2093-6/04 $20.00 © 2004 IEEE

IEEE COMPUTER SOCIETY

randomly chosen and represent the typical neighboring environment, the path delay, which is the sum of buffer-to-buffer delays, will tend to give the average path delay.

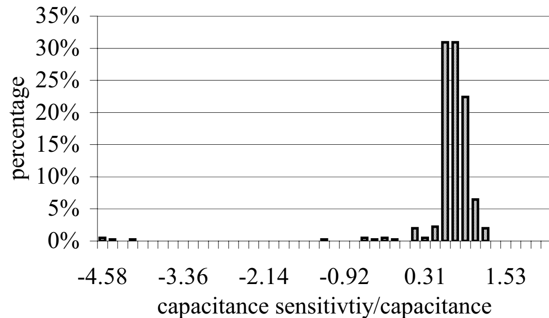

Figure 2. The results of capacitance sensitivity/capacitance with one process variation (metal 2 width) for ISCAS85 c432 on 406 samples.

In our later experiments, we choose $n=200$ and through a large number of experiments this number can give fast yet accurate estimation. After we get sc_i, for each buffer-to-buffer delay, the capacitance sensitivity $\partial C_L/\partial x_i = sc_i \cdot C_L$, where C_L is the lumped capacitance of interconnect.

2.2. Table Lookup Method

Now we present a more accurate method for delay sensitivity computation based on characterized lookup tables.

For each cell, a two-dimensional delay sensitivity table is built, where load capacitance C_L and capacitance change ΔC_L due to certain process variation are variables. Each entry of the table is a delay change Δd due to a capacitance change ΔC under the ramp input with fixed slew time (50 ps in our experiments) and load capacitance C_L. For each buffer-to-buffer segment, effective capacitance rather than lumped total capacitance is used to refer the table in order to more accurately model the interconnect resistance shielding effect. There are several effective capacitance method can be used, such as iterative method [20] and noniterative method [22]. For the speed concern, we use noniterative method here [22]. The interconnect Π model is first computed based on matching the first 3 moment of driving admittance. Then

$$C_{eff} = C_1 + C_2(1 - e^{-T/(RC_2)}), \qquad (4)$$

where C_1 is the capacitance near the driver, C_2 is the one far from the driver, R is equivalent interconnect resistance and T is the Elmore delay. For the accuracy concern, more accurate models such as [20][21] can be used here. Capacitance change ΔC_L can be derived based on process variation range and our pre-computed unit capacitance sensitivity.

For gate length variation, since it does not change the interconnect parasitic, we only need a 1-dimensional sensitivity table to get the delay sensitivity over gate length, in which the variable is load capacitance C_L, and each entry is the sensitivity $\partial d/\partial l_g$.

The whole procedure to evaluate the variational path delay for the given path is shown as following:

1. For each buffer-to-buffer delay, derive equivalent Π model. Compute the effective capacitance C_{eff} based on Π model and Eqn. (4).
2. Given Δx_i, compute $\Delta C_{eff} \approx sc_i \cdot C_{eff} \cdot \Delta x_i$.
3. For gate length variation, use C_{eff} to search for the corresponding $\partial d/\partial l_g$. Then $\Delta d=\partial d/\partial l_g \cdot \Delta l_g$. For other process variation variables, use ΔC_{eff} and C_{eff} to search for the corresponding Δd. For all tables, if the value is not at the entry point, linear interpolation is used.
4. Sum Δd at all buffer-to-buffer delays at the given path and get the variational path delay

The construction cost for the table is dependent on the number of delay evaluations and parasitic extractions. For gate length variation, suppose we need to sample the downstream capacitance by the number of r, and the number of cells to be pre-characterized is m, we need to perform $2 \cdot m \cdot r$ delay evaluations. For other process variables, we also need to sample the capacitance change by the number of t, and then the total number of delay evaluations is $2 \cdot m \cdot r \cdot t$. As shown in previous section, the cost of computing unit capacitance sensitivity is also small that needs a few hundred of small nets parasitic extractions. Therefore, the total cost of table built-up method is much smaller than traditional RSM methods, which needs to perform whole circuit parasitic extractions and delay evaluations $p+1$ times, where p is the number of process variations.

