On modeling the digital gate delay under process variation

Gao Mingzhi(高名之)†, Ye Zuochang(叶佐昌), Wang Yan(王燕), and Yu Zhiping(余志平)

Institute of Microelectronics, Tsinghua University, Beijing 100084, China

Abstract: To achieve a characterization method for the gate delay library used in block based statistical static timing analysis with neither unacceptably poor accuracy nor forbiddingly high cost, we found that general-purpose gate delay models are useful as intermediaries between the circuit simulation data and the gate delay models in required forms. In this work, two gate delay models for process variation considering different driving and loading conditions are proposed. From the testing results, these two models, especially the one that combines effective dimension reduction (EDR) from statistics society with comprehensive gate delay models, offer good accuracy with low characterization cost, and they are thus competent for use in statistical timing analysis (SSTA). In addition, these two models have their own value in other SSTA techniques.

Key words: statistical static timing analysis; comprehensive gate delay model; effective dimension reduction; artificial neural network

DOI: 10.1088/1674-4926/32/7/075010 **EEACC:** 1265A

1. Introduction

Accompanied by the further scaling of the integrated circuit, unavoidable process variation has aggregated the timing issue in advanced digital designs. With the presence of uncertainty, an aggressive design with a tight timing closure in the nominal case would probably result in a very low yield if no analysis regarding the statistical timing behavior was performed. As a response from the EDA society, much research focusing on statistical static timing analysis (SSTA) has emerged in the last decade[1−7]. Those SSTA techniques can be classified into three groups. Block based SSTAs[1−5] propagate statistical delay distributions of some specific forms from the primary inputs to the primary outputs in topological order. Path based SSTAs[6] perform statistical analysis only on those paths with the smallest slack values in a deterministic analysis. Recently, Reference [7] proved that Monte Carlo (MC) analysis, when being done cleverly, can also be a serious solution for SSTA.

Though being studied most thoroughly among these three due to their efficiency and mathematical completeness, block based SSTA usually has a strict requirement on the forms of the gate delay distributions, without which the MAX operator for two path delays that is used a lot in the calculation would lose its efficiency. The most common case is that the gate delay distributions need to be modeled as low order polynomials (a.k.a. response surface model, RSM) with respect to the device parameters under variation, such as gate length L and threshold voltage $V_{\rm th}$, with the 2nd order polynomials[3−5] being prevalent due to their higher accuracy than the 1st order ones. For a digital timing library to be useful, the delay of the gates therein should be characterized under a range of driving and loading conditions, i.e. with different input slopes and output loads (in this work, we only consider a purely capacitive load, an actual one or an effective one[8], while the method can be extended to more general loadings, like simple RC structures). A majority of the researches regarding the statistical gate delay models[9−12] suggested fitting the polynomial delay models separately for different input slopes and output loads. The number of process parameters under variation is about 5 to 10 nowadays[13] for each type of MOSFET, thus making the characterization cost of a statistical library forbiddingly increase to several tens of times of that of a deterministic library, even with some profound methods[9−11].

An alternative approach is to fit the 2nd order polynomials with respect not only to variational process parameters but also to input slope and output load[5]. This method, which we will refer to as a global second order response surface model (global 2-RSM), does cut down the characterization cost sharply, but it offers really poor accuracy. Figure 1 plots the errors in an inverter's delay compared with golden MC data over a 20X range for both input slope and output load. Errors in mean as large as a few sigmas (standard deviation) can be observed. This could be ascribed to the fact that 2-RSM is suitable for

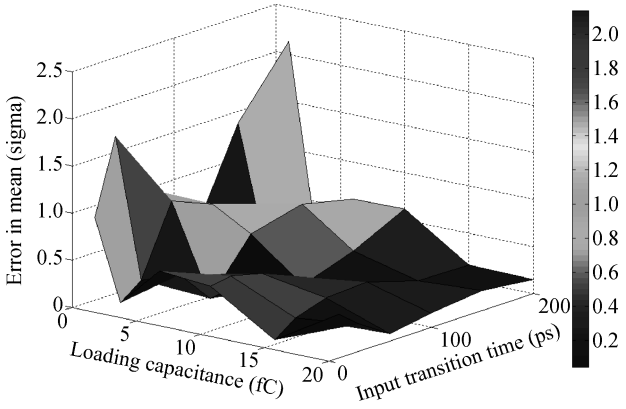

Fig. 1. Errors in the mean of the gate delays for an inverter using global 2-RSM over 20X ranges of both input transition time and loading capacitance. The values are normalized to sigma.

† Corresponding author. Email: gmz02@mails.thu.edu.cn
Received 2 October 2010, revised manuscript received 10 March 2011

expansion around one point and prone to fail when the ranges of parameters become large.

In this work, rather than fitting the polynomial models directly, we propose to use as an intermediary some more general statistical gate delay models. By first characterizing those general models with sampled data from circuit simulation and then using them to cheaply provide the larger set of data required in the fitting of polynomial models under different driving and loading conditions, the overall cost will be drastically reduced and the final model accuracy can be preserved close to that of those general models. In addition, the general statistical gate delay model can be used in other analysis directly, such as the Monte Carlo based SSTA.

The largest hindrance to a general statistical gate delay model lies in the large number of variational device parameters involved, which could further increase in the future process. In this work, we propose to address this problem with an effective dimension reduction (EDR) technique. This is promising to extract a few linear combinations of the variational device parameters, which most effectively dedicate the effect of process variation to gate delay. With those reduced variables, two general purpose statistical gate delay models considering different driving and loading conditions, both of which can be cheaply characterized and offer good accuracy over a range of different conditions and/or parameters, are suggested. The major contribution of this work can be summarized as follows:

(1) We propose to use the EDR technique to reduce the number of random variables (RV) involved in the gate delay model. To obtain the reduced variables, only dc simulation is required and the process only needs to be done for a few representative transistor sizes in a digital library.

(2) With the reduced variables, we apply an artificial neural network (ANN) to get a general purpose statistical gate delay model. With close characterization cost, this model offers obvious improvement in accuracy and stableness over the global 2-RSM model.

(3) We adopt comprehensive gate delay models (CGDM)[14−17] into statistical ones based on the reduced variables. These models apply to both single-stage inverter-like gates and cascade gates. The comprehensive models offer the best accuracy. In addition, the physical meaning of the models may also make it valuable to process engineers who are trying to study or to mitigate the effect of process variation on gate delay.

