Efficient CMOS subthreshold leakage analysis with improved stack based models in presence of parameter variations

L. Garg and V. Sahula

Presented is the error that occurs while estimating subthreshold leakage power of parallel transistor stacks in CMOS gates using leakage power models when there is no consideration of the manufacturing variations, i.e. device geometry related effects in width. For the purpose, efficient support vector machine based macromodels for characterising the transistor stacks of CMOS gates are reported, considering process parameter variations impacting e.g. length, threshold voltage, oxide thickness, supply voltage, temperature and width of the transistors. The experiments show that maximum error can go up to $\sim 15\%$ for AOI22 and OAI22 gate under nominal values of varying parameters without considering manufacturing variations in the width.

Introduction: In digital circuits, subthreshold leakage power is one of the important parameters to be accurately estimated and minimised. Subthreshold leakage power of a gate can be modelled as a function of the width of transistors on a stack (W), global process (GP) and local process (LP) parameters, supply voltage (V_{DD}), temperature (T), input vector, and body bias voltage of transistors. If there are M gates and the ith gate has k_i inputs, then a total of $\sum_{i=1}^{M} 2^{k_i}$ leakage model instances are required. Our methodology is based on first characterising the basic stacks among various CMOS gates and then estimating the leakage power under different input vector combinations using basic stack models. However, CMOS gate characterisation is a one-time effort; the stack modelling approach helps in increasing the accuracy but at the cost of increased characterisation time. Viraraghavan [1] used the neural network to model the leakage power using the transistor stacks. One of the main drawback of this methodology is that the parallel transistors with the same inputs cannot be combined, which increases the number of models to be characterised. Al-Hertani *et al.* [2] modelled the leakage power of stacks by taking the width as a varying parameter, which allows the model to calculate the leakage power of the CMOS gates containing parallel transistors with the same inputs on the parallel transistors. The drawback of this model is that it combines the parallel transistors replacing each such group by a single equivalent transistor of effective width. The effective width is computed by simply adding the width of parallel transistors in a group. Our methodology is also very useful in leakage power estimation of the circuits in [3], where the single transistor is broken down into many parallel transistors for the realisation of faster digital circuits. In our approach, we propose to consider the effective width of the so formed parallel transistors. Our experimental results show that, without considering the effect of manufacturing variations on width, the errors can be very high. The final model can be represented in process-voltage-temperature-width space by

$$Y_i^v = f_i^v(W_i, GP, LP_i, T, V_{DD}) \tag{1}$$

Stack modelling methodology: We have used conventions as in [1] for labelling stack parameters, which are reproduced as follows: {stack-type}{stacksize}/{input to the stack}. Here, stack type indicates whether it is an NMOS stack or a PMOS stack. Stack size represents the number of transistors on a stack being considered. The LSB of the input vector is applied to that transistor, which is closest to the output. Basic stacks can be extracted based on the following rules:

1. If the parallel transistors are supplied with the same inputs, then the effective width is calculated; the parallel transistors are replaced with a single transistor having equivalent effective width; and rules 3 and 4 as follows are applied.
2. If different inputs are supplied, then the 'OFF' transistor is removed because the 'ON' transistor makes the drain and source voltages equal, thus nullifying the effect of the 'OFF' transistor; and then rules 3 and 4 as follows are applied.
3. If the number of 'ON' transistors is greater than the number of 'OFF' transistors, then (i) a separate stack model is required per stack type per input vector; else (ii) the transistors applied with 1/0 input except the transistor closest to the output for the NMOS/PMOS stack are removed and then a separate stack with that input vector is built and modelled for this case.
4. Leakage due to parallel stacks simply adds up.

Fig. 1 shows the basic NMOS stack models. Similarly, PMOS stack models can be derived.

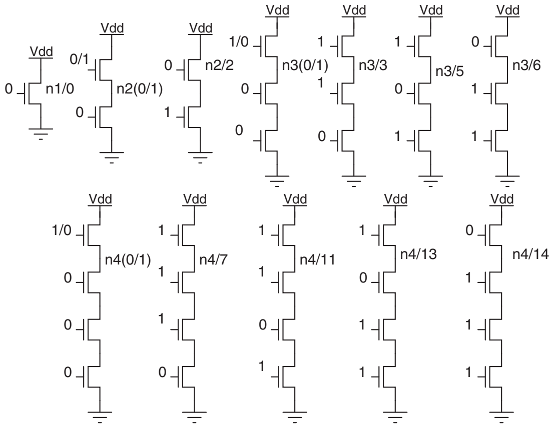

Fig. 1 *Basic NMOS stack models*

Effective width estimation of parallel transistors with same inputs: The subthreshold leakage current of a transistor can be represented by (2) as in [4]:

$$I_{sub} = \mu_N C_{ox} \frac{W}{L} V_t^2 \left[\frac{V_{gs} - V_{th}}{nV_t}\right]\left[1 - \exp\left[-\frac{V_{ds}}{V_t}\right]\right] \tag{2}$$

From (2), it can be observed that I_{sub} is directly proportional to the width of the transistor. But this width is an effective width which is actually calculated according to the device, i.e. BSIM4 equations, as follows.

