Leakage Modelling Of Logic Gates Considering The Effect Of Input Vectors

Janakiraman V[1], Bishnu Prasad Das[1], Vish Vishvanathan[2], Bharadwaj Amrutur[1]

Abstract

In this paper we present an accurate model for the leakage current of a logic gate using Neural Networks. Our model captures the effect of process variations, which include inter die and intra gate variations and the effect of supply voltage and hence it provides a very efficient tool for analysis of circuits that implement Dynamic Voltage Scaling (DVS). Further, for a particular gate, importance of leakage dependence on the input vector is brought out. We then present a technique to logically reduce the number of models by modelling different kinds of stacks present in the library. Results show that the neural network can model the distribution of the leakage current of gates to a high degree of accuracy with the error in mean being less than 1% and variance being less than 4%. It was found that Monte Carlo on our model was up to 150 × faster than Monte Carlo with SPICE.

Keywords: *Neural*, *Leakage, Modelling, Statistical, Sigmoid*

1. Introduction

Statistical analysis has taken over static analysis in digital design in nano-scale circuits, owing to the increasing impact of manufacturing variations. Statistical timing analysis has been studied in great depth for a considerable amount of time 5, but only recently has statistical leakage analysis gained some attention. As predicted in [1] leakage power will be a major contributor to the total power and shrinking transistor sizes makes this leakage power more difficult to predict. Variations in effective gate length, L_e, oxide thickness, T_{ox}, and threshold voltage V_{TH} can result in up to 20× variations in the leakage of the manufactured chips [2].

In the past few years considerable amount of work has been done in building models that predict leakage accurately. The empirical technique described in [3] captures the effect of variations in the effective gate length L_e and provides a simple analytical method to statistically analyze leakage but cannot handle variations in multiple parameters. [4] provides a generic framework to handle variations in multiple process/ random parameters but does not include deterministic variations like supply voltage. With extensive usage of Dynamic Voltage Scaling (DVS), a model that can capture variations for a range of voltages is very useful.

Another area of interest in the past few years has been the dependency of leakage current of a particular gate on the input vector combination. The most

[1] Department of Electrical Communication Engineering, Indian Institute of Science, Bangalore. jramaanv@gmail.com, bpdas@cedt.iisc.ernet.in, amrutur@ece.iisc.ernet.in

[2] Texas Instruments Pvt Ltd, Bangalore: vish@ti.com

obvious and straight forward technique is to index the leakage models by input vector. Another method is to assume average value of leakage current over all possible input vectors 5, which is highly inaccurate since all input vectors do not occur with equal probability. [5] proposes an analytical technique to estimate the leakage current of stacks based on the underlying device physics equations. The model was done for the static case and assumes that an NMOS transistor is OFF iff its gate voltage is zero which is not true as will be explained later. This leads to significant errors in predicting leakage.

In this paper, we propose a technique that empirically captures the effect of underlying process variations and additionally captures the variations in supply voltage/temperature as well. We use Neural Networks [7] to empirically model the leakage current. [8] used Neural Networks to model delay of gates; we extend the idea to model a gate's leakage current. Further we propose a technique to reduce the number of models required to capture the dependency of the leakage current on the input vector.

The rest of the paper is organized as follows: The following section describes the components of leakage and the effect of variations on them. Section 3 describes the neural network models used to model the sub-threshold leakage current of a gate. In section 4 we describe a method to reduce the number of models required to model a gate completely for all possible input combinations. Our conclusions are presented in section 5.

2. Impact of process, environmental and input vectors on leakage

The major components of leakage in CMOS circuits are broadly classified as [11], sub-threshold leakage, gate leakage and band to band leakage.

2.1. Components of leakage

Sub-threshold leakage current varies exponentially with V_{TH}. Scaling in recent times has forced V_{TH} to come down drastically and hence increased leakage by orders of magnitude.

Gate leakage results from electrons tunnelling through the oxide. The probability of electrons tunnelling through the oxide increases with decreasing oxide thickness. Expressions for Gate leakage and sub-threshold current can found in [11, 12]. Band to band tunnelling is also becoming a significant leakage component due to heavy doping in the source and drain regions.

