Static Power Estimation of CMOS Logic Blocks in a Library Free Design Environment

Hussam Al-Hertani, Dhamin Al-Khalili, and Côme Rozon

Abstract—This paper introduces a new approach to pattern dependent static power estimation in logic blocks, which are realized 'on-the-fly' in a library-free design environment. A static current model is first developed at the transistor level and then extended to the logic gate level and finally the logic block level. For varying transistors widths and input stimuli, the transistor level model has performed with good accuracy compared to SPICE for technologies ranging from 65nm down to 32nm. The gate level model is pattern dependent and deals with basic gates and complex gates. A transistor collapsing scheme was developed to achieve simpler structure leading to analytical models with high computational efficiency and good accuracy ranging from 0.1-5.4% for basic logic gates and 3.7-6.2% for complex gates.

Using these static current estimation models, a methodology has been introduced to estimate static power dissipation of logic blocks in a library-free design environment, in which the cells are generated and sized 'on-the-fly' driven by specification and targeted technology. Across several MCNC benchmarks, the estimation methodology proposed exhibits a worst case mean percentage error of 1.1% compared to SPICE. It also exhibits runtime that is on average 43 times faster than SPICE.

Index Terms—Static power dissipation, subthreshold leakage, gate tunneling leakage, library-free synthesis.

I. INTRODUCTION

SIGNIFICANT levels of static leakage exhibited by nanometer devices continue to make static power a major source of concern for VLSI designers. Even with advances such as high-k metal gate transistors [1] that reduce gate tunneling and MTCMOS based circuit design [2] that reduce subthreshold leakage, static leakage continues to cause significantly high static power dissipation levels.

The intention of the work presented in this paper is to develop an accurate and computationally efficient static power dissipation estimation model, capable of being used in a library-free [3] based ASIC design. In library free synthesis, logic cells are generated ‘on-the-fly’ as dictated by user specifications. Early estimation of various metrics is usually required, of which static power is essential and is the focus of this work. Because the component library and its constituent cells are created ‘on-the-fly’, and because the transistors comprising the cells are also sized ‘on-the-fly’ a look-up table approach to leakage modeling is insufficient. The model proposed in this work addresses these limitations while at the same time offering good accuracy and significant speed up in runtime with respect to SPICE simulation.

Static leakage estimation is first formulated at the transistor level. It is extended to the logic gate level and then finally to the logic block level. This bottom-up hierarchical approach to leakage estimation enables us to i) simplify the modeling at each level (transistor level, logic gate level and logic block level); ii) facilitate significant overall run-time savings, and iii) obtain a good model accuracy.

The four leakage mechanisms addressed at the transistor level are subthreshold leakage \(I_{SUB}\) [4], forward gate-tunneling \(I_{FCC}\) [4], reverse \(I_{RBC}\) gate-tunneling [4] and junction tunneling \(I_{JUNC}\) [5]. Subthreshold current occurs when the transistor is OFF or when operating in weak inversion \((V_{GS} < V_{TH})\). As the number of OFF series transistors in a transistor stack increase, subthreshold leakage is reduced significantly. This is known as the ‘stacking effect’ [4] and it implies that the amount of leakage in a transistor stack is a function of the applied input vector, the transistor widths and the supply voltage. The stacking effect increases the complexity of subthreshold leakage estimation considerably.

The combination of gate voltage and reduction of gate oxide thickness in UDSM processes results in an increasingly strong electric field across the gate oxide. This electric field is large enough to cause electron tunneling through the gate oxide. This phenomenon is called gate tunneling. Gate tunneling leakage current can be very large for thin oxide layers. This is very significant for UDSM processes where the effective gate oxide thickness (TOXE) is scaled significantly. In fact TOXE is 18.5, 17.5 and 16.5 Å for the PTM 65nm, 45nm and 32nm processes respectively resulting in gate leakage current magnitudes as high as several nano-amperes (nA) for unit width transistors in these processes. For NMOS transistors, forward gate tunneling is most significant when the device is ON and in strong inversion. Backward gate tunneling exhibits slightly smaller leakage currents (but within the same order of magnitude) than forward gate tunneling currents. It generally takes place when the NMOS transistor is OFF and the drain is at \(V_{DD}\).

