This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSII.2015.2504270, IEEE Transactions on Circuits and Systems II: Express Briefs

Macromodels for Static Virtual Ground Voltage Estimation in Power Gated Circuits

Lokesh Garg, Vineet Sahula, *Senior Member, IEEE*

***Abstract*—Static virtual ground voltage (V_{gnd}) is an important parameter to be accurately and efficiently estimated for fine grained power gating in logic circuits. Previous work results in large error in V_{gnd} estimation due to conservative leakage models and inaccurate assumption of voltage conditions at the input of CMOS gates in power gated circuits. To overcome these problems, we propose Support Vector Machine (SVM) based macromodels to estimate leakage current of CMOS gates and thus achieve effective reduction of error in leakage model characterization. These models are then used in SVM classifier (SVC) and regressor (SVR) to formulate SVM regression based V_{gnd} model. SVC results in 3× saving in data generation time compared to HSPICE simulation to develop final V_{gnd} model. The proposed model results in <1% error and 23000× speedup than HSPICE for the largest benchmark circuit.**

***Index Terms*—Power gating, Support Vector Machine (SVM), Leakage current, Transistor stacks, Virtual ground, CMOS gates**

I. Introduction

Power gating is one of the promising techniques to reduce leakage power in logic circuits. A power gating device can be integrated as a footer/header in ground/supply gating case in Figure 1.(a)/1.(b). When footer transistor is in 'OFF' state, the leakage current flowing through logic circuit ($I_{leak}(circuit)$) charges the V_{gnd} node and reduces the V_{dd} to V_{gnd} voltage across logic circuit and hence, reduces the leakage current across logic circuit. Various issues such as - performance degradation, ground bounce noise, wake-up energy consumption, virtual ground or virtual supply voltage etc. are required to be considered before applying it to the logic circuits. In this paper, we consider ground gating case. However, similar approach can be applied for the supply gating case.

A. Previous Work and Motivation of Our work

Singh *et. al.* [1] represented $I_{leak}(circuit)$ and $I_{leak}(footer)$ as a function of V_{gnd}, then both currents are made equal to derive Exponential Linear (EL) model of V_{gnd} as a function of design parameters of logic circuit and footer transistor. The V_{gnd} estimated in [1] is a static voltage but it is a dynamic characteristic whose value is increased from lower to higher steady state value after the circuit is gated. Authors in [2], [3] estimated the dynamic V_{gnd} during mode transition process to calculate wake-up/sleep energy consumed due to V_{gnd} transition.

Authors in [2] also stated that the leakage estimation error is higher for V_{gnd} > $500mV$. The Authors predicted the $0.7nA$ leakage of AND8 gate as $0.4nA$, resulting into $\sim$ 30% error. It was also concluded that the leakage for V_{gnd} > $500mV$ is very small and does not impact calculation in mode transition process. In our work, we are modeling static V_{gnd} > $500mV$ voltage for all benchmark circuits, excluding C17 circuit. For static model, accuracy of the leakage model is very important at higher V_{gnd}. We also consider a range of width (W_{footer}), threshold voltage (V_{thf}) of footer transistor and sleep voltage (V_g), unlike single values used in [2]. Thus, inaccurate models for V_{gnd} > $500mV$ result into large error in leakage and V_{gnd} calculation.

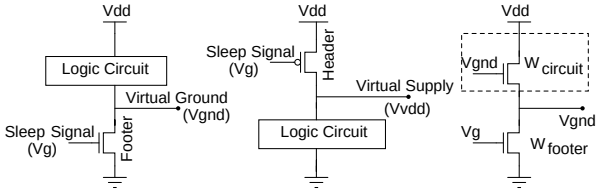


Figure 1. Power gating a) Ground gating case b) Supply gating case c) Equivalent circuit for virtual ground (V_{gnd}) model

$$E_{wake-up} = \frac{1}{2} C_{circuit} V_{gnd}^2 \quad (1)$$

Variation of V_{gnd} w.r.t. time in sleep and active mode is shown in Figure 2. Accurate calculation of energy during mode transition require steady state values in sleep and active mode as starting points. Previous work in [2], [3] assumes inaccurate starting point as V_{dd} or $0\,V$ in active and sleep mode respectively. Authors in [1] provided a first order formula to calculate wake-up energy consumption. If total capacitance at V_{gnd} node due to both logic circuit and footer transistors is denoted by $C_{circuit}$, then wake-up energy can be estimated as in (1). Accuracy of V_{gnd} model directly affects the accuracy of trade-off analysis between leakage saving in sleep mode and wake-up energy consumption.