A study from Intel Corporation shows that leakage power will contribute approximately 50% of the total power in the 90nm technology node [3]. It is thus important to accurately model leakage power. Further, leakage power depends exponentially on certain device parameters and hence very sensitive to process, temperature and supply voltage variations. With growing uncertainty in process parameters accurate estimation of leakage current becomes challenging.

2.2. Process Variations

2.2.1. Inter-die, Intra-gate and Intra-die variations

Inter-die variations [10] refer to variations that occur across dies, wafers or lots. The variation introduced is the same across the entire die. These variations include gate-length variations due to fluctuations in the time of exposure during fabrication. Intra-die [10] variation, as the name suggests, refers to variation of a particular parameter within the die. Intra-die variations are usually spatially correlated and all transistors within a gate have the same variation. Intra-gate 5 variations refer to transistor to transistor variations within a gate. The impact of intra-gate variations on statistical analysis is a drastic increase in the number random variable needed to model them. Each transistor within a gate will now have one random variable per intra-gate process parameter. A process parameter considering all types of variations can be expressed as

$$P_{ij} = P_{nom} + \Delta\ P^{inter\text{-}die} + \Delta P^{intra\text{-}die}{}_{i} + \Delta\ P^{intra\text{-}gate}{}_{ij} \ldots\ldots.. (1)$$

Where P_{nom} is the nominal value of the process parameter. $\Delta P^{inter\text{-}die}$ is a zero mean Random Variable (RV) which captures the effect of inter die variations. The zero-mean RV, $\Delta P^{intra\text{-}die}{}_{i}$ models the intra die variation for the i^th gate and $\Delta P^{intra\text{-}gate}{}_{ij}$ captures intra gate variation of the j^th transistor of the i^th gate.

Our work is addressed towards modelling leakage for a logic gate and currently accounts for inter-die and intra-gate variations only. It can easily be extended to handle intra-die variations also, provided they can be de-correlated using techniques like PCA or KLE [4]. The de-correlated intra-die parameters are then treated as inter-die parameters at gate level.

To illustrate the impact of inter-die and intra-gate variations on the leakage current, a NAND4 gate has a nominal leakage of 95.7 pA in a 130 nm process. With an intra-gate variation introduced in the VT0 parameter, with 3σ equal to 10% of its mean, there is a spread in the leakage current with mean 94.8 pA and standard deviation 10.4 pA. When an inter die variation is introduced in the effective gate Length (L_e), A 3σ of 5% of the nominal gate length results in a distribution of the leakage with mean 94.7 pA and standard deviation of 1.5pA.

2.2.2. Voltage and Temperature

Leakage is a strong function of voltage and temperature. The leakage of a four input NAND gate with an input "0011" varies from 204 pA to 250 pA when the voltage changes from 0.6 V to 1.2 V. Similarly, leakage is a strong function of temperature too. Afore mentioned leakage currents were measured at room temperature i.e. 25^0 C. Under otherwise identical conditions, a leakage of 250 pA through a four input NAND gate, at 25^0 C, can shoot up to 3.14 nA at 100^0 C. It is clear from the above data that to be able to predict leakage accurately, we need a model that can capture the dependence of leakage on inter die process parameters, intra gate process parameters, voltage and temperature. In the current literature no model can capture the effect of all these parameters together.

2.2.3.*Input Vector*

Another important factor that the leakage depends on is the input vector. Shown in Table. 1 is the leakage current of a four input NAND gate for all possible input combinations. The lowest leakage is obtained when the input is "0000" (88 pA) and the highest is when the input is set to "1110" (719.5 pA). Up to 10× increase in leakage can be obtained by simply changing the input to the gate. It is thus extremely important to consider the effect of the input vector and build models accordingly.