2. Using efficient dimension reduction in statistical delay modeling

As in other statistical analysis involving standard cells, statistical gate delay modeling usually assumes that the same variational device parameters of the MOSFETs of the same kind in a gate are perfectly correlated. Thus, we can briefly use 'device parameters' when we refer to them in discussion. Usually, in statistical gate delay models, one random variable is assigned to each device parameter under variation to describe the statistical effect. The high dimensionality of RV's does impede developing a statistical gate delay model. What makes it worse is that we cannot tell which process parameters should be regarded as 'under variation' in model development, since they can change from process to process and from time to time. To solve this problem, we employ the EDR method to find a combination of process parameters that indicates most of the effect of process variation on gate delay. In this section, we first briefly introduce EDR techniques and then show how to utilize them efficiently in our application.

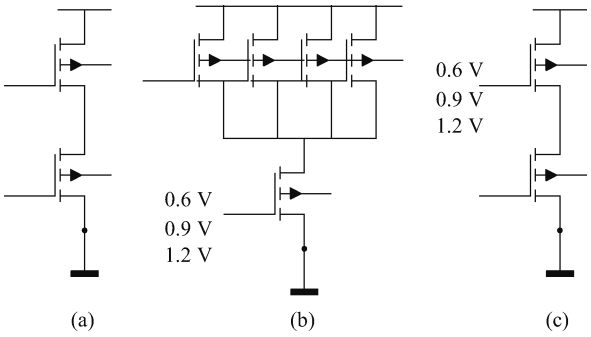

Fig. 2. Test structures for NMOS. The gates of the blue transistors are biased at 0.6, 0.9 and 1.2 V in different tests; the gates of other transistors are biased at $V_{DD} = 1.2$ V.

2.1. Efficient dimension reduction technique

Efficient dimension reduction methods, first proposed in Ref. [18], are effective means for compressing input variables in statistical regression. Assuming m outputs $\boldsymbol{y}$ are functions of n input variables $\boldsymbol{x}$ (and some other unknown factors), EDR methods are engaged to find p ($p < n$) linear combinations of $\boldsymbol{x}$, namely $\boldsymbol{z}$, such that the models of $\boldsymbol{y}$ w.r.t. $\boldsymbol{z}$ will be almost as accurate as those w.r.t. $\boldsymbol{x}$. Formally, consider the relationships between $\boldsymbol{x}$ and $\boldsymbol{y}$, i.e.,

$$y_i \approx f_i(x_1, x_2, ..., x_n), \quad i = 1, 2, ..., m, \tag{1}$$

where f_i are either functions with unknown parameters or totally undiscovered. If the difficulty lies in the high dimensionality of $\boldsymbol{x}$, EDR is used to find a set of reduced variables $\boldsymbol{z}$, i.e., linear combinations of $\boldsymbol{x}$,

$$\boldsymbol{z} = \boldsymbol{K}\boldsymbol{x}, \quad \boldsymbol{K} = (\boldsymbol{K}_{ij})_{p\times n} = (\boldsymbol{k}_1, \boldsymbol{k}_2, ..., \boldsymbol{k}_p)^{\mathrm{T}}, \tag{2}$$

where $\boldsymbol{k}_i$ are reduced directions. With $\boldsymbol{z}$, we can draw some good approximations for f_i,

$$\begin{aligned} y_i &\approx g_i(z_1, z_2, ..., z_p) \\ &= g_i(\boldsymbol{k}_1^{\mathrm{T}}\boldsymbol{x}, \boldsymbol{k}_2^{\mathrm{T}}\boldsymbol{x}, ..., \boldsymbol{k}_p^{\mathrm{T}}\boldsymbol{x}) \approx f_i(x_1, x_2, ..., x_n). \end{aligned} \tag{3}$$

The EDR method we use in this work is Yin's conditional moment based method for multiple outputs[19, 20]. The simplified algorithm is given in Algorithm 1. For further details, please refer to the original literature. One thing we should stress here is that in EDR methods, statistics such as variance and correlation needn't be evaluated very precisely for the method to give good results.

2.2. Reduction of random variables using EDR

In this subsection, we introduce how to use EDR to reduce the number of RV's that are necessary to indicate the statistical behavior in gate delay.

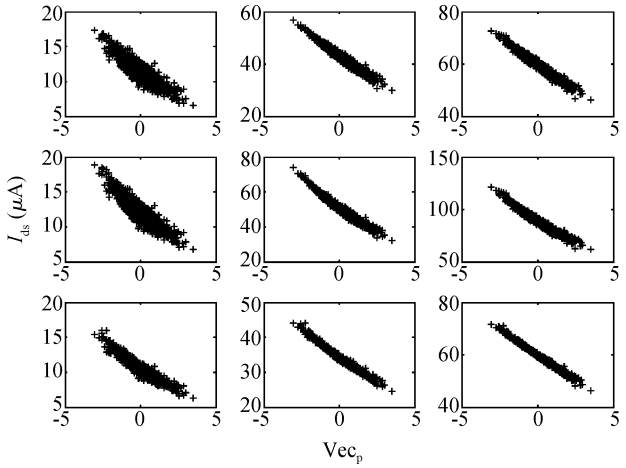

Fig. 3. I_{ds} in test structures versus first reduced variable for PMOS. Each row corresponds to one circuit structure and each column corresponds to one bias condition for the transistor under test.