3. Experiment Results

We apply our methods to ISCAS85 circuits using a UNIX server running on SunOs 5.7. The circuit layout generation and parasitic extraction is done by Cadence Silicon Ensemble™ in TSMC 180nm 1.8V 5-metal layer technology. The systematic process variation variables considered in our paper are variations of the transistor gate length, the width of 5 metal layers, the thickness of 5 metal layers and the thickness of 5 inter-layer-dielectrics (ILD). We apply the following manufacturing ranges of these variables: gate length ±6%, metal width ±5%, metal thickness ±20%, and ILD thickness ±40%. The range of delay variation is about ±10% of the nominal delay.

0-7695-2093-6/04 $20.00 © 2004 IEEE

The cell library using in our experiments consists of 27 cells. In the computation of delay sensitivity table, we sample the sink capacitance by r = 24, sample the capacitance change by t = 9. The total number of delay evaluations for the table construction is 11164, and costs about 1 hour using SPICE simulation. The time on computing unit capacitance sensitivity on 200 sample nets is 6.14 seconds. The input slew time for each buffer-to-buffer segment is fixed as 50 ps.

We first show the running time comparison between the traditional RSM and new method in Table 1. For each circuit we perform RSM and our new method respectively to generate the parasitic delay model for all buffer-to-buffer segments in the circuit. RSM is implemented by SPICE simulation with its running time listed in the third column. Note that, we run SPICE simulations for each buffer-to-buffer delay with fixed slope. The path delay is computed by summing buffer-to-buffer delays. The running time of our method is listed in followed columns. Compared to RSM, our method achieves significant speedup. The running time of the method using lumped C delay model is faster than the lookup table method by 2-5 times.

To evaluate the accuracy of our method, we perform RSM and our method on the longest path of each circuit. Results are compared under the corner condition. In our experiments, the path delay under the nominal process condition d_0 is computed by SPICE simulation. Under the corner condition, the parametric variational delay computed by the traditional RSM is denoted as d' and the parametric variational delay calculated by our method is denoted as d'' using Eqn. (1). Then the delay error under the corner condition is computed by $(d''-d')/(d_0+d')$. This value indicates the result of our method is how close to the result of RSM.

The results are shown in Table 2, where the number of cells in the longest path is listed in the second column, the path delay computed by RSM is listed in the third column and the delay variation under the corner condition is listed in the fourth column. From the table, we can conclude that the table lookup method is more accurate. Its delay error is around 1% of the path delay, where the delay error of lumped C model is less than 7%. The maximum error of the variational delay $(d''-d')/d'$ of the table method is about 10%.

4. Conclusions

In this paper, we present fast parametric delay evaluation under process variation by PARADE with analytical formulas and lookup tables. Our method avoids multiple parasitic extractions and multiple delay evaluations as did in the traditional RSM, and result in significant speedup. Table lookup method achieves high accuracy. Experiments on ISCAS85 circuits show that our methods are effective and accurate for the parametric delay evaluation under process variation. We are working to include slope effect into table lookup methods.

Table 1. Running time comparison between the traditional RSM and new methods for ISCAS85 circuits.

Circuit	# of buffer-to-buffer delays	Running time		
		RSM (hh:mm)	New Methods (s)	
			Lumped C	Table
c432	343	0:41	0.014	0.020
c499	440	1:03	0.017	0.026
c880	755	1:30	0.014	0.053
c1355	1096	2:13	0.044	0.084
c1908	1523	2:48	0.075	0.304
c2670	2292	4:19	0.108	0.456
c3540	2961	5:39	0.143	0.466
c5315	4509	>8 hr	0.196	0.785
c6288	4832	>9 hr	0.200	0.846
c7552	6253	>10 hr	0.308	1.600

Table 2. Accuracy comparison between the traditional RSM and new methods for ISCAS85 circuits.

Circuit	# of cells in path	Delay (ps)	Delay Var. (%)	Delay error under corner condition (%)	
				Lumped C	Table
c432	17	507.9	9.28	-4.21	-0.97
c499	11	447.4	8.98	-4.87	-0.53
c880	24	669.1	8.19	-6.94	-1.02
c1355	24	614.9	9.05	-5.00	-1.14
c1908	40	826.5	8.39	-5.06	-0.87
c2670	32	1103.2	8.47	-5.06	-0.87
c3540	47	1189.8	7.90	-5.33	-0.71
c5315	49	1124.2	8.36	-4.96	-0.81
c6288	124	1788.7	8.39	-6.39	-0.71
c7552	41	834.2	8.33	-4.61	-0.70

5. Acknowledgments

The authors thank Dr. Sani R. Nassif of IBM for important suggestions.