In [1], [2], logic circuit is represented as an equivalent transistor of width (W_{width}), assuming same voltage at input of the equivalent transistor and virtual ground node as shown in Figure 1.(c). Static V_{gnd} can be estimated by equalizing leakage current of equivalent logic circuit and footer transistor as in (2).

$$I_{leak}(circuit) = I_{leak}(footer) \quad (2)$$

Replacing $I_{leak}(circuit)$ and $I_{leak}(footer)$ in (2) as a function of V_{gnd} gives an equation of V_{gnd} in terms of design parameters of logic circuit and footer transistor. From [4], sub-threshold leakage (I_{leak}) current of a single 'OFF' transistor can be represented as in (3).

$$I = A.e^{1/mV_T(V_g - V_s - V_{th0} - \gamma' V_s + \eta V_{ds})}.(1 - e^{-V_{ds}/V_T}) \quad (3)$$

$$\text{with } A = \mu_0 C'_{ox} \frac{W}{L_{eff}} (V_T)^2 c^{1.8} c^{-\triangle V_{th}/\eta V_T}$$

Here, V_{th0} is the threshold voltage at zero body bias, V_T is the thermal voltage, γ' is body bias coefficient and η is the DIBL coefficient. In [1], [2], for $V_{ds} \gg V_T$, the term $(1-e^{-V_{ds}/V_T})$ in (3) is neglected, without incurring error in leakage estimation. But for lower values of V_{ds} i.e. high source voltage for fixed drain voltage (V_{dd} in case of logic circuit) causes error in estimating I_{leak} of a transistor for comparable values of V_{ds} and V_T. In [1], authors represented (3) as a EL function of V_{gnd}, logic circuit and footer transistor design parameters while in [2], it is denoted only in terms of V_{gnd} as shown in (4) and (5) respectively. The authors in [3] represented leakage current as a polynomial function of degree N = 3 (Poly3) in terms of V_{gnd} as shown in (6).

$$I_{leak} = I_0 \frac{W}{L} 10^{(-V_{th} - (\eta V_{gnd})/S_S)} \quad (4)$$

$$I_{leak} = \hat{I_N}.e^{-K_N V_{gnd}} \quad (5)$$

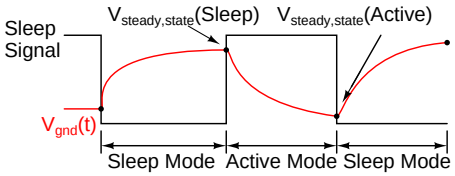
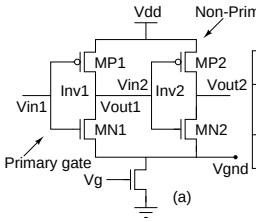


Figure 2. Typical sleep and active mode cycles in ground-gated circuits

$$I_{leak} = \sum_{j=0}^{N} p_j V_{gnd}^j \quad (6)$$

Poly3 model results in very large error in leakage estimation of logic circuit. However, for higher values of N, accuracy is improved but at the cost of increased complexity for obtaining the V_{gnd} equation. Thus, we need advanced dynamic model that does not presume any kind of EL or Poly3 form and can establish an accurate relation of $I_{leak}(circuit)$ in terms of V_{gnd}, depending on the complexity of the model. It has been shown that regression based non-linear modeling methods such as Neural Networks [5], Support Vector Machine [6] are more accurate than EL or Poly3 models for modeling of non-linear functions.



V_{gnd} (mV)	200	500	700	800	900
Vout1/Vin2 (mV)	200.01	500.01	700.63	808.61	980.87
Vout2 (mV)	999.98	999.99	999.99	999.95	992.64

(b)

Figure 3. (a) Input voltages estimation for CMOS gates with non-primary inputs (b) Vout1/Vin2 and Vout2 node voltages of circuit for varying V_{gnd}

Another disadvantage of previous work is that these models consider input voltage as V_{gnd} for whole voltage range of $0\,V$ to V_{dd}, but it is only true for the lower values of V_{gnd}. For higher values of V_{gnd}, both pull down network (PDN) and pull up network (PUN) are 'OFF', which makes the output voltage of that gate settled between V_{dd} and V_{gnd}. This alters the input voltage of other gates different from V_{gnd}. In Figure 3, 'Inv1' is the primary gate which is receiving input from primary inputs whereas 'Inv2' is a non-primary gate whose input is a output of preceding gate 'Inv1'. Now suppose Vin1 = 1, MP1 and MN1 are in 'OFF' and 'ON' condition respectively and the input value at 'Inv2' gate will depend on the value of V_{gnd}. **We define V_p as the value of V_{gnd} for which PDN is 'ON' for primary gate (In Figure 3, MN1) and PUN is 'ON' for non-primary gate (In Figure 3, MP2).** This V_p defines the input voltage at non-primary gate 'Inv2'. For $V_{gnd} < V_p$, NMOS transistor MN1 is 'ON' which makes the Vin2 similar as V_{gnd}. For $V_{gnd} > V_p$, MN1 is 'OFF', i.e. Vin2 resides at the little bit higher value than V_{gnd} i.e. V_{gnd}+ΔV, where $\triangle V$ is defined as the voltage drop across the PDN. Figure 3.(b) shows the output/input gate voltage of primary/non-primary gate Inv1/Inv2 and output of Inv2 gate of the circuit given in Figure 3.(a). The difference between the V_{gnd} voltage and Vout1/Vin2 is very less for lower values of V_{gnd} because drop across 'ON' PDN is very less and hence, can be removed from the circuit. Similarly, output of the non-primary gate 'Inv2' is close to the V_{dd} for lower V_{gnd} values and for higher values, this difference is high due to 'OFF' PUN network. By observing the node voltages in Figure 3.(b), V_p can be given as $0.7\,V$.