Algorithm 1: Yin's EDR algorithm[19]

Input: samples of (X, Y)
Output: reduced directions K

1. $\Sigma_X = \mathrm{corr}(X)$ (X_i's correlation), $\Lambda_Y = \mathrm{diag}(\mathrm{var}(Y))$ (Y_j's variance)
2. $U = {\Sigma_X}^{-1/2} * (X - E(X))$, $W = {\Lambda_Y}^{-1/2} * (Y - E(Y))$ (whiten)
3. $K_1 = \mathrm{E}(\mathrm{UW}^{\mathrm{T}})$
4. $K_2 = \mathrm{E}(\mathrm{UW}^{\mathrm{T}} \otimes \mathrm{W}^{\mathrm{T}})$ ($\otimes$ for Kronecker direct product)
5. $K_3 = \mathrm{E}(\mathrm{UU}^{\mathrm{T}} \otimes \mathrm{W}^{T})$
6. $K = [K_1, K_2, K_3]$
7. $K = \mathrm{svd}(K)$ (do singular value decomposition)

In EDR methods described in Section 2.1, in order to get reduced variables, samples of $(\boldsymbol{x}, \boldsymbol{y})$ are needed. To address our problem in gate delay models under variation, $\boldsymbol{x}$'s are no doubt variational device parameters. However, in order to avoiding time-consuming tran analysis on different gates switching under different conditions (this analysis is exactly what we want to save in the characterization of gate delay models), we need $\boldsymbol{y}$ to be an intermedium that doesn't rely on specific gate structures but does tightly relate to gate delay. We note, as have some other researchers[13, 21, 22], that gate delays, to a great extent, depend on I_{ds} of transistors. In other words, I_{ds}, denoting the driving strength of the transistor, will also be a good indication of the driving strength of the gate. Therefore, we pick $\boldsymbol{y}$ as I_{ds} of a transistor under several test conditions. For instance, the test conditions used in this work are shown in Fig. 2. The effects of different loads and biases are considered, as well as the stack effect. We draw random instances of the variational transistor and get the I_{ds} of the test structures with SPICE. Then we employ the EDR algorithm to extract the reduced variables. This procedure is very cheap, since only dc analysis is required and the number of different test conditions, as well as the sample size, can be quite small. More importantly, we don't have to apply this process repeatedly for each gate; we just need to do this for a few transistors to cover a typical range of size in a digital library. A further discussion about this extra cost is included in Section 5.3.

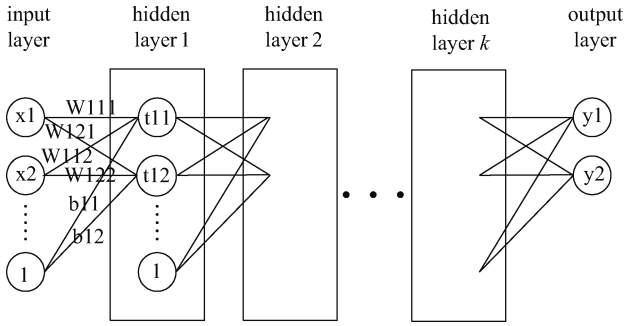

Fig. 4. Illustration of artificial neural network models.

In our experiment, to be described in detail in Section 5, we find from the singular values that the first (i.e. most significant) reduced variables catch 80%–85% of the variation in a transistor's I_{ds}. In a PMOS case, the I_{ds} of different test structures is plotted in Fig. 3 against the first reduced variable, namely $\mathrm{vec_p}$ ($\mathrm{vec_n}$ for NMOS). In all test structures, the (I_{ds}, $\mathrm{vec_p}$) points scatter just in a small range around a curve, indicating that the prediction of $\mathrm{vec_p}$ on I_{ds} is good. This allows us to use only one combined RV to describe most of the statistical effect on the gate delay from one type of MOSFET in the gate. Reference [21] showed that I_{ds} has a linear relation with respect to some major process parameters over a rational region, which also encourages us to use only one reduced RV.

We are not the first to introduce EDR methods to EDA society. The authors of Refs. [9, 23] suggested applying EDR to simplify the models after obtaining the 2-RSM's. Whereas, here, we draw the reduced RV's from transistor's dc current information and find those RV's good indicators of the fluctuation in the driving strength of the gates consisting of these transistors. We will develop statistical gate delay models based on them in the next couple of sections. The idea of employing EDR to obtain some meaningful combined RV's may be useful in other studies of process variation. Such a method can be used in a flexible way in the development of statistical models and algorithms.

3. Gate delay model using an artificial neural network

An artificial neural network is a general methodology for modeling undiscovered relationships. A handful of proven properties indicate that the flexibility in this model is much larger than in polynomials. As shown in Fig. 4, an ANN model can be conceptually denoted by a diagram of nodes in a sequence of layers with weighted connections between nodes in neighboring layers. The first and last layers are input and output layers, respectively, and there is usually at least one hidden layer between them. Every node in this diagram stands for a variable (the nodes in the first and last layer are for input and output). Let t_{ij} be the variable corresponding to the jth nodes in the ith layer, where $i = 0$ stands for the input layer and $i = k+1$ for the output layer. Then, the forward propagation of ANN can be expressed as

$$t_{ij} = f_{ij}\left(\sum_{s=1}^{n_{i-1}} W_{ijs}t_{(i-1)s} + b_{ij}\right), \quad (4)$$

where f_{ij} are pre-determined activation functions. The most common function forms include linear and arctan. The parameters in ANN models, i.e. the weights, can be extracted by a back-propagation training process[24], in which the gradients of the parameters are fed back from the output to the input. To accelerate the training, several algorithms including Levenberg-Marquardt (LM) are often employed.

To apply ANN to a statistical gate delay model, the variables in the input layer are related to the variational device parameters and the output layer indicates the gate delay. A possible choice is to use one ANN for the delays of multiple gates that share a same set of inputs and have similar structures, but in this work, we stick to the basic case that one ANN is used for each gate in the library. The advantage of ANN is that it could handle a large range of parameters and stronger non-linearity, thus making it sensible to include the input slope and the output load in the input variables and still hope for reasonably good results.

Both the training process and the network structure have an effect on the accuracy of the resulting ANN. Not every relation between input and output can be described with a small ANN. However, with a large ANN, a larger set of training data or more reliable training method are required for the model to give a robust result. When the training set is relatively small (as in our case where we may want to cut down the times of the calling circuit simulators,) ANN is prone to be over-fitted. That means that ANN may predict poorly for the input variables not contained in the training set, though it gives acceptable accuracy on the training set.

An approach to make the training process more robust was given in Ref. [25]. In this Bayesian regularization algorithm, not only the errors between the model output and exact values but also the sum of the square of the connection weights are linearly combined into the target of the minimization. With some assumption on the prior distribution of the parameters, a process consistent with a LM algorithm was proposed in which the linear combination coefficients can be automatically selected.

