Voltage Scalable Statistical Gate Delay Models Using Neural Networks

Bishnu Prasad Das[1], Bharadwaj Amrutur[2], H.S. Jamadagni[3]

Abstract

We propose a technique to model the delay of logic gates which captures the variability of process parameters considering intra-gate variability and is voltage scalable, using feed forward neural networks. These models can be used to efficiently generate the delay statistics across different supply conditions for use in statistical timing analysis, with very small loss of accuracy compared to SPICE based Monte-Carlo approach. The models incorporate the load and input slews as parameters and hence do away with table lookups as in the conventional approach. The model also captures temperature variation inside the chip. We demonstrate an application of the voltage scalability by using the models to do statistical timing analysis in a Dynamic Voltage Scaling framework, to obtain the optimum supply voltage for a target delay with specified statistical guarantees for some ISCAS 85 benchmark circuits.

Keywords: Intra-gate Variability, Neural network, Voltage Scalable models, statistical timing analysis

1. Introduction

On-chip variations are becoming an increasing concern in integrated circuits as transistor densities continue to increase and feature sizes continue to shrink. As device parameters such as width, length, threshold voltage of the transistor and environmental conditions such as temperature shows variability, the prediction of circuit performance, both in terms of delay and power has become a challenging task [1-3]. The conventional approach to handle this problem of variability is to use a few process corners, which capture the boundary of the process variations, to analyze the design. But such an approach is not suitable for nanometer scale devices because of the pessimistic prediction of performance spread, which inevitably leads to over design and performance loss for the typical case [2]. Hence, a new analysis paradigm based on statistical models is emerging, which attempts to incorporate the complexities of intra-die and inter-die variations in a more fine-grained way to result in less over designing and better quality results [5-9] .

Most works reported till now [5-10], only consider gate level intra-die variations. The authors in [4] show that without considering intra-gate variation, the errors can be substantial. However, to accurately model intra-gate variations, one has to consider every transistor in the gate separately, thus greatly

[1] CEDT, IISc, Bangalore, India, Email: bpdas@cedt.iisc.ernet.in
[2] Department of ECE, IISc, Bangalore, India, Email: amrutur@ece.iisc.ernet.in
[3] CEDT, IISc, Bangalore, India, Email: hsjam@cedt.iisc.ernet.in

increasing the number of parameters. For a complex gate with *p* transistors and *n* parameters per transistor, we would need *np* parameters as input to the delay model. The authors in [4] propose using sensitivity based enhancements to a linear delay model. A linear delay model is obtained first (by not considering intra-gate variations), using the Response Surface Method (RSM). Linear sensitivity based terms capturing the intra-gate variability effects are then added to this model. The models however don't incorporate the loads and input slews explicitly. Rather the model parameters are obtained for a range of load and slew values and stored in a table. Furthermore, the delay errors are quite substantial under low load and large slew conditions. Another limitation of sensitivity based linear models is that they work only for small variations. With technology scaling however, the percentage variation is actually increasing and is especially worse for small sized gates which need to be used for low power applications. Another drawback of existing modeling techniques is that they are generated for a specific voltage value. Applications which need to operate over a wide range of voltage will need a table of models, similar to the load, slew tables.

In this paper, we propose a closed form delay and output slew model based on feed forward neural networks, which addresses the aforementioned shortcomings of the existing models. Neural networks are known to provide good approximations to high dimensional non linear functions [13-15]. Because of variability, the delay of gate is high dimensional and nonlinear, hence we chose to use this to model gate delays and output slews with intra-gate variations. The delay and output slew of a gate are modeled as a neural network function of transistor width, length and threshold of every transistor, temperature, load, slew (rise time and fall time) and supply.

$$D = f(\overline{W}, \overline{L}, \overline{Vth}, V_{dd}, Temp., Load, rt, ft) \quad (1)$$

Here $\overline{W}, \overline{L}, \overline{Vth}$ are vector of width, length, threshold voltage of all the transistors in the gate. $V_{dd}, Temp., Load$ are the supply voltage, temperature of the substrate and the output load capacitance of the gate respectively. *rt and ft* are the rise and fall slew time of the gate input. A similar equation models the slew rate of the output of the gate.