6. References

[1] Y. Liu, S. R. Nassif, L. T. Pileggi and A. J. Strojwas, "Impact of interconnect variations on the clock skew of a gigahertz microprocessor," *DAC* 2000, pp. 168–171.

[2] V. Mehrotra, S. L. Sam, D. Boning, A. Chandrakasan, R. Vallishayee and S. Nassif, "A methodology for modeling the effects of systematic within-die interconnect and device variation on circuit performance," *DAC* 2000, pp. 172–175.

0-7695-2093-6/04 $20.00 © 2004 IEEE

IEEE COMPUTER SOCIETY

[3] E. Malavasi, S. Zanella, C. Min J. Uschersohn, M. Misheloff and C. Guardiani, “Impact analysis of process variability on clock skew,” *ISQED* 2002, pp. 129–132.
[4] R. B. Brawhear, N. Menezes, C. Oh, L. T. Pillage and M. R. Mercer, “Predicting circuit performance using circuit-level statistical timing analysis,” *DATE* 1994, pp. 332–337.
[5] H. Chang and S. S. Sapatnekar, “Statistical timing analysis considering spatial correlations using a single PERT-like traversal,” *ICCAD* 2003, pp. 621–625.
[6] A. Agarwal, D. Blaauw and V. Zolotov, “Statistical timing analysis for intra-die process variations with spatial correlations,” *ICCAD* 2003, pp. 271–276.
[7] M. Orshansky, L. Milor, P. Chen, K. Keutzer and C. Hu, “Impact of systematic spatial intra-chip gate length variability on performance of high-speed digital circuits,” *ICCAD* 2000, pp. 62–67.
[8] E. Acar, S. N. Nassif, L. Ying and L. T. Pileggi, “Assessment of true worst case circuit performance under interconnect parameter variations,” *ISQED* 2001, pp. 431–436.
[9] A. Gattiker, S. Nassif, R. Dinakar and C. Long, “Timing yield estimation from static timing analysis,” *ISQED* 2001, pp. 437–442.
[10] S. Borkar, T. Kamik, S. Narendra, J. Tschanz, A. Keshavarzi and V. De, “Parameter variations and impact on circuits and microarchitecture,” *DAC* 2003, pp. 338–342.
[11] G. M. Luong and D. M. H. Walker, “Test generation for global delay faults,” *ITC* 1996, pp. 433–442.
[12] J. J. Liou, A. Krstic, L. C. Wang and K. T. Cheng, “False-path-aware statistical timing analysis and efficient path selection for delay testing and timing validation,” *DAC* 2002, pp. 566–569.
[13] A. Krstic, L. C. Wang, K. T. Cheng and J. J. Liou, “Diagnosis of delay defects using statistical timing models,” *VTS* 2003, pp. 339–344.
[14] X. Lu, Z. Li, W. Qiu, D. M. H. Walker and W. Shi, “Longest path selection for delay test under process variation,” *ASP-DAC* 2004.
[15] A. D. Fabbro, B. Franzini, L. Croce and C. Guardiani, “An assigned probability technique to derive realistic worst-case timing models of digital standard cells,” *DAC* 1995, pp. 702–706.
[16] B. Stine, D. Boning and J. Chung, “Analysis and decomposition of spatial variation in integrated circuit process and devices,” *IEEE Trans. on Semiconductor Manufacturing*, 10(1), 1997, pp. 24–41.
[17] S. R. Nassif, “Modeling and analysis of manufacturing variations,” *CICC* 2001, pp. 223–228.
[18] D. Blaauw, V. Zolotov and S. Sundareswaran, “Slope propagation in static timing analysis,” *IEEE Trans. CAD*, 21(10), 2002, pp. 1180–1195.
[19] C. J. Alpert, A. Devgan and C. V. Kashyap, “RC delay metrics for performance optimization,” *IEEE Trans. CAD*, 20(5), 2001, pp. 571–582.
[20] J. Qian, S. Pullela and L. Pillage, “Modeling the “Effective capacitance” for the RC interconnect of CMOS gates,” *IEEE Trans. CAD*, 13(12), 1994, pp. 1526–1535.
[21] A. B. Kahng and S. Muddu, “Improved effective capacitance computations for use in logic and layout optimizations,” *VLSI DESIGN* 1999, pp. 578–582.
[22] C. V. Kashyap, C. J. Alpert and A. Devgan, “An “Effective” capacitance based delay metric for RC interconnect,” *ICCAD* 2000, pp. 229–235.

0-7695-2093-6/04 $20.00 © 2004 IEEE

IEEE COMPUTER SOCIETY