Nevertheless, a more efficient approach to reduce the sample size to an endurable number while ensuring that the trained ANN is still of consistently high accuracy is to use a small set of input variables. A large set of inputs usually complicate the structure of ANN. With the increase in nodes and connections, more variables and parameters are involved, which makes the training process more difficult to stabilize. The EDR technique in Section 2 is an efficient way to cut down the number of inputs. With EDR, one variable for each type of MOSFET rather than the original 5-10 is sufficient. Thus, the number of variables in the input layer is fixed at 4 (including the slope and the load). This also allows more effort to be focused on discovering good ANN structures hopefully working for different manufacturing processes. We found that ANN works well with a structure of either 4-12-1 (one hidden layer) or 4-4-6-1 (two hidden layers) when taking $\mathrm{vec_n}$, $\mathrm{vec_p}$, slope and load as the four inputs. Actually, in our test, the EDR technique, combined with Bayesian regularization, allows the ANN to be stable with $\sim$100 sample data.

After obtaining the ANN model for interesting ranges of working conditions, to extract the polynomial gate delay model is much cheaper, since we only need to sample the gate delay as a response to the deviations in device parameters from the ANN rather than from the costly circuit simulation. With the given driving and loading conditions, the gate delay is a function of locally varying device parameters, thus the low order polynomials are hopefully of high accuracy, which suggests that the final model accuracy will be close to that of the ANN. The detailed procedure is similar to that with the statistical CGDM model, which is presented in Section 4.3.

4. Statistical comprehensive gate delay model

The ANN model, as well as the polynomial response surface, is a general purpose model in function fitting or in 'regression' as used in statistics. Usually, a carefully designed specific model will outperform those general ones by discovering the underlying mechanics, as was done with deterministic CGDMs. After the EDR technique digging out abstract but dominant variables thus addressing the problem with high dimensionality, we are able to develop statistical CGDM for simple inverter-like gates and cascade gates. The basic modeling methodology is to adopt some deterministic models into statistical ones based on observation and induction. Whereas, since the parameters in CGDM are physically meaningful, it is easy to show that the statistical modification is reasonable in most cases. After developing the models, we will introduce the procedure to extract the 2-RSM's used in block-based SSTA via those general models, which can also be applied to the ANN gate delay model. In the following discussion, we usually presume that a rising switching occurs at the input pin of the gate, which makes the description succinct without lose of generality.

4.1. Statistical CGDM for inverter-like gates

In the deterministic context, a variety of physical-based, widely adopted comprehensive models are developed for gate delay[14–17]. A starting point of these models is the behavior of a simple inverter. For CGDMs, there are two typical cases to handle, namely the fast transition case and the slow transition case. In the fast transition case, the input transition rate is fast relative to the output, with the latter majorly depending on the load. The fast changing input signal makes the PMOS off and the NMOS into the saturated region immediately. While in the slow transition case, where the input transition is slow or the loading capacitance is small, PMOS will enter the linear region or even get saturated. Simultaneously, NMOS cannot always discharge the load with its saturation current.

With some mild assumptions, Reference [14] gave a simple analytical form for the gate delay t_d in the fast transition case, i.e.

$$t_\mathrm{d} = \frac{v_\mathrm{TH}}{2}t_\mathrm{in} + (C_\mathrm{Le} + C_\mathrm{Li} + 2C_\mathrm{M})\frac{\tau_\mathrm{ST}}{2C_\mathrm{N}}, \quad (5)$$

where t_in is the input transition time, whose reciprocal is the input slope, v_TH is NMOS's threshold voltage normalized with VDD, C_Le and C_Li are the extrinsic and intrinsic parts of the loading capacitance, C_N is the gate capacitance of the

NMOS, and τ_{ST} is a characteristic parameter standing for driving strength of the switching transistor (see the original paper).

Reference [15] discussed the extremely slow transition case. The output will follow the input as on the dc transfer trajectory. The slope of the output will be proportional to that of the input, and the gate delay is also asymptotically proportional to the input transition time. Thus, we have

$$t_d = t_{in}\left[\frac{1}{2m} - \frac{V_{inv}(1-m)}{mV_{DD}} - \frac{1}{2}\right], \tag{6}$$

where V_{inv} is the threshold of the gate, i.e. the point on the dc curve where V_{in} equals V_{out}, and m is the dc transfer slope, i.e.,

$$m = \Delta V_{out}/\Delta V_{in}. \tag{7}$$

To get a simple but accurate CGDM, we blend the fast transition model in Ref. [14] and the slow transition model in Ref. [15]. We join the two cases with the following factor k and $1-k$,

$$k = \frac{1}{1+\alpha\left\{t_{in}/\left[\left(\frac{\tau_{ST}}{2C_N}\right)(C_{Le}+C_{Li})\right]\right\}^{\beta}}, \tag{8}$$

where α and β are two fitting parameters, whose values change moderately among different gates. In the fast and slow transition cases, Equations (5) and (6) work as the gate delay model, respectively. Then our deterministic CGDM for inverter-like gates, with 7 parameters, is

$$t_d = k[p_1 t_{in} + p_3(C_{Le}+p_2)] + (1-k)p_7 t_{in}, \tag{9}$$

$$k = \frac{1}{1+p_4[t_{in}/p_3/(C_{Le}+p_6)]^{p_5}}. \tag{10}$$

Among the seven parameters, p_1, p_2, p_3, p_6 and p_7 have physical meanings while p_4 and p_5 change in a small range, so the characterization is simple and can be solved by some gradient based methods. In our test, the errors of this model and of the deterministic CGDM for cascade gates given below are well under 1% over a 20X range of both input slopes and loads for various gates.

After developing the deterministic CGDM, we give a statistical CGDM for inverter-like gates and one for cascade gates like AND in the next sub-section. The basic logic underneath these models is that the parameters in a deterministic model have clear physical meanings related to the driving strength of the gate, and the latter is indicated by the reduced variables of the transistors in that gate, as discussed in Section 2. To reveal the relation between the model parameters and reduced variables, a method of observation and induction is employed.