Input	**I_{SUB} (pA)**	**Input**	**I_{SUB} (pA)**
0000	**88**	0001	123.3
0010	124.1	0011	204.2
0100	124.0	0101	205.2
1000	124.0	1001	205.1
0110	210.4	0111	591.6
1010	210.3	1011	602.4
1100	210.3	1101	621.5
1110	**719.5**	1111	519.7

Table 1: NAND4 Leakage vs Input vector

3. *Leakage modelling - Neural Networks*

Neural networks have been used for over a decade in pattern recognition applications. The ability of a neural network to model complex systems, which are dependent on a number of parameters, makes it a good candidate for statistical modelling. It is well known that any continuous function f(**x**), where **x** is an input vector, can be modelled very well by a neural network with a single hidden layer [7] even for a non-linear functions. The BSIM3 equation for the leakage current is a smooth continuous, but non linear, function of the underlying process parameters and hence it is expected that the leakage current of a gate can be modelled well by a neural network. In [4] the dependence of the leakage current on the random process parameters is modelled very well by the polynomial chaos expansion. However it does not capture the dependence on the supply voltage/ temperature. [9] models the dependence of temperature and voltage accurately but does not capture process dependent parameters. Modelling leakage with neural networks has the advantage of capturing variations of process parameters and supply voltage together making it a suitable model when used for statistical analysis along with DVS.

An Artificial Neural Network (ANN), simply known as a neural network, has one input layer, few hidden layers and an output layer [7]. In this work, we have a used an ANN with a single hidden layer as shown in Figure.1. Further details about an ANN with a single hidden layer can be found in [7].

3.1. *Training and Testing the Neural Network*

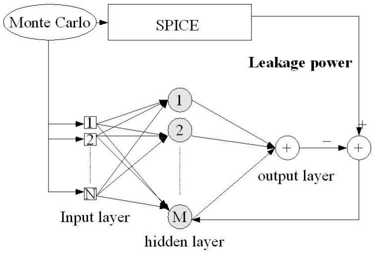

Figure 1: Training a Neural Network with a single hidden layer

The basic structure of the training algorithm is shown in Figure 1. Data is generated through Monte Carlo simulation in SPICE and then a fraction of the generated data is used to train the ANN. We use the standard back propagation algorithm [7] for training. We train the ANN till the error is within acceptable limits. Depending on the implementation of the back propagation algorithm, repeated training of the ANN can produce different results. Thus it is important to stop the training once an acceptable error has been obtained. During the process of training, all the weight matrices are updated. Once the training is done, these values are fixed for a given gate. Having trained the ANN, it needs to be tested with a different data set. As mentioned earlier, we used a fraction of the SPICE data for training the ANN. We therefore use the rest of the data to test it. If the error between the SPICE leakage and the one predicted by the ANN is not acceptable we have to re-train the ANN either by changing the number of hidden neurons or simply starting with a different initial weight vector. In our case the latter was used to re-train the ANN. It was, however, observed that amongst the gates that we trained we did not have to re-train the ANN for any gate more than twice.

3.2. Results

We have modelled the leakage current of a few basic gates for all possible input combinations. The BSIM3 leakage model of an industrial 130nm technology was used with HSPICE to generate Monte Carlo data samples. MATLAB's in built ANN toolbox was used to train and create the Neural Network model. We used the tan-sigmoid function as the ANN's activation function. The Tan-Sigmoid function is given by

$$\varphi\ (x) = \tanh(x)$$

Simulations were carried out on a machine with Intel P4 processors and 512 MB RAM.

3.2.1.*Supply voltage variation*

Our models take into account intra gate variations in three process parameters, namely, effective gate length (L_e), oxide thickness (T_{ox}) and threshold voltage (V_{TH}) of the transistor. Further we have assumed an independent inter die variation in the gate length. Thus if a gate consists of N transistors, the ANN will have to capture the effect of 3N+1 random parameters along with the effect of - supply voltage (V_{DD}). According to ITRS [1], all random parameters have a 3σ variation of 10% of their mean. While the nominal supply voltage is 1.2 V we consider an extended supply range of 0.6 to 1.2V to model the gate in order to support DVS. During the testing phase the supply voltage is held constant at some desired value and all other parameters vary according to their distributions.

The input and output data were normalized before training the ANN. If X is a parameter (either input or output) then the data was normalized according to

$$X_{NORM} = X/[\max(X) - \min(X)]$$

Thus the ANN was trained with X_{NORM} as its input/ output. The number of hidden Neurons in all our ANN's is *I+1* where *I* is the number of inputs to the ANN.