Since we include the supply voltage explicitly in our model, we can create voltage scalable models which are suitable for use in emerging statistical timing analysis in Dynamic Voltage Scaling (DVS) framework.

Our contributions in this paper includes

1. The proposal and evaluation of feed forward neural networks for voltage scalable gate delay (and output slew) models to be used in statistical timing analysis.
2. A computationally more tractable modeling framework with accuracy close to full SPICE based models.
3. A single model which covers (i) each transistor threshold voltage, width and length variation **(intra-gate variability)** (ii) 0^0-120^0C temperature range (iii) 1x-10x output load range (iv) range of input slew rates (v) range of supply voltage. These many parameters and variation ranges have not been considered in the previous papers [4-9].

4. The model is independent of distribution of underlying parameters and hence can be used with non-Gaussian distributions too.
5. A Monte-Carlo analysis on the proposed model can produce delay **(rise delay, fall delay, rise slew and fall slew)** Probability Density Function (PDF) specific to each supply, load and slew which will helpful in finding the accurate delay statistics in a statistical timing analysis. Monte-Carlo using the model is significantly faster than doing Monte-Carlo on SPICE, yet the accuracies are comparable.

In the next section, we will provide a brief background of the neural network and explain our methodology of using these. In Section 3, we study the problem of creating a statistical delay model for logic gates and show how neural network based model can provide significant computational speedups without too much loss of accuracy. Simulation and results are described in section 4. We finally present our conclusions in Section 5.

2. Background on Neural Network

We have used a three layer neural network with tan-sigmoid activation functions (Figure 1) [13]. The network has an input layer, an output layer and an internal hidden layer.

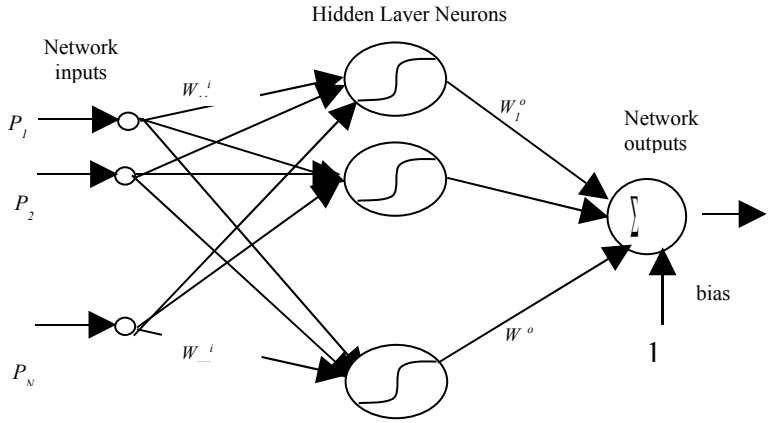


Figure 1. Schematic of tan-sigmoid Neural Network

The network can be used to approximate a real multi-variable function and its output is given as:

$$\hat{f}(P) = \sum_{j=1}^{M} \phi_j \left\{ \sum_{k=1}^{N} (P_k * W_{jk}^{i}) + b_j \right\} * W_j^o + bias \qquad (2)$$

The tan-sigmoid function used in the network is given by

$$\phi_j(X) = tanh(X) \qquad (3)$$

Where, W_{jk}^{i} is the weight in the input layer.

W_j^o is the weight in the output layer.

P is the vector of inputs to the network

bias and b_j are the bias input at the output layer and hidden layer (not shown in the figure for simplicity) respectively.