The development of the statistical comprehensive gate delay model is based on the following observation. When we fit the above deterministic model for gates under variation, the parameters vary around their nominal values. Figure 5 shows the plot of parameters of the CGDM for an inverter-like gate against vec_n, the reduced variable for NMOS. We can see from the plots that all of the parameters except p_2 depend heavily on vec_n. Also, p_6 and p_7 are found to have relations with vec_p. The relations between model parameters and reduced variables are

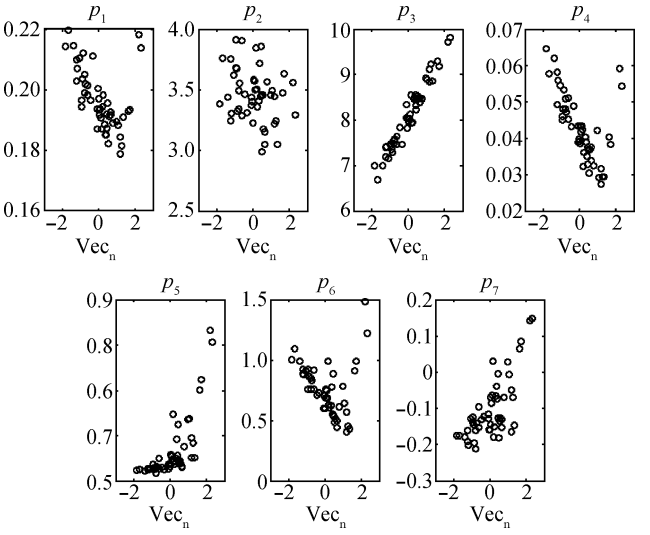

Fig. 5. Parameters in comprehensive gate delay model versus vec_n. This is an illustration of the correlation between reduced variables standing for process variation and the gate delay model parameters.

universal, i.e. they can be observed from different gates in our experiment. Most of these relations are reasonable in a physical sense. For example, p_3 is proportional to τ_{ST}, a substantial parameter indicating the driving strength of the switching transistor, so p_3 has a linear relation with vec_n. p_7 is related to the dc slope and gate threshold voltage, which reflect the driving strength of both NMOS and PMOS, so it depends on both vec_n and vec_p. Based on this observation and its physical interpretation, we modify the CGDM into a statistical one, the parameters of which are simple functions of reduced variables, i.e.

$$\begin{cases} p_1 = p_{10}(1 + p_{1n}\text{vec}_n), \\ p_2 = p_{20}, \\ p_3 = p_{30}(1 + p_{3n}\text{vec}_n), \\ p_4 = p_{40}(1 + p_{4n}\text{vec}_n), \\ p_5 = p_{50}(1 + p_{5n}\text{vec}_n), \\ p_6 = p_{60}(1 + p_{6n}\text{vec}_n + p_{6p}\text{vec}_p), \\ p_7 = p_{70} + p_{7n}\text{vec}_n + p_{7p}\text{vec}_p, \end{cases} \tag{11}$$

together with Eqs. (9) and (10). p_7 is of a different form because p_{70} may be close to zero. The fitting of the statistical model can also be done by a gradient based optimization method with an initial guess set to the values in a deterministic counterpart.

4.2. Statistical CGDM for cascade gates

A widely accepted assumption for cascade gates is that the delay of one cascade gate is the sum of the delays of the inverter-like gates of which the cascade gate is made. That means that we could simply develop a delay model for 2-cascade gates like AND by summing two models for inverter-like gates (which we will refer to as simple gates in this subsection). However, there are still some simplifications that could be made.

First, the input capacitance of the second simple gate is given. This makes the first simple gate's delay only rely on the input slope (or input transition time). Second, the only impact

of the first simple gate on the second is that the former's output slope is the latter's input slope. Furthermore, since the first simple gate only needs to drive a small load, the output transition time is basically small and nearly proportional to the input transition time (except for when the input slope of the first simple gate is quite large), thus making the second gate usually experience a fast transition. Based on these observations, we model the delay of the two simple gates as

$$t_{\mathrm{d1}} = (p_1 t_{\mathrm{in}} + p_2)(1 - p_3 t_{\mathrm{in}}), \tag{12}$$

$$t_{\mathrm{d2}} = (p_4 t_{\mathrm{in}} + p_5 C_{\mathrm{Le}} + p_6)[1 - (p_7/C_{\mathrm{Le}})^{p_8}]. \tag{13}$$

With a similar observation and induction scheme, we modify the CGDM for cascade gates into a statistical one. The parameters in Eqs. (12) and (13) are changed into some simple function of $\mathrm{vec_n}$ and $\mathrm{vec_p}$, just as those shown in Eq. (11), i.e., (note, for a rising input, there is an input falling switch for the second part of the cascade gates).

$$\begin{cases} p_1 = p_{10}(1 + p_{1\mathrm{p}}\mathrm{vec_p}), \\ p_2 = p_{20}(1 + p_{2\mathrm{n}}\mathrm{vec_n} + p_{2\mathrm{p}}\mathrm{vec_p}), \\ p_3 = p_{30}(1 + p_{3\mathrm{n}}\mathrm{vec_n}), \\ p_4 = p_{40}(1 + p_{4\mathrm{p}}\mathrm{vec_p}), \\ p_5 = p_{50}(1 + p_{5\mathrm{p}}\mathrm{vec_p}), \\ p_6 = p_{60}(1 + p_{6\mathrm{n}}\mathrm{vec_n}), \\ p_7 = p_{70}(1 + p_{7\mathrm{p}}\mathrm{vec_p}), \\ p_8 = p_{80}(1 + p_{8\mathrm{p}}\mathrm{vec_p}). \end{cases} \tag{14}$$

For those n-cascade gates, we would not give the models here. These gates are rare in a standard library and the development of their CGDM is easy with the only thing to note being that those simple gates between the first and the last have a nearly constant delay.

We should point out here that our method is capable of creating other statistical CGDMs if some other underlying deterministic models are used, as long as the first few reduced variables could capture most of the delay fluctuation.

4.3. Obtaining 2-RSM's from statistical CGDM

When the statistical CGDMs are obtained, we can extract 2-RSM's (or other gate delay model) for any input slope and load within a range simply by sampling, model evaluation (which only demands a few simple computations) and linear least square fitting. Using this approach, we can obtain gate delay 2-RSM's with respect to reduced variables (i.e. $\mathrm{vec_p}$ and $\mathrm{vec_n}$). Because reduced variables are linear combinations of device parameters, we also obtain 2-RSM's with respect to device parameters. We can use the latter in block based SSTA or we can just use the 2-RSM's with respect to reduced variables (they are close to those being pursued in Refs. [9, 23]). The correlation between variational device parameters can be turned into the correlation between reduced variables. Using the 2-RSM's with respect to reduced variables will offer two other benefits. First, with the reduction in the number of random variables SSTA will run much faster, since the complexities of ADD and MAX operators for 2-RSM based SSTA are at least quadratic functions with respect to the number of RV's in the models. Second, by the combination of device parameters, the normality of the resulting distribution will increase[18], which adds robustness to those SSTA methods assuming normal distributions.