The effective width can be represented in terms of drawn width W_{drawn} as in (3), (4) and (5):

$$Weff = \frac{W_{drawn}}{NF} + XW - 2dW \tag{3}$$

$$\begin{aligned} dW &= dW' + DWG.V_{gsteff} \\ &\quad + DWB\left(\sqrt{\phi_s - V_{bseff}} - \sqrt{\phi_s}\right) \end{aligned} \tag{4}$$

$$dW' = WINT + \frac{WL}{L^{WLN}} + \frac{WW}{W^{WWN}} + \frac{WWL}{L^{WLN}W^{WWN}} \tag{5}$$

Here, NF = number of device fingers, XW = parameter to account for channel width offset due to mask/etch effect, DWG, DWB = to account for the contribution of both gate and substrate effect. $WINT$, WL, WW, WWL, WLN, WWN = model parameters to describe the dependence of dW on device geometry. $WINT$ is calculated in the traditional manner from which 'delta W' is extracted, from the intercept of straight lines on a $1/R_{ds}$–W_{drawn} plot.

In general, $WL = WW = WWL = 0$, $WLN = WWN = 1$, which simplifies (4) and (5) into (6) and (7):

$$dW' = WINT \tag{6}$$

$$\begin{aligned} dW &= WINT + DWG.V_{gsteff} \\ &\quad + DWB\left(\sqrt{\phi_s - V_{bseff}} - \sqrt{\phi_s}\right) \end{aligned} \tag{7}$$

Since in our model we are focusing only on parameter variation in length, threshold voltage, oxide thickness, supply voltage, temperature and width of the transistors, the parameters related to gate and substrate effects can be assumed to be zero, i.e. $DWG = DWB = 0$, which results in a simplified expression for (7) as in (8):

$$dW = WINT \tag{8}$$

From (8) and (3), we get

$$Weff = \frac{W_{drawn}}{NF} + XW - 2WINT \tag{9}$$

In our case, we choose the number of fingers, $NF = 1$ and $XW = 0$, the final value of $Weff$ can be represented by

$$Weff = W_{drawn} - 2WINT \tag{10}$$

Now, let us consider that the two transistors have width w_1 and w_2 with identical potential at respective terminals and also have other parameters i.e. L, V_{th}, T_{ox} etc. identical. These transistors can be replaced by a single transistor with equivalent width as $w_{eq,\ drawn}$, which can be calculated as follows:

$$w_{eq,eff.} = w_{1,eff} + w_{2,eff} \tag{11}$$

$$\begin{aligned} w_{eq,drawn} - 2.WINT = w_{1,drawn} - 2.WINT \\ + w_{2,drawn} - 2.WINT \end{aligned} \tag{12}$$

$$w_{eq,drawn} = w_{1,drawn} + w_{2,drawn} - 2.WINT \tag{13}$$

For N parallel transistors, this can be generalised as follows:

$$\begin{aligned} w_{eq,drawn} &= w_{1,drawn} + w_{2,drawn} \\ &\quad + \ldots - 2.N.WINT + 2.WINT \\ &= w_{1,drawn} + w_{2,drawn} + \ldots - 2.(N-1).WINT \end{aligned}$$

Results: We have used 45 nm predictive technology mapping (PTM) model file for all simulations presented in this Letter. We use LS-SVM toolbox in MATLAB for support vector machine (SVM) training. Inter-die and intra-die variations were considered on three process parameters, length (L), threshold voltage (V_{th}) and oxide thickness (T_{ox}), with Gaussian distribution ($3\sigma = 10\%$). Supply voltage (0.6–1.2 V) and temperature (0–100°C) and width (45–200 nm) were sampled with uniform distribution. RBF kernel is used with 5000 training and 500 testing samples to accurately train and test SVM-based leakage power models. We simulated the AOI22 gate and OAI22 gate across all input vector combinations in order to compare the accuracy with the approach in [2]. Table 1 shows that the training and testing correlation coefficient is greater than 0.99 which shows the higher accuracy of our model. Significant improvement in runtime using our model has been achieved and is compared to SPICE-runtime for 5000 Monte Carlo simulations. Table 2 shows that our approach has less than 0.5% error accross all input vector combinations compared to ~15% for both AOI22 and OAI22 gates using the approach in [2]. In Fig. 2, we compare the error in estimating the subthreshold leakage current for increasing the number of 'OFF' parallel NMOS transistors. Even when the number of parallel transistors is large, our proposed model incurs less than 0.15% error compared to larger than 25% error using the model of [2] for 10 parallel 'OFF' transistors. Similarly, for PMOS transistors, our model incurs very little error compared to the error of the model in [2].