The weights on the input layer and output layer can be adjusted, making it very suitable for approximating high dimensional functions [14-15]. The functions to model delays of gates are very high dimensional and nonlinear, and hence the neural network is a suitable template for this modeling task. Thus the inputs to the network in the case of a gate delay model will be the width, length, threshold voltage of all the transistors, the supply voltage, temperature, output load and the input edge rates and the output of the network will be the gate delay. A similar network can be made to model the output slew (or edge rate). Note that one would need separate models for the rising and falling edges of the gate, leading to two models each for delay and output slew, per input of the gate. The neural network is trained (i.e., the input weights, output weights and bias are determined), with a small set of data points called training set which represents the functionality of the underlying system being modeled. While it is not necessary to know the internal structure of a system in order to model it using neural networks, it is necessary to have examples corresponding to the behavior between the inputs and outputs. We use a circuit simulator, HSPICE, to generate the data sets needed for training. A small number of Monte-Carlo simulations of the circuit to be modeled, is carried out to generate the training data sets. Hence, even though complicated MOS equations are not used inside the neural network, we are able to get reasonably accurate models. We use MATLAB's built in functions to train the network. The Levenberg-Marquardt training algorithm is used during model creation. The number of iterations of this algorithm, called epochs, determines the training time. A second set of data points, again generated via HSPICE simulations, but not present in the training data set, is used to validate the neural network model and compute the approximation error. The network is accepted as a valid model only if the error is within some limits as shown in the flow chart in Figure 2.

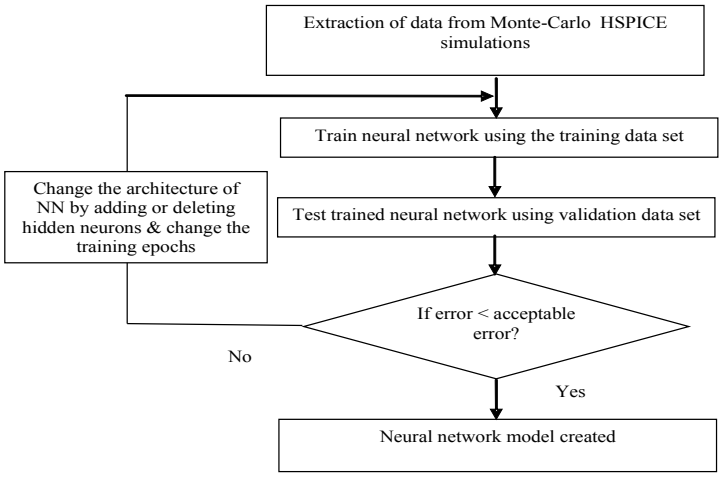


Figure 2. Flow chart for creating neural network model

The process of training and modification of the network in case it is not suitable, are well established techniques and details can be found in reference

[13]. An important point to note here is that all the inputs to the network have to be suitably scaled between 0 and 1, and the output of the network is also scaled by dividing the output by the maximum of the output data. Next we will discuss in detail the application of these networks for delay modeling of digital logic gates.

3. Gate Delay and Output Slew Modeling using Neural Network

Traditionally, a SPICE model of the circuit block implicitly captures the delay model of Equation 1. Exhaustive Monte-Carlo SPICE simulations are carried out, using parameter values randomly drawn from their distributions. The resultant delay values are then used to form the delay PDF. The problem is further complicated if one wanted to obtain the PDF for different voltage values. In this paper, we propose an approach of using a small number of SPICE based Monte-Carlo simulations to train a neural network to model the delay in Equation 1. Monte-Carlo evaluation of the model with random parameter values, drawn from their distributions, is then used to characterize the delay PDF. Since the second, more exhaustive, Monte-Carlo analysis is done on the model, it is computationally more efficient than SPICE simulations. Also the model incorporates supply voltage into it, so one can create a PDF at any supply voltage easily.

During the training of the neural network, the sample for the supply voltage, temperature, rise time and fall time of the inputs are taken using uniform sampling. But, however the samples for output load are taken non-uniformly. This is because the delay variation for smaller load is more nonlinear as compared to the larger load. The width and length of the transistors are taken from uniform distribution. The threshold voltage of all the transistors are varied as Gaussian distribution [11-12] with worst case variation specified in the model file. The percentage variation of parameters is shown in Table 1.