Table 1. Some statistical assumptions in our experiment.

	W	L	T_{ox}	N_{ch}	μ_0	V_{th0}
Relative 3σ	20%	20%	5%	20%	20%	10%
Corr. grad. k	0.05	0.05	0.02	0.03	0.1	0.1
Large 3σ	30%	30%	10%	30%	30%	15%

Algorithm 2: 2-RSM characterization procedure

Input: net-lists for logic gates, process variation information, ranges and specific case for input slopes and loads
Output: 2-RSM's for a series of input slope and load conditions for different gates with respect to reduced variables

1. Extract reduced variables for each transistor in the library or generalized reduced variables for transistors in a region of size using Yin's EDR
2. For each gate, each input port, each switching case
3. Sample the gate delays with different slopes, loads and device parameters
4. Fit corresponding statistical CGDM
5. For each specific slope and load case
6. Fix slope and load, sample $\mathrm{vec_n}$ and $\mathrm{vec_p}$, calculate the gate delay with the model
7. Fit 2-RSM's with reduced parameters
8. end for
9. end for

If a gate delay 2-RSM's library is required, we can use the procedure in Algorithm 2 to characterize the library with our statistical CGDM, which will offer high efficiency and high accuracy, as will be shown in the next section.

5. Computer experiment results and discussions

In this section, we first verify that reduced variables are sufficient to build statistical models for gate delay. Then, we focus on the accuracy of the proposed statistical gate delay models. To compare those models with others, such as global 2-RSM's mentioned in the introduction, we find it fair to use all of the competitors to draw 2-RSM's over a range of input slope and output load for various gates and then compare the accuracy of the resulting 2-RSM's. After that, a discussion on the extraction of reduced variables, and which is the extra cost of those statistical models, is included.

All of the computations are carried out with a single-core CPU working at a frequency of 2.8 GHz. In our experiment, we consider the following six parameters as random sources: W, L, T_{ox}, dopant density in the channel, i.e. N_{ch}, low field mobility, i.e. μ_0, and V_{th0}. All of the sources are assumed to have normal distribution. The 3σ's of these sources are listed in Table 1 (the variation in T_{ox} is small since high-k gate dielectrics would be used in the process node where variation is serious) and we also assume no correlation between different device parameters. In fact, the distribution and the correlation are irrelevant in EDR based methods. Although in most parts of the experiment we use a library of a 0.13 μm process, the extent of variation is set large enough to aim at the use in smaller process

Table 2. Average errors for simple inverter-like gates. Errors in mean and sigma normalized to sigma for the 2-RSM's drawn from the proposed statistical gate delay models and global 2-RSM's are compared to HSPICE MC data. The average is over 36 cases covering 20X ranges of both input slope and load. Only rising signals at the input are tested. The gate with a '@65' string in the name is tested with a 65 nm library.

		Via statistical CGDM (%σ)		Via ANN gate delay model (%σ)		Via global 2-RSM (%σ)	
Gate	#smpl	μ	σ	μ	σ	μ	σ
INV	200	7.88	3.91	7.18	5.33	48.77	42.38
	500	7.30	3.96	5.57	3.83	51.63	12.49
NAND2	200	5.20	3.32	7.56	5.75	18.73	10.36
	500	4.53	3.53	4.57	5.61	17.60	5.12
NOR2	200	6.67	4.77	7.24	5.93	42.11	29.62
	500	5.25	4.25	5.94	5.65	42.78	8.81
NAND3	200	4.97	4.88	6.13	4.95	5.09	3.43
	500	4.32	3.62	4.76	5.50	4.77	2.90
NOR3	200	6.17	3.97	9.37	5.46	39.34	10.89
	500	4.98	2.99	5.50	3.73	40.67	7.49
INV@65	200	8.50	5.23	8.01	5.60	58.41	62.38
	500	8.41	3.46	6.57	5.47	57.31	21.52
NOR2@65	200	6.07	5.28	8.67	4.91	48.58	37.13
	500	5.93	4.09	6.80	4.51	48.42	16.30

nodes. We use NMOS of 0.3 μm/0.13 μm (W/L) and PMOS of 0.6 μm/0.13 μm (W/L) in all of the gates for simplicity. In the revision of this work, we also tested the proposed method with a 65 nm library (see Section 5.2 for details). We normalize all of the errors in both mean and sigma to sigma in this work. All of the errors are calculated in absolute value before they are normalized or averaged.

5.1. Basic results on delay 2-RSM's with respect to reduced RV's

We first demonstrate the accuracy of the 2-RSM's with respect to reduced variables compared with those with respect to original device parameters. This gives us the basic expectation of how well reduced RVs can describe the gate delay. After getting the reduced variables, we do random sampling and fit gate delay 2-RSM's with those reduced variables directly. 2-RSM's with respect to the first reduced variable of each transistor type, 2-RSM's with respect to the first two reduced variables and 2-RSM's with respect to original variables are gained with 30, 50, and 200 samples, respectively. The errors in mean and sigma, compared with the SPICE Monte Carlo results, of the three cases are under 5%, 5% and 1.5% sigma, respectively. The extra error caused by using reduced variables is below 4% sigma, even with far fewer samples. The effects of using the first one or two reduced variables of each MOSFET are close, thus we use only one reduced variable in statistical gate delay models.

5.2. Results on the statistical gate delay models

Here we present the experimental results on proposed statistical gate delay models. To compare their accuracy with global 2-RSM, which has a close characterization cost, we suggest comparing the accuracy under some specific input and output condition. And since the global 2-RSM was proposed in 2-RSM based SSTA, we feel it is fair to compare the accuracy of 2-RSM's drawn from all of the models.