To check whether V_p voltage will be the same for larger circuits, we simulate C880 ISCAS'85 benchmark circuit for different values of V_{gnd}. Same output node voltage conditions are obtained for this circuit also, as shown in Figure 4. It is observed that V_p will be same for different circuits consist of CMOS gates from the same logic library. Input gate voltages of the CMOS gates in any circuit can be predicted for $V_{gnd} < V_p$ only. For $V_{gnd} > V_p$, we can not use leakage models for CMOS gates due to unknown input gate voltages. **The novel contributions of the proposed methodology are:**

- SVM based regression models are developed for leakage current estimation of CMOS gates as a function of gate input voltages ($0\,V$ to V_{dd}) and V_{gnd} ($0\,V$ to V_p).
- SVM based regression methodology is used to develop static V_{gnd} models for higher accuracy and efficient computation.
- Since generation of simulation data for regression based V_{gnd} models take longer time for larger CMOS circuits. We propose a V_{gnd} partition based sample generation methodology to reduce model generation time, that uses SPICE simulation for $V_{gnd} > V_p$ and proposed leakage models for $V_{gnd} < V_p$. SVM based classifier is developed to partition the input parameter range based on V_{gnd} value.

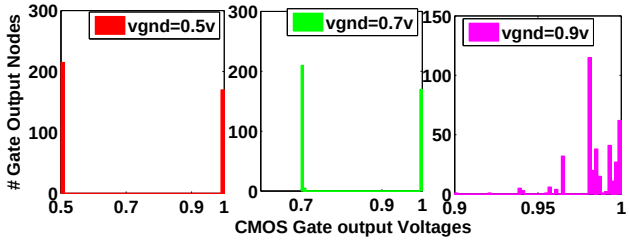


Figure 4. Fraction of number of CMOS gates presented in C880 ISCAS'85 benchmark circuit with output node voltages ranging between $0\,V$ to V_{dd} for varying virtual ground voltage (V_{gnd})

II. Equivalent Stack Extraction and Leakage Modeling of Stacks using SVM

We formulate some rules for finding the equivalent stack model of a CMOS gate on the basis of gate type (primary or non-primary gate) and input to the gate (input vectors have the significant impact on the leakage current of a CMOS gate). Consider a 2-input NAND gate as shown in Figure 5.(a) with different input voltage conditions of primary and non-primary gates. In our methodology, we remove either PDN or PUN based on the input voltages and V_{gnd} value. For AND family of gates, rules can be described as follows: **(1)** For any primary gate, if any input to the CMOS gate is at logic '0', then remove PUN from that CMOS gate and connect output node to the V_{dd} because logic '0' input will make the PUN 'ON'. This rule is independent from the value of V_{gnd} because it does not affect the V_{gs} of PMOS transistors in PUN. **(2)** For any primary gate, if all inputs are at logic '1' and if V_{gnd} is less than V_p, remove PDN and connect output node to V_{gnd} otherwise don't remove PUN and PDN. **(3)** For any non-primary gate, remove PUN if $V_{gnd} < V_p$ because PMOS transistor in PUN makes it 'ON' and output node can be connected to V_{dd} otherwise don't remove PUN and PDN.

Above rules can also be applied to parallel 'OFF' transistor stacks, consisting one 'OFF' transistor in each stack or multiple 'OFF' transistors with 'ON' transistors in a stack. Equivalent model can be derived by summing the currents from all the stacks. In this work, we assume that the maximum stack-size is 4, as higher order stack will increase the delay of a gate due to increased logical effort. We have used conventions for labeling transistor stacks as follows: {stack type}{stack size}. Here, stack type indicates NMOS/PMOS stack. Stack size represents the number of transistors on a stack.