The variation of rise and fall slew and capacitive load has been taken very high to show how effectively neural network is able to incorporate these kinds of large variation in delay modeling.

Table 1. Parameter statistics for gate delay modeling.

Statistical parameters	Variation/Range	Statistical distribution
W_n, W_p	±10%	Uniform
L_n , L_p	±10%	Uniform
Vth_n	±12%	Gaussian
Vth_p	±12%	Gaussian
Supply Voltage	0.5v- 1.1v	Uniform Sampling
Temperature	0^0C to 120^0C	Uniform sampling
Rise slew	0-100ps	Uniform sampling
Fall slew	0-100ps	Uniform sampling
Output Load	1x-10x	Non-Uniform sampling

3.1 Dividing and conquering the data set

Our aim is to produce voltage scalable delay PDFs across the supply voltage range from 0.5v to 1.3v across process, edge rate, load and temperature. We

found it effective to divide the whole data into four regions with respect to supply to increase the accuracy of estimation and reduce the number of data samples required for training. We used four regions across the supply i.e 0.5-0.7, 0.7-0.9, 0.9-1.1, 1.1-1.3. This leads to four model sets for each gate, depending upon supply voltage range. Each model set was created using 675 training samples.

4. Simulation & Results

All gates are simulated using an industrial 130nm model file in HSPICE environment. A small data set of 675 samples, generated from the HSPICE Monte-Carlo simulation, is used to train the neural network. Table 2 indicates the number of hidden layer nodes, the epochs and training timing needed to obtain the models for the gate delay for the various gates, in the supply voltage range of 1.1-1.3v. Even though the delay of a gate is a very high dimensional function, the maximum number of hidden nodes in the network is 12.

Table 2. Number of Hidden nodes, Epochs and Training Time

Supply Range =1.1v -1.3v				
Gate types	Rise Delay/Fall delay/Rise Slew/Fall Slew			
	Hidden Nodes	Epochs	Training Time in seconds	No. of input parameters to Model
Inverter	12	80	5.76	11
NAND2	12	80	9.6	17
NAND3	12	120	22.91	23
NOR2	12	150	17.57	17
NOR3	12	150	28.83	23

A major benefit of this model is that it includes supply voltage as one of the parameters and hence can be used to generate probability density functions across supply voltages. A Monte-Carlo model evaluation of the neural network model is performed to predict the output delay (and slews) using random input data from the same distribution. The same input data is also fed to the SPICE simulation for accuracy.

Figure 3 shows the comparison of probability density functions generated by the neural network and HSPICE for 2-input NAND gate and 3-input NOR gate at the supply voltage of 0.5v, 0.9v and 1.2v. It can be seen from the figure that the PDF generated by the NN model matches closely with that generated using HSPICE even across this large voltage range.

Table 3 shows the mean and variance comparison between HSPICE and NN model. The result shows that the % error in mean and variance for supply voltage 0.5v and 1.2v is within 2%. Figure 4 shows the comparison of the PDFs for different types of delay like rise delay, fall delay, rise slew and fall slew for 2-input NAND gate generated with both their respective neural network (NN) models and HSPICE. It shows that neural network models can accurately model all these types of delays accurately. Tables 4 shows the mean and variance comparison for 2-input NAND gate between HSPICE and NN-model at 0.5v

supply voltage for different delay types. The maximum error by the NN model is within 2% of HSPICE.