The tested range of input transition time is [10 ps, 200 ps] and the range of load is [1 fF, 20 fF]. We pick some typical inverter-like gates and some cascade gates in this experiment. For each input of each gate, we first fit both models with SPICE results, and then extract 2-RSM's with specific slopes and loads. The results in Tables 2 and 3 are averaged over 36 specific input slope and load conditions and over three independent runs. The results of the same gate but at different pins are averaged to save the length. In our test, to fit the models, sample sizes of 200 or 500 are used. For the ANN model, as we mentioned in section 3, a structure of either 4-12-1 (one hidden layer) or 4-4-6-1 (two hidden layers) works well. The results of these two structures are close and appear a bit better than other structures with one or two hidden layers in our experiment. We list the results from the 4-12-1 in Table 2.

The average errors in mean (μ) and in sigma (σ) normalized to sigma for inverter-like gates are listed in Table 2. The errors of global 2-RSM's can be very large or quite small, depending on the gate. Since it is hard to predict what gates will result in good accuracy, the use of the global 2-RSM method is limited to handling situations where small ranges of input slope and load are supposed. The proposed ANN model offers better and more stable results. Since the mean/sigma ratio is about 8–9 in our experiment, the relative error in mean is always below 1%, while in a few cases the relative error in sigma will be around, or a little above, 5%. If we just use the original variational device parameters rather than the reduced variable as the input of the ANN, sometimes we can obtain acceptable results. However, it also happens sometimes that the results will be quite poor. The statistical CGDM gives the best results among the three where the average errors are below 5% sigma even with as few as 200 data (it can be fitted with even $\sim$100 data). In all of the runs, the maximum errors of the CGDM are less than 18% sigma in mean values and less than 12% sigma in sigma values. Considering the 8–9 mean/sigma ratio, the maximum relative error in mean is around 2%. The results on cascade gates are listed in Table 3. The ANN model behaves very similarly and we choose to omit the numbers to allow this table to be put in one column. Interesting enough is to have a look at the results from global 2-RSM's. In this case, it even outperforms the proposed statistical CGDM, even though the latter continues to provide solid results. It can be ascribed to the fact that for cascade gates, the input slope only has a small effect,

Table 3. Average errors for some cascade gates. The settings are the same as in Table 3. Only results from the proposed statistical CGDM and the global 2-RSM are listed to save space.

		Via statistical CGDM (% σ)		Via global 2-RSM (% σ)	
Gate	#smpl	μ	σ	μ	σ
XOR	200	7.56	4.59	7.09	5.03
	500	6.23	3.83	7.28	4.18
BUF	200	4.60	5.59	5.04	2.58
	500	4.32	4.60	5.17	2.77
AND2	200	4.43	4.40	6.74	2.24
	500	4.38	3.80	6.68	2.51
OR2	200	3.62	5.25	3.85	2.33
	500	2.72	3.11	3.57	1.85
AND2@65	200	5.62	5.71	7.31	4.58
	500	5.25	4.34	6.99	2.40

and the gate delay depends almost linearly on the output load in the test range. However, it is still of some risk to use 2-RSM's when the concerned range changes.

We just mentioned in Section 4.3 that we can rely the SSTA on those 2-RSM's with respect to reduced variables, i.e., vec_p and vec_n, which is a more natural approach with our models and enjoys some theoretical advantages. As a simple test to see how well the 2-RSM's with reduced variables will do in SSTA, we use these models to predict the periods of inverter ring oscillators (RO). The correlation of each device parameter between gate i and gate j is assumed to have a linear form $c_{ij} = \max(1 - k*|i - j|, 0)$. The gradients k for different random sources are listed in Table 1. The correlation between reduced variables of different gates can be drawn from these correlation values. We then perform principal component analysis (PCA)[2] to get a further reduction of random variables in the circuit. Finally, we use the resulting principal components and the 2-RSM to predict the period of the RO in the same way as in SSTA. The results are listed in Table 4 compared with the SPICE MC results. Errors of less than 4% sigma can be observed in most cases. The pdf and cdf are plotted in Fig. 6.

Actually, our model will offer good accuracy even when the variations are quite large. The variations being as large as those listed in the last line of Table 1, the average errors in gate delay 2-RSM's of an inverter fitting from a statistical CGDM are 1.3% and 4.0% for mean and sigma, respectively. In this test, 300 SPICE data are used to fit the model.

Finally, to address the generality of the proposed method, tests with another 65 nm library were also done. The NMOS of 150 nm/65 nm (W/L) and PMOS of 300 nm/65 nm (W/L) are used in the tests. The ranges of slope and load change to [5 ps, 100 ps] and [0.2 fF, 4 fF] accordingly. The results on a few gates are listed in Tables 2 and 3 with the gate names suffixed with '@65'. We come to the same conclusion with the different library.

5.3. On the extraction of reduced RV's

The reduced RV's are important to both of our models. The cost of extracting those reduced variables is thus worth discussion since it adds to the total cost of those models and thus to the polynomial gate delay model characterization procedure using those models as intermediaries.

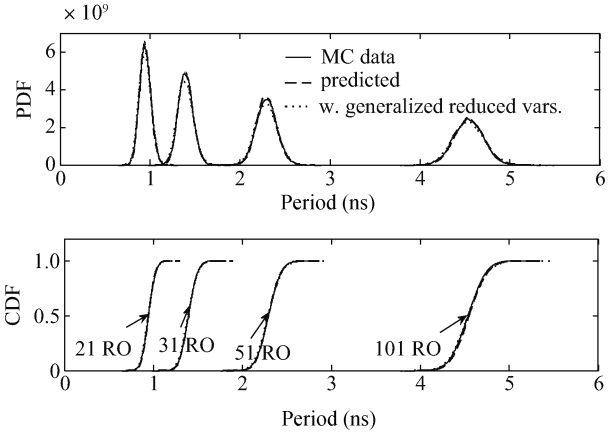

Fig. 6. PDF and CDF for distribution of INV ROs periods. MC results and the results predicted using 2-RSM with reduced variables and generalized reduced variables are presented.

Table 4. Errors in some statistics of the predicted RO periods normalized to sigma. Results in parentheses are with generalized reduced variables.