Junction tunneling is caused by the existence of a high electric field across the reverse biased p-n junction causing significant current to flow through the junction due to tunneling of electrons from the valence band of the p-region to the conduction band of the n-region. In NMOS transistors, junction tunneling takes place when the drain is at a higher potential than the body of the transistor. This causes significant energy band bending in the drain region, allowing electron-hole pair generation through avalanche multiplication and band-to-band tunneling.

Several transistor-level static leakage current models exist [6], [7], [8], [9], [10], [11], [12]; but almost all of them involve

Hussam Al-Hertani is with the Department of Electrical and Computer Engineering, Royal Military College of Canada, Kingston, ON, K7K7B4, Canada. E-mail: halherta@gmail.com.

Dhamin Al-Khalili and and Côme Rozon are with Royal Military College of Canada.
only one or two leakage current mechanisms at the most. They also tend to be strongly based on the pre-extraction of leakage current per unit transistor width for various scenarios (i.e., operating regions). Other models such as those presented in [13] tend to be purely deterministic with strong emphasis on the physical transistor structure. The model proposed here estimates leakage current due to all mechanisms. It is mostly deterministic with some required pre-characterization. Static leakage in transistor stacks with varying transistor widths may also be estimated.

The maximum stack length considered in this work is limited to 4 NMOS transistors and 3 PMOS transistors. This is because in sub-100nm technologies, due to the small supply voltage and the effect of velocity saturation, there is a limitation to how many transistors one can have in series to achieve reliable circuit operation. This limitation is also dictated by the constraints of the virtual library approach [3]. Additionally, most of the literature investigating the stack length variable (e.g., [7], [9], [10], [14]), limit the stack length to a maximum of 3 or 4 transistors, even in the older technologies where higher supply voltages (3.3-1.8V) facilitates significantly more reliable operation for larger stack lengths.

The paper is divided into six sections. First, the complete transistor level framework is introduced. This is followed by an introduction to the methodology for estimating static leakage in basic and complex logic gates. The next section extends the gate-level static leakage estimation to the logic block level. Finally, the results are presented. The performance of the proposed static leakage estimation approach at the transistor level, the logic gate level and the logic block level is measured and analyzed in comparison with SPICE simulations using PTM’s [15] BSIM4 65nm, 45nm and 32nm model files. Runtime comparisons are also made.

II. TRANSISTOR LEVEL LEAKAGE ESTIMATION

The proposed framework for transistor level leakage estimation involves modeling the four leakage mechanisms; the subthreshold leakage current (\(I_{SUB}\)), the forward gate tunneling leakage current (\(I_{FGC}\)), the reverse gate tunneling leakage current (\(I_{RGC}\)), and the junction tunneling current (\(I_{JUNC}\)).

The constituent models for each leakage current mechanism is technology dependent and requires some device model parameters from the BSIM4 technology model files as well as some a priori sample simulation data for use in either fitting (forward/reverse gate tunneling leakage and junction tunnelling) or for use by the Bayesian inference approach required for estimating the subthreshold leakage.

A. Subthreshold Current

The expression for subthreshold current can be given by [16]:

\[
I_{SUB} = \mu_0 C'_{ox} \left( \frac{1}{E_{eff}} \right) \frac{V^2}{n_l} \exp(1.8) W \cdot \exp \left( \frac{V_{GS} - V_{Th} - V_{OFF} + \eta V_{DS}}{n_l \cdot V_l} \right) \left( 1 - \exp \left( -\frac{V_{DS}}{V_l} \right) \right) \tag{1}
\]

where \(V_{GS}\) is the gate to source voltage, \(V_{DS}\) is the drain to source voltage, \(V_l\) is the thermal voltage, \(\mu_0\) is the subthreshold swing coefficient, \(V_{Th}\) is the threshold voltage, \(V_{OFF}\) is a BSIM fitting coefficient, \(\eta\) is the DIBL coefficient, \(C'_{ox}\) is the gate oxide capacitance per unit width and \(\mu_0\) is the zero bias mobility.