The training and testing data set was generated through 5000 SPICE Monte Carlo simulations and we used 30% of them for training the ANN and the rest for testing the model. All the results presented here are on the testing data. The training and testing data sets are disjoint. While training the ANN, all random parameters vary according to the distribution they come from, which we have assumed as Gaussian. But the supply voltage varies from 0.6 V to 1.2 V in steps of 0.4 mV. Fig 2 shows the actual and predicted leakage current probability distribution functions(PDF) of a four input NAND gate for two different supply voltages (0.6 V and 1.2 V). The input vector was set to "0011" and at a temperature to 25^0 C. A more detailed result for other input vectors is presented in section 5.

Including both voltage and temperature in the model introduces unacceptable error and we either include voltage or temperature into the model. However, it is not necessary to model temperature over a continuum of values. The area on a chip can be divided into a few regions, in each of which the temperature is essentially constant. Areas where there is a lot of activity will result in high temperatures and other areas will be at much lower temperature. Thus we would only encounter a small and finite set of temperature regions for which we can have models indexed by the temperature. However, we could model the effect of temperature and index by voltage if the afore mentioned assumption does not hold. The next result shows that the ANN is able to capture variations in process and temperature with supply held constant.

3.2.2. *Temperature variation*

All process parameters (L_e, T_{OX}, V_{TH}) varied according to Gaussian distributions with 3σ equal to 10% of their mean. Like in the previous model an independent inter die Gaussian variation was introduced in the gate length. Here the supply voltage was held constant at 1.2 V and the temperature was varied from $0\text{-}100^0$ C in steps of 0.067^0C. The training and testing phase are identical to the previous section. Fig. 2 shows the actual and predicted PDFs of a NAND4 gate at four different temperatures (25^0 C, 50^0 C, 75^0 C, 100^0 C). The input to the NAND4 gate was "0000" and the supply was set to 1.2 V.

It is interesting to note that the effect of temperature is greater on leakage than supply voltage for the 130 nm process. Yet we present our model with voltage as a part of it and not temperature. We only present a technique to model variations along with voltage. Depending on the application and accuracy required we could choose to model it with either voltage or temperature.

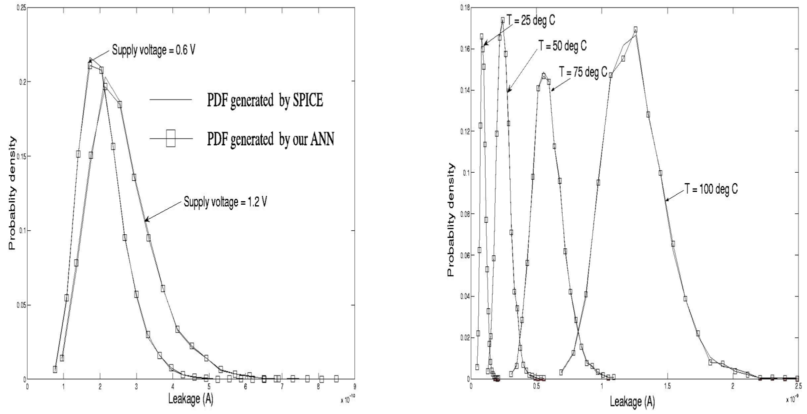

Figure 2: Comparison of the PDF generated by SPICE and our ANN for a NAND4 gate at (i) 0.6 V and 1.2 V , (ii) 25^0C, 50^0C, 75^0C and 100^0C

4. Modelling Stacks

As mentioned earlier, an order of magnitude increase in leakage current can be obtained just by changing input vector. Thus a model built to predict the leakage for a particular input I_1 may not accurately predict it for another input I_2. The most obvious way out of this is to have one model per input vector per gate. Thus with a NAND4, NAND3 and NAND2 gate we require 28 models in all to predict the leakage of the three gates for any input combination. We now present a technique to reduce the required number of models. We explain the principle with the help of a four input NAND gate.

Consider a four input NAND gate shown in Fig. 4. With 4 inputs, we have a total of 2^4 input combinations. With inter die and intra gate variations that we have considered, each input vector can result in a different distribution of leakage current. To model the effect of input vectors, we need a generic model that can be used across gates so that the number of models reduce. We can achieve this by modelling different kinds of stacks. We explain the procedure of using stack models to predict the leakage of a NAND4 gate. Similar arguments can be used for other gates.