Figure 5.(b) shows characterized transistor stacks for AND family of CMOS gates. Similarly, rules and transistor stacks for OR family of gates can also be described. Models n1 to n4 are for CMOS gates in logic circuit and n1,f is for footer transistor due to the use of different model parameters than n1. Now, consider a 2-transistor NMOS stack model n2, as shown in Figure 5(b). From [2], leakage current of n2 stack can be given as:

$$I_{leak} = \hat{I_{leak}} w_{stack} e^{-A.Vin1 - B.Vin2 - C.V_{gnd}} \quad (7)$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSII.2015.2504270, IEEE Transactions on Circuits and Systems II: Express Briefs

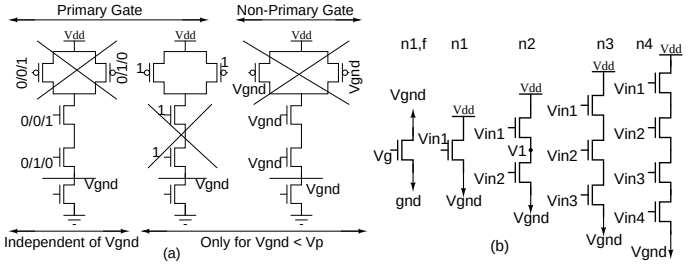


Figure 5. (a) Equivalent stack models for different input vectors of NAND2 gate (b) Characterized NMOS stack models

We use SVM based regression models to reduce the error in leakage modeling of transistor stacks caused by neglecting the term $(1-e^{-V_{ds}/V_T})$ in (3). The advantages are accurate and reduced number of models, consideration of input vectors and effect of 'ON' transistors in 'OFF' network. Proposed SVM models can also be used for leakage estimation of CMOS gates, consisting parallel combination of 'OFF' transistors or series of 'OFF' stacks. In this work, we only consider simple CMOS gates in logic circuits because most of the gates have the simple parallel structures such as- NAND, Buffers, NOR, OAI, AOI.

III. SVC MODEL FOR INPUT SPACE PARTITIONING

SVC model classifies the training data into two sets in the multi-dimensional parameter space, depending on the constraints (C_f). In this work, we need to develop a classifier to separate V_{gnd} values either less than or greater than V_p in input space (W_{footer}, V_{thf}, V_g). First we define the feasible space i.e. $V_{gnd} < V_p$, for which we can use our pre-characterized leakage models for data generation to formulate SVR V_{gnd} models. The feasible design space $S \subseteq R^n$ can be defined as in (8). Note that x is the vector for all design (footer transistor) parameters. Feasible function, $y(x)$, can only take two values {+1,-1} depending on whether $x \epsilon S$, as defined in (9).

$$S = \{x : x \in R^n, C_f\};\ C_f = \{V_{gnd} < V_p\} \tag{8}$$

$$y(x) = \begin{cases} +1 & if\ x \in S \\ -1 & if\ x \notin S \end{cases} \tag{9}$$

Using $\{(x_1, y_1), (x_2, y_2), ...(x_k, y_k)\}$ as a set of training samples, the objective of SVC is to find a hyper-plane with maximum separation between data points of +1 and -1 type classes. In high dimensional space, two classes may not be linearly separable. SVM maps input data into high dimensional feature space to create a optimal separating hyperplane using kernel functions. SVM uses quadratic programming to solve this problem, instead of gradient based optimization used by traditional Neural Network approaches [7]. Neural Network suffers with the problem of being trapped in multiple local minima. Least square version of SVM (LSSVM) uses the equality type constraints in the problem formulation, thus the solution is obtained through solving a set of linear solutions. Therefore, LSSVM is free of local minima and also has low computational cost [7]. We also apply active learning methodology to train SVC model. In Active learning [8], learner has a control over supplied training data set, which reduces the sample size and increases accuracy. Learning starts with the fewer training samples, further samples are added according to the requested query to achieve a goal of low error rate with minimum training samples. Authors in [9] have used active learning with SVM, based on version space reduction for text classification. While in [10], active learning is applied to optimize expected future error. In our methodology, we use active learning process to generate new training samples around maximum error sample to achieve desired accuracy. We start training with fewer samples and only generate two samples around maximum error sample to add them with previous training data. Since, $V_{gnd} > V_p$ and $V_{gnd} < V_p$ input spaces are separated by smooth surface. Thus, active learning method based on maximum error sample do not bias the classifier and provides accurate results for unknown testing samples. The resulting SVC model can be given as:

$$y_k = sign\left[\sum_{k=1}^{N} \alpha_k K(x_k, x) + b\right] \tag{10}$$

Where $K(x_k, x)$ is kernel function and α_k, b are solutions of the linear systems. +1(-1) output from (10) shows $V_{gnd} < V_p (V_{gnd} > V_p)$. $I_{leak}(footer)$ in terms of V_{gnd} can be given as [2].

$$I_{leak}(footer) = \begin{cases} \hat{I_f}.e^{K_N(V_{gnd}-V_{dd})} & V_{gnd} > 4V_T \\ 0 & V_{gnd} < 4V_T \end{cases} \tag{11}$$