The threshold voltage in the subthreshold leakage equation maybe approximated with minimal error as:

\[
V_{TH} = V'_{TH0} + \gamma' V_{SB} \tag{2}
\]

where \(V'_{TH0}\) and \(\gamma'\) are ‘linearized’ versions of the zero bias threshold voltage and the body effect coefficients respectively.

In order to derive a simplified model, the last exponent term in Equation (1) is manipulated such that the expression is equated to \(\exp(-x)\) expression is equated to \(\exp(y)\) where \(y = m_n x + c_n\) and \(x = \frac{V_{DS}}{V_l}\) (Note that when \(x = \frac{V_{DS}}{V_l} \geq 4, y \approx 0\)). Now consider the equation:

\[
y = \begin{cases} 
0, & x \geq 4.00 \\
\ln(1 - \exp(-x)) = m_A x + c_A, & 0.60 \leq x < 4.00 \\
\ln(1 - \exp(-x)) = m_B x + c_B, & 0.01 \leq x < 0.60
\end{cases} \tag{3}
\]

This shows that function \(y\) can be approximated by a piecewise linear function as seen in Figure 1. This linear interpolation provides an optimal fit w.r.t to the transition point between them. The optimal transition point was found using an exhaustive search that would provide the best overall fit using two subsequent linear functions. This point was determined to be \(x = 0.6\). Taking the exponent of both sides of Equation (3) yields the following approximation:

\[
\exp(y) = \begin{cases} 
\exp(0) = 1, & x \geq 4.00 \\
1 - \exp(-x) = \exp(m_A x + c_A), & 0.60 \leq x < 4.00 \\
1 - \exp(-x) = \exp(m_B x + c_B), & 0.01 \leq x < 0.60
\end{cases} \tag{4}
\]

Using this result, the subthreshold current can be rewritten as follows:
From simulations it was also found that $V_{TH}$ for a single transistor in all relevant regions is not process dependent. The last set of equations provide a term. Note how the piecewise linear functions presented above approximate the mathematical expression in Equation (1) and is not process dependent. The last set of equations provide a simplified and comprehensive estimate of subthreshold leakage for a single transistor in all relevant $\left(\frac{V_{DS}}{V_{t}}\right)$ regions.

$$I_{SUB} = \begin{cases} AW \exp \left(\frac{V_{GS} - V_{TH} - V_{OFF} + nV_{T}}{nV_{t}}\right), & x \geq 4.00 (a) \\ AW \exp \left(\frac{V_{GS} - V_{TH} - V_{OFF} + vV_{T}}{nV_{t}}\right), & 0.60 \leq x < 4.00 (b) \\ AW \exp \left(\frac{V_{GS} - V_{TH} - V_{OFF} + vV_{T}}{nV_{t}}\right), & 0.01 \leq x < 0.60 (c) \end{cases}$$

where $A = \mu C_{ox} \frac{V_{DS}}{I_{DSF}} v_{t}^{2} \exp (1.8)$ and $V_{TH}$ is calculated according to Equation (2). The simplification shown in Equations (4) and (5) allows us to approximate the BSIM subthreshold leakage expression using a single exponential term. Note how the piecewise linear functions presented above approximate the mathematical expression in Equation (1) and is not process dependent. The last set of equations provide a simplified and comprehensive estimate of subthreshold leakage for a single transistor in all relevant $\left(\frac{V_{DS}}{V_{t}}\right)$ regions.

$$I_{SUB1} = AW_{1} \cdot \exp \left(-\frac{V_{X} - V_{TH}^{0} - V_{OFF} - \gamma V_{X1} + \eta (V_{DD} - V_{X})}{nV_{t}}\right)$$

Note that $A$, $n$, and $\eta$ can be calculated based on their BSIM equations or via fitting. The linearized body coefficient $\gamma$ can also be derived through fitting.