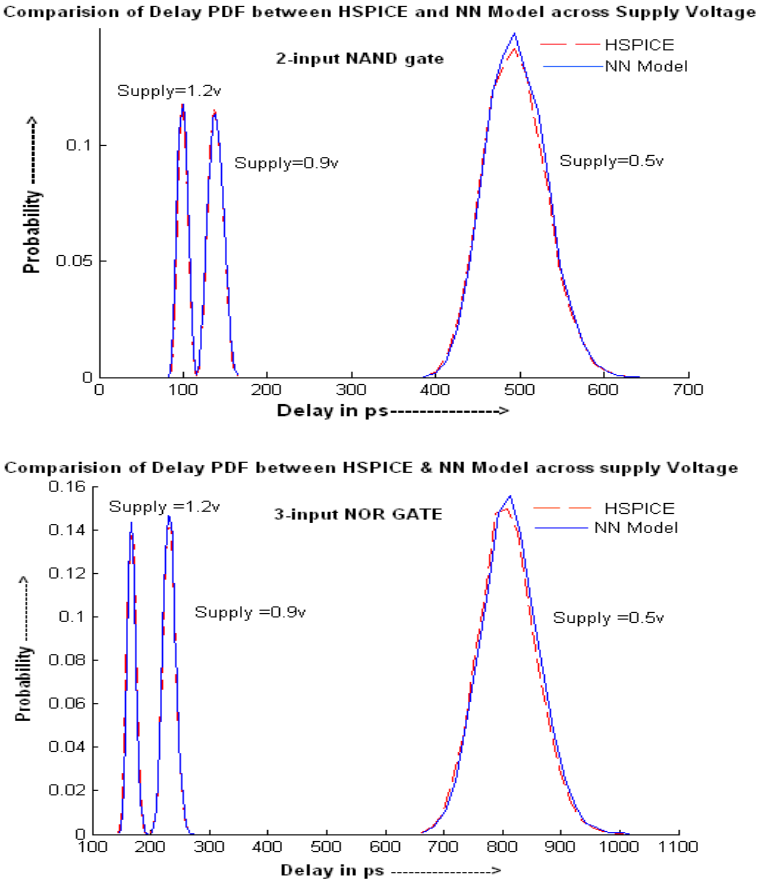


Figure 3. Comparison of Delay PDF of 2-input NAND & 3-inputNOR gate across supply voltage 0.5v, 0.9v and 1.2v

Table 3. Statistical comparisons between HSPICE & NN Model across supply voltage

For Supply Voltage 1.2v						
Gate type	Mean in ps			Std dev. in ps		
	HSPICE	NN Model	% Error	HSPICE	NN Model	% Error
Inverter	87.25	87.27	0.02	4.68	4.69	0.21
NOR2	131.66	131.7	0.03	6.27	6.29	0.31
NAND2	98.08	98.44	0.36	5.54	5.58	0.72
NOR3	166.05	166.23	0.10	7.29	7.19	1.37
NAND3	115.47	115.94	0.40	6.71	6.77	0.89

For Supply Voltage 0.5v						
Gate type	Mean in ps			Std in ps		
	HSPICE	NN Model	% Error	HSPICE	NN Model	% Error
Inverter	355.21	355.97	0.21	28.01	28.30	1.03
NOR2	603.34	601.79	0.25	38.46	38.93	1.22

NAND2	487.59	488.91	0.27	36.0	35.42	1.61
NOR3	799.64	803.45	0.47	48.80	48.73	0.14
NAND3	615.01	613.97	0.16	43.74	44.50	1.73

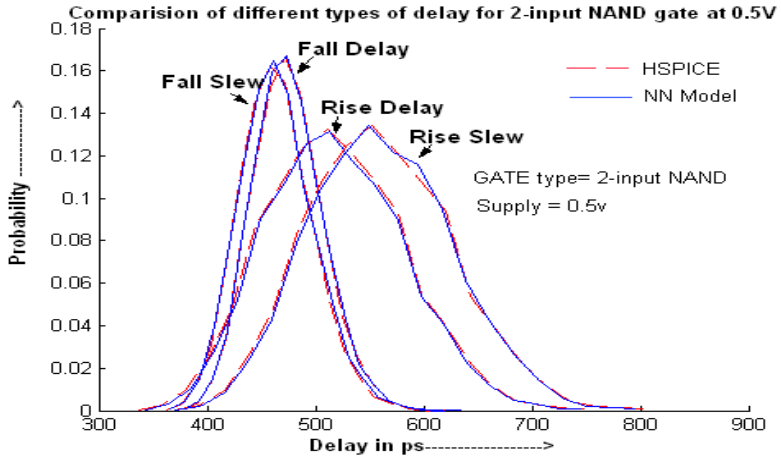


Figure 4. Comparison of different types of delay for 2-input NAND gate at 0.5v

Table 4. Statistical comparison between HSPICE & NN Model for different types of delay