Circuit leakage and V_{gnd} can be related using (2) and (11) as:

$$I_{leak}(circuit) = \hat{I_f}.e^{K_N(V_{gnd}-V_{dd})} \tag{12}$$

$$V_{gnd} = V_{dd} + K_N^{'}.log(\frac{I_{leak}(circuit)}{\hat{I_f}}) \tag{13}$$

From (13), V_{gnd} will be higher for high gate leakage. Generally leakage of a single gate is not enough to force the V_{gnd} to cross V_p. But, in a circuit, maximum V_{gnd} for a gate will be decided by the leakage contribution of other CMOS gates also and can cross the V_p voltage. V_{gnd} is also a strong function of input vector supplied to logic circuit and footer transistor parameters. Before developing SVM classifier, we first need to check whether SVM classifier is required or not in the given input parameter space. This simple analysis can result in saving the high modeling time by avoiding the development of classifier model. Singh *et. al.* [1] developed the first order V_{gnd} model as in (14).

$$V_{gnd} = \frac{-V_g + S_S log_{10}(\frac{W_{circuit}}{W_{footer}}) + (V_{thf} - V_{thc}) + \eta V_{dd}}{2\eta} \tag{14}$$

Here, V_g is footer gate voltage, V_{thf}, $W_{circuit}$ and V_{thc}, W_{footer} are threshold voltages and widths of logic circuit and footer transistor respectively, η is the DIBL coefficient and S_S is the subthreshold slope. In this work, we are only considering V_g, W_{footer}, V_{thf} as design variables. Thus, maximum value of V_{gnd} will be obtained at minimum V_g, W_{footer} and maximum V_{thf} . If this maximum value is greater than V_p, then such classifier is required. From the benchmark circuits used to evaluate our approach, only C17 circuit do not need any classifier model to be developed.

IV. SVR MACROMODELS FOR V_{gnd} ESTIMATION

SVR function to relate input and output can be described as:

$$y_k = \sum_{k=1}^{N} \alpha_k K(x_k, x) + b \tag{15}$$

SVR uses the actual values of output parameters, unlike +1/-1 used in SVC. Our SVR model generation methodology starts with the initial training data set which is directly taken from the data used for SVC model with the actual values of V_{gnd}. We can do this without incurring any error in models because SVC and SVR model are completely independent of each other. This process saves our time for data generation which can be very high for larger CMOS logic circuits. At this point, model is trained and tested on testing data set. Accuracy of the model is calculated based on mean square error ($Error_{est}$) between model output $Y_{test,est}$ and actual output Y_{test} using (16).

$$Error_{est} = \frac{1}{n}\sum_{i=1}^{n}(Y_{test} - Y_{test,est})^2 \tag{16}$$

If the accuracy of the model is less than the desired accuracy, new samples are generated around maximum error sample. SVM

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSII.2015.2504270, IEEE Transactions on Circuits and Systems II: Express Briefs

classifier selects whether our stack models or HSPICE simulation is used for the V_{gnd} estimation for these newly generated samples. For a given CMOS circuit with applied input vector and $V_{gnd} < V_p$, we separate each gate according to the stack type and input pattern. The aim of V_{gnd} estimation problem is is to find V_{gnd} for which $I_{leak}(circuit) = I_{leak}(footer)$ and solved using algorithm in Figure 6

Input: Pre-characterized leakage current models, V_p (Maximum value that can be predicted using our leakage current models)
Output: Virtual ground voltage (V_{gnd})
1. $V_{gnd,high} = V_p$
2. $V_{gnd,low} = 0V$
3. $V_{gnd,start} = (V_{gnd,high} + V_{gnd,low})/2$
4. Estimate $I_{leak}(circuit)$, $I_{leak}(footer) = f(V_{gnd,start})$
5. $\Delta I_D = I_{leak}(circuit) - I_{leak}(footer)$
6. $\Delta I_{D,Max} = \varepsilon$
7. **While** ($|\Delta I_D| \geq \Delta I_{D,Max}$) **do**
8. **If** ($\Delta I_D > \varepsilon$) **then**
9. $V_{gnd,low} = V_{gnd,start}$
10. **Else**
11. $V_{gnd,high} = V_{gnd,start}$
12. **end If**
13. $V_{gnd,start} = (V_{gnd,high} + V_{gnd,low})/2$
14. Estimate $I_{leak}(circuit)$, $I_{leak}(footer) = f(V_{gnd,start})$
15. $\Delta I_D = I_{leak}(circuit) - I_{leak}(footer)$
16. **end While**
17. $V_{gnd} = V_{gnd,start}$

Figure 6. Bisection search algorithm to find virtual ground node value for $V_{gnd} < V_p$