For the three transistor stack shown in Figure 2, there are two unknown inter-nodal voltages $V_{X1}$ and $V_{X2}$. To solve for these two unknowns, two equations in terms of the unknown voltages $V_{X1}$ and $V_{X2}$ can be generated. These equations can be derived by equating the subthreshold equations of the first and second transistor as well as the first and third transistor. The derivation is shown in sets of Equations (9)-(10) and (11)-(12).

$$I_{SUB1} = I_{SUB2}$$

$$AW_{1} \exp \left(-\frac{V_{X1} - V_{TH}^{0} - V_{OFF} - \gamma V_{X1} + \eta (V_{DD} - V_{X1})}{nV_{t}}\right) =$$

$$AW_{2} \exp \left(-\frac{V_{X2} - V_{TH}^{0} - V_{OFF} - \gamma V_{X2} + \eta (V_{DD} - V_{X2})}{nV_{t}}\right)$$

$$I_{SUB1} = I_{SUB3}$$

$$AW_{1} \exp \left(-\frac{V_{X1} - V_{TH}^{0} - V_{OFF} - \gamma V_{X1} + \eta (V_{DD} - V_{X1})}{nV_{t}}\right) =$$

$$AW_{4} \exp \left(-\frac{V_{X3} - V_{TH}^{0} - V_{OFF} - \gamma V_{X3} + \eta (V_{DD} - V_{X3})}{nV_{t}}\right)$$

Equation (10) and (12) can then be redefined in matrix form as shown in Equation (13) and solved to determine $V_{X1}$ and $V_{X2}$.

$$\begin{bmatrix} 1 + \gamma + 2 \cdot \eta + n \cdot m_{2} - (1 + \gamma + \eta + n \cdot m_{2}) \\ 1 + \gamma + \eta \end{bmatrix} \begin{bmatrix} \frac{V_{X1}}{V_{X2}} \\ \eta \cdot n \cdot m_{3} \end{bmatrix} = \begin{bmatrix} n \cdot v_{1} \left(\ln \left(\frac{w_{1}}{w_{2}}\right) - c_{2}\right) + \eta \cdot V_{DD} \\ n \cdot v_{1} \left(\ln \left(\frac{w_{1}}{w_{3}}\right) - c_{3}\right) + \eta \cdot V_{DD} \end{bmatrix}$$

For the four transistor stack, the subthreshold equation for the top transistor is equated with that of the second, third and fourth transistors. This results in having a system of three linear equations in terms of the unknown nodal voltages $V_{X1}$, $V_{X2}$ and $V_{X3}$. The system of equations for the four transistor stack is shown in Equation (14).

 Determining the variables $m_{n}$ and $c_{n}$ for the 3 and 4 transistor stacks will ultimately depend on the $V_{DS}/V_{t}$ ratio of the nth transistor in each stack. This paper proposes using a Bayesian classification [17] approach to accomplish this task. Consider the three transistor stack. For this scenario as mentioned earlier, the top transistor is always in the $V_{DS}/V_{t}$, $V_{TH}$
and from extensive simulations, it was observed that the middle transistor is always in the 0.6 ≤ \( V_{DS}/v_t \) < 4 region. So the goal is to determine the region where the \( V_{DS}/v_t \) ratio of the bottom transistor \( (V_{X2}/v_t) \) exists. The \( V_{DS}/v_t \) ratio of the bottom transistor can be in either the 0.6 ≤ \( V_{DS}/v_t \) < 4, or the 0.01 ≤ \( V_{DS}/v_t \) < 0.6 range. These regions are associated with classes (output variables) A and B respectively.

\[
\begin{bmatrix}
1 + \gamma + 2 \gamma n - m - m_2 & -1 - \eta - n - m_3 & 0 \\
1 + \gamma + \eta & \eta + n - m_3 & 1 - \eta - n - m_3 \\
1 + \gamma + \eta & \eta + n - m_4 & 0
\end{bmatrix}
\]