Num.	Mean (% σ)	Sigma (% σ)	CDF (% σ)
21	0.18 (5.08)	1.81 (5.71)	3.55 (4.67)
31	3.12 (3.85)	0.15 (7.23)	1.91 (6.68)
51	1.77 (3.01)	3.99 (4.75)	4.01 (4.71)
101	7.12 (2.30)	1.85 (9.07)	3.02 (7.71)

In our experiment, we extract the reduced variables from 1000 random data, which take about one hour on a personal computer (single core) with most of the time spent on file I/O and parsing. Furthermore, we find that the reduced variables do not need to be very accurate to get good models for gate delays. A set of 300 samples seems enough, since the cosine values of the angle between the first reduced directions obtained from a run of 1000 data and a run of 300 data is always more than 0.96 in our test.

For practical libraries consisting of an amount of different sized transistors, we do not need to extract reduced variables for every single transistor. Transistors of similar physical size can be grouped and we only need to draw generalized reduced variables for each group. To do this, we only need a way to express the variations on different transistors with a uniform random variable. This relies on the properties of process variations, i.e. for each parameter, whether relative or absolute variation is constant, or the variation obeys Pelgrom's formula, etc. When the transistors in a gate are not the same (but close to each other, as in most practical cases), we can also describe the variations with generalized reduced variables. The extra effort of extraction of reduced variables is further cut down by using generalized reduced variables. For example, we draw generalized reduced variables from NMOS of 0.3 μm/0.13 μm and of 0.6 μm/0.13 μm, assuming that the relative variations are the same. Generalized reduced variables are also drawn from PMOS of 0.6 μm/0.13 μm and of 1.2 μm/0.13 μm. The predicted periods of inverter ROs using these generalized reduced variables are still of good quality, as shown in Table 4 and Fig. 6.

6. Conclusion

In this work, we have proposed an artificial neural network based model and statistical comprehensive gate delay models, which stem from physically meaningful models to support the statistical timing analysis, such as providing an efficient method to build a statistical polynomial gate delay library as required in block based SSTA or being used directly in Monte Carlo based SSTA. The major enabler of those models is the use of an effective dimension reduction technique in statistical gate delay modeling, cutting down the number of variables sharply.

Those models take into account the effect of both process variation and gate operation conditions including input slope and output load, and they exhibit good accuracy over a practical scope, even though the fluctuations in device parameters are large. The characterization cost of those models is well under control and only 100–200 SPICE runs will be enough to cover all operation conditions, especially in the case with the statistical CGDM. The physical meaning of the model also provides a manner to connect process variation with gate delay directly.

We have shown how to extract 2nd order response surface models with the proposed models. The resulting accuracy is much better than an existing method of a close characterization cost. We have also argued that the extra cost of our method is very low. The EDR method is among the key techniques in developing this model and we are hopefully to see more applications of this technique in the study of process variation.

References

[1] Visweswariah C, Ravindran K, Kalafala K, et al. First-order incremental block-based statistical timing analysis. DAC, 2004

[2] Chang H, Sapatnekar S. Statistical timing analysis considering spatial correlations using a single pert-like traversal. ICCAD, 2003

[3] Zhan Y, Strojwas A, Li X, et al. Correlation aware statistical timing analysis with non-Gaussian delay distributions. DAC, 2005

[4] Zhang L, Chen W, Hu Y, et al. Correlation-preserved analysis with non-Gaussian statistical timing quadratic timing model. DAC, 2005

[5] Bhardwaj S, Ghanta P, Vrudhula S. A framework for statistical timing analysis using non-linear delay and slew models. ICCAD, 2006

[6] Amin C S, Menezes N, Killpack K, et al. Statistical static timing analysis: how simple can we get. DAC, 2005

[7] Singhee A, Singhal S, Rutenbar R A. Practical, fast Monte Carlo statistical static timing analysis: why and how. ICCAD, 2008

[8] Abbaspour S, Fatemi H, Pedram M. VGTA: variation-aware gate timing analysis. ICCD, 2005

[9] Feng Z, Li P. A methodology for timing model characterization for statistical static timing analysis. DAC, 2007

[10] Kumar Y S, Li J, Talarico C, et al. A probabilistic collocation method based statistical gate delay model considering process variations and multiple input switching. DATE, 2005

[11] Li X, Le J, Pileggi L, et al. Projection-based performance modeling for inter/intra-die variations. ICCAD, 2005

[12] Okada K, Yamaoka K, Onodera H. A statistical gate-delay model considering intra-gate variability. ICCAD, 2003

[13] Watts J. Modeling circuit variability. Workshop on Compact Variability Modeling, 2008

[14] Daga J M, Auvergne D. A comprehensive delay macro modeling for submicrometer CMOS logics. IEEE J Solid-State Circuits, 1999, 34: 42

[15] Dutta S, Shetti S S M, Lusky S L. A comprehensive delay model for CMOS inverters. IEEE J Solid-State Circuits, 1995, 30: 864

[16] Bisdounis L, Nikolaidis S, Koufopavlou O. Analytical transient response and propagation delay evaluation of the CMOS inverter for short-channel devices. IEEE J Solid-State Circuits, 1998, 33: 302

[17] Sakurai T, Newton A R. Alpha-power law MOSFET model and its applications to CMOS inverter delay and other formulas. IEEE J Solid-State Circuits, 1990, 25: 584

[18] Li K C. Sliced inverse regression for dimension reduction. J Amer Statist Assoc, 1991, 86: 316

[19] Yin X, Bura E. Moment-based dimension reduction for multivariate response regression. Journal of Statistical Planning and Inference, 2006, 136: 3675

[20] Yin X, Cook R D. Dimension reduction for the conditional kth moment in regression. J Roy Statist Soc Ser B, 2002, 64: 159

[21] Shinkai K, Hashimoto M, Kurokawa A, et al. A gate delay modeling focusing on current fluctuation over wide-rang of process and environmental variability. ICCAD, 2006

[22] Wang V, Markovic D. Linear analysis of random process variability. ICCAD, 2008

[23] Mitev A, Marefat M, Ma D, et al. Principle hessian direction based parameter reduction with process variation. ICCAD, 2007

[24] Rumelhart D E, Hinton G E, Williams R J. Learning internal representations by error propagation. Parallel Data Processing, 1986, 1(8): 318

[25] Foresee D, Hagan F. Gauss-Newton approximation to Bayesian learning. International Conference on Neural Networks, 1997