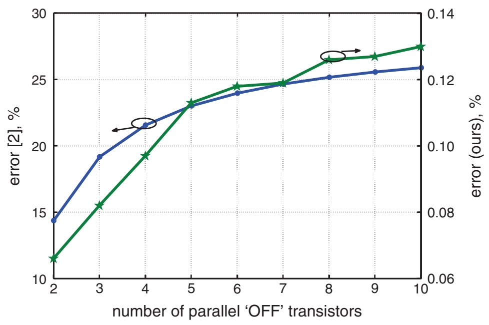

Fig. 2 *Comparison of error in leakage current between [2] and our approach for varying number of 'OFF' NMOS transistors*

Table 1: Training, testing error for NMOS stacks and SVM simulation time for 5000 MC samples

Model	Training time (s)	Training/testing corr. coefficient	SPICE/SVM time (s)	Speedup
n4/0	81.3595	0.9998/0.9998	105.36/5.6670	19×
n3/0	24.9007	0.9995/0.9990	82.90/5.4690	15×
n2/0	13.9523	0.9988/0.9906	62.06/5.2831	12×
n1/0	10.2730	0.9994/0.9992	53.55/5.0358	11×

Table 2: Comparison of percentage error in leakage current between our approach and [2] for AOI22 and OAI22 gate

Input	AOI22			OAI22		
	Leakage (nA)	[2]	Ours	Leakage (nA)	[2]	Ours
0	0.178	0.001	0.001	0.177	13.53	0.593
1	1.418	0.001	0.001	2.475	12.68	0.024
2	3.669	0.052	0.052	2.475	12.68	0.024
3	1.352	14.17	0.089	2.659	14.27	0.075
4	1.418	0.001	0.001	7.149	14.51	0.203
5	2.659	0.004	0.004	1.352	0.012	0.012
6	4.908	0.072	0.072	3.072	0.016	0.016
7	1.238	12.21	0.113	0.701	0.003	0.003
8	3.669	0.052	0.052	7.149	14.51	0.203
9	4.908	0.072	0.072	3.072	0.016	0.016
10	7.154	0.137	0.137	4.792	0.027	0.027
11	1.238	12.21	0.113	2.422	0.012	0.012
12	4.792	14.39	0.086	7.154	14.52	0.197
13	4.783	14.34	0.094	0.701	0.003	0.003
14	4.783	14.34	0.094	2.422	0.012	0.012
15	0.049	13.47	0.416	0.051	0.001	0.001

Conclusion: We have presented the SVM-based macromodel to predict the subthreshold leakage power of CMOS logic gates. About 30 stacks have been modelled to predict the leakage power of simple gates and gates containing parallel transistor stacks. Our experiments show that our methodology predicts the leakage power efficiently with significantly less error compared to the existing methodologies. It is highly desirable to consider the manufacturing variations in width while calculating the effective width of parallel transistors with the same input. Since we have modelled the leakage power under full variations in process–voltage–temperature–width space, the developed macromodel is a generalised model which efficiently predicts the leakage power of a gate under the effects of variations.

19 December 2012
doi: 10.1049/el.2012.4311
One or more of the Figures in this Letter are available in colour online.

L. Garg and V. Sahula (*Department of ECE, MNIT, Jaipur, India*)

E-mail: lokesh_garg20@yahoo.co.in

References

1 Viraraghavan, J.: 'Statistical leakage analysis framework using artificial neural networks considering process and environmental variations', PhD thesis, Indian Institute of Science (IISc), 2011

2 Al-Hertani, H., Al-Khalili, D., and Rozon, C.: 'Static power estimation of CMOS logic blocks in a library free design environment', *Int. J. Des. Anal. Tools Integr. Circuits Syst.*, 2011, **1**, (1), pp. 41–52

3 Muker, M., and Shams, M.: 'Designing digital subthreshold CMOS circuits using parallel transistor stacks', *Electron. Lett.*, 2011, **47**, (6), pp.372–374

4 Tanvir Hasan Morshed *et al.*: 'BSIM4.6.1 MOSFET model - user's manual. Technical report', University of California, Berkeley, 2011