The next step is to determine the key features or input variables. Assuming that the technology and supply voltage are fixed, the widths of the three transistors \( (W_1, W_2, \text{and } W_3) \) can be used as input features to create a dataset. This involves simulating a three transistor stack in SPICE with randomly chosen transistor widths at a given \( V_{DD} \) and recording the \( V_{DS}/v_t \) ratio of each one of these entries. The output variable/class for the dataset is then determined by assigning each entry to the class representing the \( V_{DS}/v_t \) range in which it lies. Through simulations, it was found that 100 data points per class was adequate. As an example the first five entries are shown in Figure 3. In order to acquire accurate statistics about each class, the entries in the table are divided into smaller tables where all entries in each of these smaller tables correspond to one class. The sample mean vector \( \bar{m}_i \) and the sample covariance matrix \( \Sigma_i \) for each class \( i \) are then calculated. The steps described so far are those required for training and need to happen only once before the classification/estimation process is initiated. They are based on a fixed technology.

![Image](https://example.com/image.png)

**Fig. 3.** Training process for Bayesian inference.

Once the pre-extraction process is complete, the classification process may begin. Assume that we design a three transistor stack with normalized widths \( W_1 = 9.5, W_2 = 7.5 \) and \( W_3 = 8.0 \). This can be considered as a feature (input) vector \( x = \{9.5, 7.5, 8.0\} \). To determine whether the input vector belongs to either class A or B we evaluate the feature vector with the conditional density distribution \( P(x|W_i) \) of each class as shown in Equations (15) and (16), given their mean vectors and covariance matrices.

\[
P(x|W_A) = \frac{1}{(2\pi)^{\frac{D}{2}}|\Sigma_A|^{\frac{1}{2}}} \exp \left( -\frac{1}{2} (x - \bar{m}_A)^T \Sigma_A^{-1} (x - \bar{m}_A) \right)
\]

\[
P(x|W_B) = \frac{1}{(2\pi)^{\frac{D}{2}}|\Sigma_B|^{\frac{1}{2}}} \exp \left( -\frac{1}{2} (x - \bar{m}_B)^T \Sigma_B^{-1} (x - \bar{m}_B) \right)
\]

Feature (input) vector \( x \) then belongs to the class whose conditional density distribution \( P(x|W_i) \) is largest. This approach for determining the \( V_{DS}/v_t \) region may also be used for the 4 transistor stack. However, unlike the three transistor stack case, only the \( V_{DS}/v_t \) region for the top transistor is known; it is \( V_{DS}/v_t \geq 4 \) as in all the other cases. The other three transistors are either in the 0.01 ≤ \( V_{DS}/v_t \) < 0.6 region or the 0.6 ≤ \( V_{DS}/v_t \) < 4 region. This means that the number of classes goes up from \( 2^3 \) to \( 2^4 \). The rest of the procedure is identical to that described for the 2-class (3 transistor stack) scenario.

To ensure the suitability of the Bayesian Classification algorithm [18], its accuracy was evaluated using the probability of misclassification \( (P_{err}) \) error metric. First, a separate set of 1000 randomly chosen input vectors (composed of 3 and 4 normalized width values for the 3 and 4 transistor stacks respectively) were generated for the three and four transistor stacks. Two and eight classes were considered for the three and four transistor stacks respectively. Their "true" classes were determined from SPICE data. For example, for the three transistor stack with a particular set of widths, a SPICE operating point analysis was performed and \( V_{X2} \) is calculated. If \( V_{X2}/v_t \) was less than 0.6 for a particular set of widths, then this set of widths is associated with class A else class B. Each of these input vectors is then evaluated using the Bayesian Classification approach. The \( P_{err} \) of the Bayesian Inference approach w.r.t. to that of SPICE is then determined for the 3 and 4 transistor stack scenarios in each of the three process nodes considered in this paper. The results are shown in Table I and confirm the suitability of this approach. For more details on this approach to subthreshold leakage estimation the reader is referred to [(18), (19)].