Algorithm for V_{gnd} estimation starts with inputs as pre-characterized leakage current models, V_p value. Parameters $V_{gnd,high}$ and $V_{gnd,low}$ define the maximum and minimum value as V_p and $0V$ respectively. $V_{gnd,start}$ in line 3 defines the middle value in the search range. In line 4, SVM regression macromodels are used to estimate $I_{leak}(circuit)$ and $I_{leak}(footer)$ which depends on current $V_{gnd,start}$ value and difference between $I_{leak}(circuit)$ and $I_{leak}(footer)$ is calculated in line 5. $\triangle I_{D,Max}$ defines the maximum tolerable limit that can be possible in difference between $I_{leak}(circuit)$ and $I_{leak}(footer)$. If $\triangle I_D$ is positive and larger than the $\triangle I_{D,Max}$, then lower value of V_{gnd}, $V_{gnd,low}$ is set to $V_{gnd,start}$ otherwise higher value of V_{gnd} is set to $V_{gnd,start}$. Now, new $V_{gnd,start}$ value is calculated by averaging the new $V_{gnd,high}$ and $V_{gnd,low}$ value. Performing the steps in lines 9 - 13, reduce the search space half in the next iteration. This process is repeated until difference in $|\triangle I_D|$ is under tolerable limits.

V. Experimental Results

We use RBF kernel $K(x, x_k) = exp(-||x - x_k||^2/\sigma^2)$ and select SVM model regularization parameter (γ), kernel function variable (σ) as 1 and 10 respectively. We compare our results on ISCAS'85 benchmark circuits in $45nm$ technology. V_{gnd} model is developed as a function of W_{footer}, V_g, V_{thf} of the footer transistor in the range of $45nm$ - $500nm$, $0V$ - $1V$ and $0.4V$ - $0.6V$ respectively. All samples used for training and testing are generated using Latin Hypercube Sampling method.

A. Accuracy and Efficiency Evaluation of Leakage models

Table I shows the accuracy of characterized NMOS stack models in terms of correlation coefficient (ρ) and MSE w.r.t. HSPICE output. We use 1000 training and 5000 test samples. Our leakage model's accuracy is very high when compared to the HSPICE results. In Table I, information related to PMOS stack models is not included. In case of PMOS stacks also, training time, simulation time, ρ and MSE are approximately of the same order because these parameter less depend on type (NMOS/PMOS) of the models.

Figure 7 shows circuit diagram of C17 ISCAS'85 benchmark circuit and equivalent stack models (extracted using rules described in Section II) for input vector '00000'. Figure 8 shows

Table I
Accuracy and efficiency evaluation of leakage models

Model	*Training time (ms)*	*MSE*	ρ	*Simulation Time* (mS)
n4	5.12	5.5712e-9	0.9995	0.2042
n3	4.32	3.3912e-9	0.9995	0.1978
n2	2.87	0.2358e-9	0.9996	0.1904
n1	1.71	0.1102e-9	0.9998	0.1846
n1,f	1.83	2.0112e-9	0.9994	0.1892

comparison of logic circuit leakage for varying V_{gnd} using model in [2], [3] and our model. Fitting curve of model in [2], [3] is obtained using curve fitting toolbox of MATLAB. It can be observed that our model accurately matches with the HSPICE output compared to the large error of model in [2], [3].

B. Static V_{gnd} model Analysis

We verify proposed V_{gnd} model on ISCAS'85 benchmark circuits. Maximum 100 training samples for SVC and 350 samples for SVR V_{gnd} model are used. Table II shows the results of proposed final V_{gnd} model and EL model [2]. In Figure 8, leakage from our model for C17 circuit is very small w.r.t. HSPICE and average error for V_{gnd} model of C17 circuit is 0.14%. For all circuits, error in V_{gnd} voltage is due to cumulative error of leakage, SVC and SVR models. For larger circuits, maximum error is always less than the 0.8%, 13× lower than the previous work in [2]. This comparison is based on 5000 test samples across 100 input vectors. Column 2 of Table II shows time required to develop circuit level V_{gnd} model, combining the time of data generation to develop both SVC and SVR model. Proposed stack models with SVC help in reducing time to generate the samples for SVR model of V_{gnd}. SVC and SVR models are trained with different number of samples, therefore runtime of both models is different. Testing sample in input space are only simulated from final SVR V_{gnd} model. Thus, runtime of the proposed approach is only decided by final SVR V_{gnd} model. Proposed V_{gnd} model's runtime is approximately equal for all circuits because model's runtime mainly depends on number of training samples and variables in the training data. Both parameters are same in all circuits. Model has maximum runtime improvement of 23000× on C7552 benchmark circuit. We have applied previous and our V_{gnd} model to calculate $E_{wake-up}$ using (1). Last column in Table II shows the error in energy estimation. Our model results in average 3.5% error, compared to 13% by previous work. In this work, we are only developing accurate V_{gnd} model, thus error in energy estimation in our case is due to first order energy model and inaccurate capacitance (capacitance will also change with V_{gnd}, can not be defined by a single value).