For Supply Voltage 0.5v , GATE = 2-input NAND						
Delay type	Mean in ps			Std in ps		
	HSPICE	NN Model	% Error	HSPICE	NN Model	% Error
Rise delay	507.30	508.14	0.16	63.18	62.42	1.2
Fall delay	467.88	467.68	0.04	32.82	32.67	0.45
Rise Slew	548.43	549.79	0.24	65.87	64.96	1.38
Fall slew	456.86	457.53	0.14	33.74	34.34	1.77

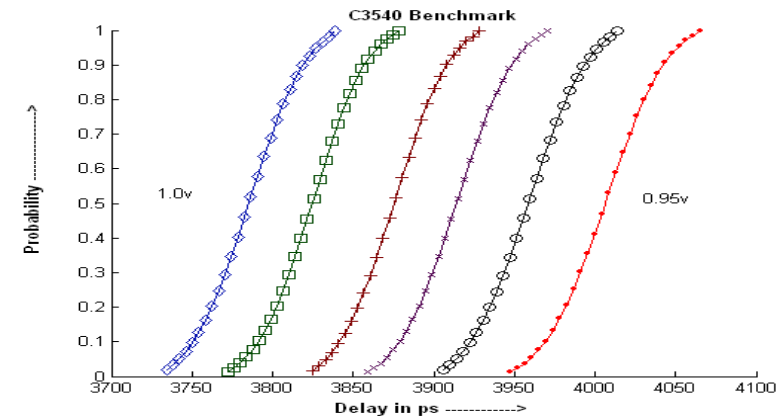
4.1 Optimum Supply Voltage with Statistical Guarantees

We demonstrate the usefulness of voltage scalable delay models by applying them to do statistical analysis in the Dynamic Voltage Scaling (DVS) framework. DVS is a low power technique where the supply voltage is adjusted according to the target speed requirements [16]. The main problem here is given a target delay, to find the minimum supply voltage such the circuit will operate correctly. When one considers variations, then the problem needs to be formulated as: given a target delay and yield, find the minimum supply voltage.

We have implemented a statistical analysis engine in C++, which propagates the delay PDFs from inputs to outputs of a circuit as in [5][8]. Since our delay models are voltage scalable, we can not only find the delay PDF at any supply voltage, but also find the minimum supply voltage which will guarantee a delay target with a certain yield.

Figure 5 shows the delay CDF (Cumulative Distribution Function) of the ISCAS C3540 benchmark circuit across supply voltages from 0.95v to 1.0v, with 10mV steps. For any supply voltage, the engine first obtains the PDF of each gate in the circuit by doing a Monte Carlo evaluation of the gate's delay

model using the following variations: a ±4% uniform variation in the gate's load capacitance and process parameters like width, length and threshold voltage are given in Table 1, ±5% variation of input edge rate and a uniform variation of the temperature from 70^0C to 110^0C. The delay PDF is evaluated at a worst case supply which is –5% below the target supply as the probability distributions for the supply variations are not easy to obtain. Note that one can use the same argument to also use a worst case value for the temperature. Each circuit path's CDF is then calculated by the statistical timing analysis engine and is shown in Figure 5. One can use such an analysis to predict the lowest supply voltage which meets a certain target delay with a specified yield. For e.g., from the Figure 5, we can see that a better than 95% yield for a delay of 4ns is obtained for supplies greater than 0.96V.



From these curves we can also address another question, viz., supply voltages which meet a target delay at two different yield points. This will enable the designer to back away from over designing for worst case corners and choose more realistic yield points. Hence such statistical analyses will enable aggressive scaling of supply voltages and consequently power reduction, which have hitherto not been done due to the lack of visibility into variations.