**Table I**  
**ACCURACY OF THE BAYESIAN CLASSIFICATION SCHEME**

<table>
<thead>
<tr>
<th>Stack Length</th>
<th>Probability of misclassification ( (P_{err}) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>0.036 0.040 0.052</td>
</tr>
<tr>
<td>4</td>
<td>0.062 0.095 0.106</td>
</tr>
</tbody>
</table>

**B. Forward and Reverse Gate Tunneling Currents**

The BSIM4 equation for forward and reverse gate tunneling is too complicated and not practical for use in our estimation model. A simpler equation was chosen to model both forward and reverse gate tunneling. Gate tunneling data curves \( (I_{GFC} vs V_{GS} \text{ and } I_{RGC} vs V_{PGC}) \) extracted from SPICE simulations were fitted to the equation for different normalized widths. The goodness of fit metric (R) of the model to SPICE data for
forward gate tunneling as a function of $V_{GS}$ was found to be 0.979, 0.964 and 0.997 for 32nm, 45nm and 65nm respectively. And for reverse gate tunneling data (fitting against $V_{DG}$), $R$ was 0.9984, 0.9936 and 0.9961. The models used for fitting are:

$$I_{FGC} = (A_F \cdot W + B_F) \cdot V_{GS}^{C_F} \cdot \exp \left( -D_F \cdot V_{GS} \right) \tag{17}$$

and,

$$I_{RGC} = (A_R \cdot W + B_R) \cdot V_{DG}^{C_R} \cdot \exp \left( -D_R \cdot V_{DG} \right) \tag{18}$$

where $I_{FGC}$ and $I_{RGC}$ are forward and reverse gate tunneling currents respectively. $V_{GS}$ is the gate source voltage, $V_{DG}$ is the drain to gate voltage, $W$ is the width normalized with respect to the process’s minimum feature size and $A_F$, $A_R$, $B_F$, $B_R$, $C_F$, $C_R$, $D_R$ and $D_F$ are all process dependent fitting parameters.

C. Junction tunneling leakage

Because of the intractability of the BSIM4 equation for junction tunneling a simpler equation was used to model its effect in the 32, 45 and 65nm PTM processes:

$$I_{JUNC} = A \cdot V_{DB} \cdot \exp \left( -\frac{B}{V_{DB}} \right) \tag{19}$$

where $I_{JUNC}$ is the junction tunneling current, $V_{DB}$ is the drain to body voltage and $A$ and $B$ are fitting parameters determined through fitting the model to SPICE derived junction tunneling vs $V_{DB}$ curves. Fitting in this case was perfect ($R=1$) across all PTM processes considered.

D. Framework for Total Leakage Estimation in NMOS Stacks

The estimation framework starts with applying the input vector and supply voltage to the stack. If one or more transistors in the stack is OFF, the inter-nodal voltages $V_{Xn}$ between the OFF transistors and the subthreshold leakage across the sub-stack of OFF transistors are estimated. The $V_{GS}$ of each ON transistor and $V_{DG}$ of each OFF transistor are then assigned values of $V_{DD}$, 0 or $V_{DD} - V_{TH}$ based on the input vector and the relative transistor position in the stack. The derived $V_{GS}$ and $V_{DG}$ voltages for each transistor are then applied to Equations (17) and (18) respectively to determine the forward and reverse gate-tunneling leakage currents. Finally, junction tunneling is calculated whenever the drain is at a significantly high voltage i.e. ($V_{DD}$ or $V_{DD}V_{TH}$). All these leakage sources are then superposed to give the total leakage unless the input vector is an exception vector. Exception vectors include all vectors that have one or more ON transistors sandwiched between two OFF transistors in the stack. For the three transistor NMOS stack, ‘010’ is the only exception vector. In this case, the large OFF resistance of the top and bottom OFF transistors reduces the forward gate-leakage current exhibited by the middle ON transistor. This in turn causes the voltages at the internal nodes $V_{X1}$ and $V_{X2}$ to rise to a small but significant voltage. This voltage at the internal nodes is process dependent and will be referred to as an exception voltage $V_{ex}$. For the three-transistor NMOS stack, in the PTM 45nm process, $V_{ex} = 0.30V$ (for $V_{DD} = 0.9$). This voltage causes the $V_{GS}$ of the top transistor in the stack to become negative i.e. $V_{GS} = 0.30V$ and in turn the subthreshold leakage becomes negligible. Gate tunneling leakage is also reduced because $V_{GS} = V_{DD}V_{ex}$ for the middle (ON) transistor.