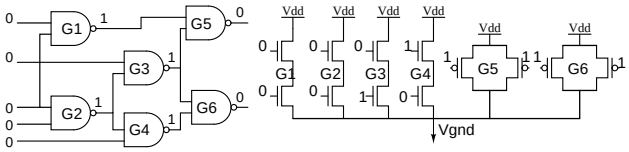


Figure 7. C17 circuit and it's equivalent stack model for input '00000'

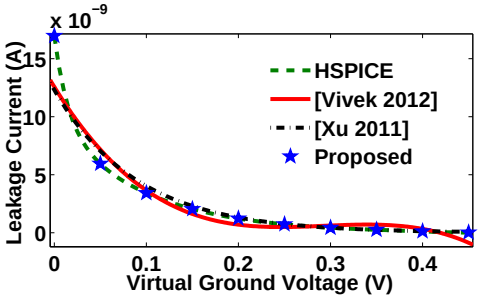


Figure 8. Comparison of leakage models for C17 circuit with '00000' input

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSII.2015.2504270, IEEE Transactions on Circuits and Systems II: Express Briefs

Table II
V_{gnd} MODEL RESULTS FOR ISCAS'85 BENCHMARK CIRCUITS

Circuit	# Gates	Model generation time (S)			Error (%) for V_{gnd} estimation						Model Evaluation time (mS)			Speedup ($\times$)		Error (%) for energy estimation	
		HSPICE (without SVC)	Proposed (with SVC)		[2]		Proposed				HSPICE	[2]	Proposed	[2]	Proposed	[2]	Proposed
			SVC	SVR	max	avg	(without SVC) max	(without SVC) avg	(with SVC) max	(with SVC) avg							
C17	6	7.44	3.13	2.57	7.01	5.93	0.17	0.10	0.25	0.14	24.80	0.053	0.093	468	266	10.29	2.79
C432	261	114.21	32.63	14.82	6.29	4.21	0.15	0.10	0.22	0.11	325.69	0.054	0.093	6031	3502	8.66	2.56
C880	383	140.81	40.23	16.63	10.92	9.42	0.29	0.14	0.43	0.26	401.20	0.055	0.093	7295	4314	13.78	2.95
C1908	972	231.52	66.15	14.44	6.72	4.57	0.40	0.22	0.55	0.38	660.10	0.056	0.091	11787	6948	8.79	3.28
C2670	1211	359.11	102.60	23.91	11.89	10.33	0.44	0.27	0.57	0.39	1025.71	0.052	0.091	19725	11271	15.33	3.36
C6288	2351	527.63	125.75	25.7	10.78	9.01	0.72	0.58	0.80	0.73	1505.67	0.053	0.090	28409	16729	13.13	4.08
C7552	3624	766.23	218.92	38.98	13.47	12.27	0.63	0.41	0.75	0.69	2188.56	0.050	0.095	43771	23037	17.45	3.69

C. Effects of SVC model on Accuracy, Runtime and Modeling time of V_{gnd} model

We conducted three set of experiments on C7552 benchmark circuit to analyze the effect of SVC model on accuracy and modeling time of the proposed approach. For this purpose, we assume 5% to 100% misclassification in the step size of 5%.

Case (I) - Training samples for SVR model from $V_{gnd} < V_p$ space are misclassified as $V_{gnd} > V_p$. In this case, more number of training samples need to be simulated through SPICE, instead of proposed leakage models to obtain V_{gnd} voltage. It increases the accuracy of V_{gnd} model but at the cost of increased modeling due to the use of increased SPICE simulations. Error reduction and modeling time increment for case (I) can be verified from Figures 9.(a) and 9.(b).

Case (II) - Training samples for SVR model from $V_{gnd} > V_p$ space are misclassified as $V_{gnd} < V_p$. SVC model suggests to use leakage models with algorithm in Figure 6. But, our leakage models are only valid and can generate samples in $V_{gnd} < V_p$ range only and hence, V_{gnd} value can not be found. These samples are assumed to be misclassified. We keep these samples in $V_{gnd} > V_p$ range and simulate them through SPICE to obtain V_{gnd} voltage. Evaluation of correct class for misclassified samples results in negligible increase in modeling time. Negligible increase in both modeling time and error for case (II) can be verified from Figures 9.(a) and 9.(b).

Case (III) - Training samples for SVR model from $V_{gnd} < V_p$ space are misclassified as $V_{gnd} > V_p$ and vice-versa. Error and modeling time in this case is approximately equal to case (I) because these parameters in case (II) are same for all misclassification rates, as shown in Figures 9.(a) and 9.(b).