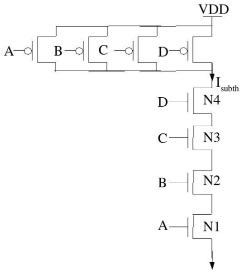

Figure 3: A Four Input NAND gate

Before we proceed, it is necessary to define two new terms for sake of clarity. A *static* transistor is a transistor whose parameter variations (L_e/ T_{OX}/ V_{TH}) do not result is a *significant* change in the PDF of the leakage current while a *statistical* transistor is one whose parameter variations result in a significant change in the PDF of the leakage current. We now consider all possible input combinations and explain the technique to reduce the number of models required.

4.1. *Input vector(0000)*

In this state all four NMOS transistors are turned OFF while all four PMOS transistors are turned ON and are as good as short circuits [5]. As far as statistical leakage modelling is concerned it is a valid assumption to treat these PMOS transistors as static transistors and the four NMOS transistors as statistical transistors because the leakage current is determined by one or more OFF transistors on the stack. The ON transistors have sufficient gate overdrive that any variations in their parameters cannot affect the leakage. Thus to predict the leakage PDF of a NAND4 gate, for the input vector 0000, we need to model a four transistor NMOS stack, which we will refer to as ***n4/0***. The "0" in the model name refers to the input vector applied to the stack. In this paper we will treat the input to the transistor which closest to the output as the LSB and the one closest to V_{DD}/GND as the MSB.

4.2. *Input vectors (1000 / 0100 / 0010)*

Here the top most transistor, N4, has its gate connected to gnd and the one of the other 3 NMOS transistors have their gates connected to V_{DD}. At least one PMOS transistor is completely turned on in all three cases and hence forces the other three parallel PMOS transistors to behave as static transistors. Similarly the one NMOS transistor that has its gate connected to V_{DD} behaves as a short circuit and can be treated as a static transistor. Thus to predict the leakage for this set of input combinations we need to model a 3 transistor NMOS stack with all inputs grounded i.e. we need to model an ***n3/0***. However we have to scale the currents by an appropriate factor as the widths of the transistors are lesser for the n3/0 model when compared to the widths of the NAND4 gate. This scaling factor can be obtained by simply measuring the leakage current through the NAND4 gate with its input set to "1000" and dividing it by the leakage current that flows through the three transistor NMOS stack with its input set to "000". Both currents have to be measured with all other process parameters set to their nominal values.

4.3. *Input vector (0001)*

With the gate of the top most transistor N4 connected to V_{DD} [5] treats N4 as a short circuit. Unfortunately the N4 is trying to pass V_{DD} and it will *turn off* as soon as the source charges up to V_{DD}- V_{TH}. We are thus left with a case where the bottom three transistors are turned off as their gates are grounded and the top most transistor is turned off even though its gate is connected to V_{DD}. Hence we need to model a four transistor NMOS stack with the gate of the top most transistor connected to V_{DD} which will be called as ***n4/1***.

4.4. *Input vectors (1100 / 1010 / 0110)*}

As predicted by [5] the two transistors with their gates connected to V_{DD} can be treated as short circuits and hence as static transistors which reduce the NMOS stack to a 2 transistor NMOS stack ***n2/0***. Accounting for the difference in the widths of the 2 transistor NMOS stack and the NAND4 gate we can scale the currents predicted the n2/0 model to predict the NAND4 leakage for these input vector combinations. The scaling factor is obtained as mentioned in the earlier case.

4.5. Input vectors (1001 / 0101 / 0011)

Just as in the 0001 case we observe that the top most transistor is turned off even though its gate is connected to V_{DD}. Hence we need to have a model ***n3/1*** with appropriate scaling to predict the leakage for this combination.