1. Conclusions

We have found that feed forward neural networks can accurately model a digital logic gate's delay and output slew. The model can incorporate a gate's per-transistor parameters and hence allow for considering intra-gate variations. They also include output loads, input slews, temperature and supply voltage, all within a single model, unlike in conventional table based models. Non-linear relationships are also accurately captured and hence allow the model to be used in deep submicron regimes which have large variations. The inclusion of supply voltage enables new applications like statistical analysis in DVS framework. We have created delay and output slew models for an inverter, 2-input NAND and NOR and 3 input NAND and NOR gates. The model has 12 hidden nodes for modeling the delay and output slew within 2% of SPICE. The PDFs for

delay and output slew generated from the models match very closely to that from SPICE across a large supply voltage range of 0.5V to 1.3V, yet quite faster than SPICE. As an example, we have used these models in a Statistical Timing Analysis engine to obtain optimum voltages for target delays with specified statistical guarantees, for a DVS framework.

While this approach looks quite promising, the main limitation of neural network based models is their computational complexity compared to simpler linear models, though the latter sacrifices significant accuracy for speedup and might not be suitable for large variations in the deep submicron regime.

References

[1]S. Borkar et. al., "Parameter Variations and Impact on circuits and Microarchitecture," *DAC*, pp. 338-342, June 2-6, 2003.

[2]K. A. Bowman et. al., "Impact of Extrinsic and Intrinsic Parameter variations on CMOS System on chip Performance," *IEEE*, pp.267-271, 1999.

[3]S. Nassif et. al., "Within-chip variability analysis," *IEDM Tech. Digest*, pp.283-286, Dec. 1998.

[4]K. Okada, K.Yamaoka and H.Onodera, "A Statistical Gate-Delay Model considering Intra-gate Variability," *ICCAD*, pp.908-913, 2003.

[5]A. Devgan et. al., "Block-based Timing Analysis with Uncertainty," *ICCAD*, pp. 607-614, Nov. 2003.

[6] H.Chang et. al., "Parameterized Block-Based Satistical Timing analysis with Non-Gaussian Parameters, Nonlinear Delay Functions," *DAC*, pp. 71-76, June 2005.

[7]H. Chang and S.S. Sapatnekar, "Statistical Timing analysis considering spatial correlations using a single PERT-like Traversal," *ICCAD*, pp.621-625, Nov., 2003

[8]J. J. Liou et.al, "Fast Statistical Timing Analysis by Probabilistic Event Propagation", *DAC* pp. 661-666, June 2001.

[9]C. Visweswariah, et. al , "First-order Incremental Block-Based Statistical Timing Analysis", *DAC* , pp. 331-336, 2004.

[10] S.R. Nassif, A.J. Strojwas and S.W. Director, "A methodology for worst-case analysis of integrated circuits," *IEEE TCAD*, Vol CAD-5, no.1, pp. 104-113, Jan 1986.

[11]S.J. Lovett, M. Welten, A. Mathewson and B. Mason, "Optimizing MOS Transistor Mismatch", *IEEE JSSC*, Vol. 33, No.1, pp. 147-150, Jan., 1998.

[12]M.J.M. Pelgrom et.al, "Matching properties of MOS Transitors," *IEEE JSSC*, Vol. 24, No.5, pp. 1433-1440, Oct., 1989.

[13]S. Haykin, "Neural Network A comprehensive foundation," PHI, New Delhi, 1999.

[14]G. Wolfe, R. Vemuri, "Extraction and Use of Neural Network Models in Automated synthesis of Operation Amplifiers," *IEEE TCAD*, Vol.22, No. 2, Feb. 2003.

[15]V.K. Devabhaktuni et. al., "Neural Networks for Microwave modeling: Model development issues and nonlinear modeling techniques," *Int. J. RF Microwave Computer-aided Eng.* , vol.11, No. 1, pp4-21, Jan 2001.

[16]Calhoun, B.H., Chandrakasan, A.P, “Ultra-dynamic Voltage scaling (UDVS) using sub-threshold operation and local Voltage dithering”, *IEEE JSSC*, Volume 41, Issue 1, pp.238 – 245, Jan 2006.