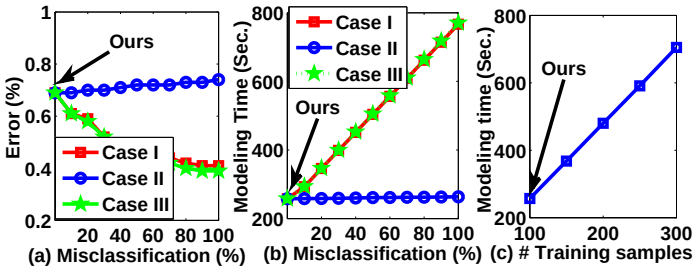


Figure 9. Effect of SVC misclassification on (a) Accuracy (b) Modeling time of V_{gnd} model (c) Effect of SVC training samples on modeling time of V_{gnd} model

Total modeling time of the proposed approach is also affected by number of SVC training samples. If SVC model is trained with higher number of samples (100 in this work), then SVC model generation time and time required to classify training sample is also high because more number of α_k in (10) are used to classify a sample. Figure 9.(c) shows the effect of number of SVC model training samples. We have varied the number of training samples from 100 to 300. Training time of SVC model varies from 0.025 Sec. to 0.59 Sec. with complexity of $O(n^3)$, while time to generate training samples varies 257 Sec. to 705 Sec. with complexity $O(n)$. Hence, total model generation time is dominated by samples generation time with linear increment w.r.t. number of training samples. Our overall experiments related to SVC model imply that the runtime and accuracy of the proposed SVC model impacts modeling time without degrading the accuracy of V_{gnd} model. For C7552 circuit, 0.74 Sec. is required to generate, while only 0.095 mSec. is required to evaluate single sample. Even though the model generation time is higher than the model evaluation time, but the proposed model is reusable. However, the use of SVC model results in $\sim 3\times$ saving in model generation time.

The value of V_p used to build SVC model also affects the accuracy and modeling time of the proposed approach. If $V_p > 0.7\,V$, then less number of SPICE simulation will be required due to the reduction of $V_{gnd} > V_p$ space and maximum samples can be evaluated using proposed leakage models. It will have the advantage of reduced modeling time but negligible reduction in the accuracy due to the use of accurate leakage models in larger $V_{gnd} < V_p$ space. If $V_p < 0.7\,V$, then SPICE simulations will be used for larger number of input samples. It will increase the modeling time but also improve the accuracy of the model.

VI. CONCLUSION

We have developed SVM regression based macromodels for leakage current modeling and SVC model to save time for training data generation for final V_{gnd} model. We also proposed SVM regression based V_{gnd} model with less than 1% error. The proposed V_{gnd} model is 23000$\times$ faster than HSPICE for the largest C7552 ISCAS'85 benchmark circuit.

REFERENCES

[1] H. Singh, K. Agarwal, D Sylvester, and K.J. Nowka. Enhanced leakage reduction techniques using intermediate strength power gating. *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, 15(11):1215–1224, 2007.

[2] Hao Xu, R. Vemuri, and Wen-Ben Jone. Dynamic characteristics of power gating during mode transition. *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, 19(2):237–249, 2011.

[3] V.D. Tovinakere, O. Sentieys, and S. Derrien. A semiempirical model for wakeup time estimation in power-gated logic clusters. In *Design Automation Conference, 2012 49th ACM/EDAC/IEEE*, pages 48–55.

[4] K. Roy, S. Mukhopadhyay, and H. Mahmoodi-Meimand. Leakage current mechanisms and leakage reduction techniques in deep-submicrometer cmos circuits. *Proc. of the IEEE*, 91(2):305–327, 2003.

[5] J. Viraraghavan, B.P. Das, and B. Amrutur. Voltage and temperature scalable standard cell leakage models based on stacks for statistical leakage characterization. In *VLSI Design, 2008. 21st International Conference on*, pages 667 –672.

[6] L. Garg and V. Sahula. Variability aware support vector machine based macromodels for statistical estimation of subthreshold leakage power. In *Synthesis, Modeling, Analysis and Simulation Methods and Applications to Circuit Design, 2012 International Conference on*, pages 253–256.

[7] J.A.K. Suykens and J. Vandewalle. Least squares support vector machine classifiers. *Neural Processing Letters*, 9(3):293–300, 1999.

[8] Burr Settles. Active learning literature survey. Technical report, University of Wisconsin–Madison, 2010.

[9] Simon Tong and Daphne Koller. Support vector machine active learning with applications to text classification. *The Journal of Machine Learning Research*, 2:45–66, March 2002.

[10] Nicholas Roy and Andrew McCallum. Toward optimal active learning through sampling estimation of error reduction. In *ICML*, pages 441–448, 2001.