4.6. *Input vectors (0111 / 1011 / 1101)*

The leakage current is dominated by a single transistor whose gate is connected to ground. We would expect the model ***n2/1*** to predict the leakage for these input combinations. But unfortunately the DIBL effect on the transistor whose gate is grounded is different in all three cases and hence there is a significant deviation in the PDF predicted by the n2/1 model. As can be seen from Fig 4, transistor N1 dominates the leakage in the 0111 case. Let the drain source voltage across it be V_{DS1}. In the 1011 case N2 dominates the leakage, let the drain source voltage across N2 be V_{DS2}. Similarly the leakage in the 1101 case is dominated by N3 and let the drain source voltage across N3 be V_{DS3}. In the 0111 case all four transistors are OFF and the V_{DD} drop is across all four transistors. In the 1011 case the bottom transistor, N1, is ON and is essentially a short circuit and hence the V_{DD} drop is across three transistors (N2, N3 and N4), similarly, in the 1101 case the V_{DD} drop is across N3 and N4 alone. Hence $V_{DS1} < V_{DS2} < V_{DS3}$, which implies that the effect of DIBL increases as the OFF transistor moves up the stack i.e. $I_{0111} < I_{1011} < I_{1101}$ as can be seen from Table. 1. We, therefore, need a separate model for these three input vectors.

A NAND4 gate behaves like an inverter with its input set to “111X” (X =0 or 1). Hence its leakage corresponding to the input vectors “1110” and “1111” can be predicted using an ANN model that predicts leakage of a unit inverter.

From the above discussion it is clear that we need to model all kinds of stacks for a selected few input combinations to be able to predict the leakage of any gate. A NAND3 and NAND2 gate will re-use most of the stack models to predict their leakage currents. For example, a NAND3 gate with its input set to "100" and a NAND2 gate with its input set to "00" will use the same n2/0 stack model that the NAND4 used when its input is set to "1100". Similar arguments will explain the re-use of stack models for other input vectors as well. PMOS stack models can similarly be used to predict leakage of NOR gates.

4.7. *Results*

Shown in Table. 2 is a comparison between the results obtained from HSPICE and that predicted by our ANN model different gates two input vectors i.e. the input vector that gives minimum and maximum mean and std dev error. Our

ANN model captures all process variations and supply voltage (0.6 - 1.2 V). All the results presented are for a supply voltage of 0.9V and a temperature of 25^0 C. The columns in Table 2 have the following meaning

- Gate – Name of the gate
- Input - Decimal value of Input to the gate
- Model - The stack model used to predict the leakage
- μ_{SP} - Sample mean obtained from SPICE simulation
- μ_{ANN} - Sample mean obtained from our ANN model
- σ_{SPICE} - Sample standard deviation obtained from SPICE simulation
- σ_{ANN} - Sample standard deviation obtained from the ANN model
- $\Delta\mu$ - Percentage error between the μ_{SP} and μ_{ANN}
- $\Delta\sigma$ - Percentage error between the σ_{SP} and σ_{ANN}

The ANN stack models predict the mean and standard deviation of a gate for a particular input vector combination very accurately. A single stack model, say n2/0, is able to predict the leakage current PDF accurately for 6 different cases i.e. NAND4-(1100, 1010, 0110), NAND3-(100, 010) and NAND2(00). For a set of NAND gates (4, 3 and 2 input) we trained a total of 8 different stack models to predict the leakage PDFs for 22 different input vectors. From Table 2 we can see that the error in μ and σ is greater when the two stack models are used to predict the leakage for a 4 input gate. This is expected; as we are trying to predict the leakage distribution caused by variations in 16 transistors by considering variations in just two transistors. But, the error is very much within acceptable limits. Similar interpretation can be given for the other gates and models as well. It was found that Monte Carlo on our model of a NAND4 gate was 150 × faster than Monte Carlo with SPICE.

5. Conclusion

The Neural Network model captures the variations in all process parameters and is able to predict the PDF of the leakage current accurately. The predicted mean and standard deviation was erroneous by less than 1% and 4% respectively. Further it is also able to handle dependency of the leakage current on supply voltage, which not only makes the ANN model a good statistical model but also a good candidate as a voltage scalable model. We further established that modelling all kinds of stacks can accurately predict the PDF of a gate for any given input combination, there by reducing the required number of models. The ANN is 150 × faster than SPICE and hence a valuable model to use.

REFERENCES

[1] International technology roadmap for semiconductors. 2003. February 2003

[2] Sherkar Borkar et al (2003). *Parametric variations and impact on circuits an dMicro architecture.* In Proc. DAC 2003.

[3] R. Rao, A. Srivastava, D. Blaauw and D. Sylvester (2004). *Statistical analysis of sub-threshold leakage current for VLSI circuits* IEEE. Trans. Very Large Scale, 12(2);131-139, 2004.

[4] Sarvesh Bharadwaj and Sarma Vrudhula (2007). *A Fast and Accurate approach for Full Chip Leakage Analysis of Nano-scale circuits considering Intradie Correlations* Conference Proceedings. 20th VLSI Design,January 2007.

[5] Z. Chen, M. Johnson, L. Wei and K Roy (1998). Estimation of standby leakage power in CMOS circuits considering accurate modelling of transistor stacks In *ISLPED '98: Proceedings of the 1998 international symposium on Low power electronics and design*, pages 239-244. New York, NY, USA, 1998. ACM Press.

[6] Wen-Tsong Shiue (2001). Leakage power estimation and minimization in VLSI circuits IEEE, 2001.

[7] Simon Haykin (1998) *Neural networks a comprehensive foundation*, 2 edition Prentice-Hall Engineering, USA, 1998

[8] Bishnu Prasad Das, Bharadwaj Amrutur, H.S.Jamadagni (2006). *Critical path modelling for dynamic voltage scaling (DVS) in low power applications* VDAT, 2006.

[9] H.Su, E. Acar and S. R. Nassif (2003). Full chip leakage estimation considering power supply and temperature variations In *ISLPED '03: Proceedings of the 2003 international symposium on Low power electronics and design*, pages 78-83. New York, NY, USA, 2003. ACM Press.

[10] A. Srivastava, D. Sylvester and D. Blaauw (2005) *Statistical analysis and optimization for VLSI: Timing and Power*, Springer, USA, 2005

[11] K Roy, S Mukhopadhyay and Hamid Mahmood (2003) *Leakage current mechanisms and leakage reduction techniques in deep sub-micrometer CMOS circuits* Proceedings of the IEEE, Vol 91, No.2, February 2003

[12] Chandrakasan, W. Bowhill and F. Fox (2001). *Design of high-performance microprocessor circuits.* IEEE press, 2001

[13] Ashish Dobhal, Vishal Khandelwal, Azadeh Davoodi and Ankur Srivastava (2007), *Variability Driven Joint Leakage-Delay Optimization Through Gate Sizing with Provabale Convergence*, Conference proceedings VLSI Design, 2007

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

Gate	**I/p**	**Model**	**μ_{SP} (pA)**	**μ_{ANN} (pA)**	**$\Delta\mu$ (%)**	**σ_{SP} (pA)**	**σ_{ANN} (pA)**	**$\Delta\sigma$ (%)**
NAND4	0*	*n4/0*	*90.51*	*90.51*	*0.00*	*20.46*	*20.44*	*0.09*
NAND4	13	n2/1	1015.87	1028.02	1.2	387.74	401.60	3.5
NAND3	0*	*n3/0*	*89.01*	*88.98*	*0.04*	*23.18*	*23.17*	*0.05*
NAND3	5	n2/1	556.21	558.77	0.46	222.66	228.31	2.54
NAND2	0*	*n2/0*	*98.86*	*98.86*	*0.01*	*32.82*	*32.81*	*0.05*
NAND2	1	n2/1	349.24	349.20	0.01	140.36	140.69	0.23
NOR4	2	p2/2	1457.62	1445.64	0.82	700.33	694.72	0.80
NOR4	***15***	*p4/15*	*98.45*	*98.46*	*0.00*	*24.78*	*24.77*	*0.05*

	*							
NOR3	2	p2/2	711.91	710.41	0.21	341.42	342.33	0.27
NOR3	7*	p3/7	96.83	96.83	0.00	27.91	27.94	0.10
NOR2	2	p2/2	430.15	403.22	0.02	190.91	191.56	0.34
NOR2	3*	p2/3	103.91	103.88	0.03	37.69	37.75	0.15

Table 2: Mean and Std Dev – Best and Worst case error details.

*** Input vector for which mean and std dev error is minimum.**