E. PMOS Modeling Considerations

The equations presented this far concern the NMOS case. Since this work is intended to be extended for use in static leakage estimation in combinational CMOS basic and complex gates, PMOS equivalent models were also developed. The equations required by the model for subthreshold leakage, gate tunneling leakage and junction tunneling for PMOS transistors are identical to that of the NMOS transistors but with the voltage polarities reversed. In the case of subthreshold leakage equation the PMOS version becomes:

$$I_{SUBP} = A \cdot W \cdot \exp \left( \frac{-V_{GSP} + V_{THP} + V_{OFFP} - \eta \cdot V_{DSP}}{\eta \cdot \nu_{t}} \right) \tag{20}$$

where $V_{GSP} = -V_{GSN}$, $V_{THP} = -V_{THN}$, $V_{DSP} = -V_{DSN}$ and $V_{OFFP} = -V_{OFFN}$. Alternatively one can also model the PMOS static leakage models using the NMOS static leakage model equations by using the absolute magnitudes of the voltages $V_{DGN}$, $V_{GSN}$ $V_{THN}$ and $V_{DBN}$. This is essentially
identical to negating all the voltages for the PMOS case. Having said that, it is important to note that it was necessary to generate a second set of SPICE simulations for fitting and dataset creation purposes for the PMOS model, and that the PMOS model is not simply a mirror image of the NMOS model.

In the literature [20] it was reported that gate tunneling leakage in PMOS transistors is negligible. In this work, it was found through simulation that gate tunneling and junction tunneling currents were smaller by approximately an order of magnitude than their NMOS counterparts but still substantial in the 32nm, 45nm and 65nm PTM processes and as such have all been included in the PMOS model.

Finally the framework presented for combining each leakage mechanism to provide the total leakage current can also be used for PMOS stacks. The only difference is that the exception vector for the three transistor stack in PMOS is ‘101’ instead of ‘010’. Also for PMOS only 1, 2 and 3 transistor stacks are considered and as such extracting the exception vectors in a 4-transistor PMOS stack is not required. The exception vector for the 3-transistor PMOS must be pre-characterized independently of the NMOS 3-transistor stack, since they are not a function of each other.

III. LOGIC GATE LEVEL LEAKAGE ESTIMATION

This section introduces the proposed total static leakage estimation approach in both basic and complex combinational CMOS gates by decomposing the gates into their constituent transistor stacks. The leakage exhibited by these stacks is estimated via a stack estimator tool, which is based on the transistor level leakage framework presented in the previous section. The results of each of the stack leakage estimations can then be superposed based on a set of rules to estimate the total static leakage of the gate. The stack estimator tool is capable of estimating leakage due to each mechanism in a stack as well as the total leakage of the stack.

Combinational CMOS gates are traditionally composed of an NMOS pull down transistor network and a PMOS pull-up network. For basic logic gates, each network is composed of either one or more transistor stacks, while for CMOS complex gates, each network may be composed of transistor stacks with possible inter-transistor nodal (ITN) branching. Stacks with ITN branching are referred to as complex stacks. The ITN branching in complex stacks gives rise to interconnected sub-stacks (See Figure 5). A pull down NMOS network composed of one (basic) stack and two complex stacks is shown in Figure 5. Note how each stack is connected to the output at one of its end nodes. This is a defining feature to what constitutes a stack or a complex stack as opposed to a sub-stack which is connected to at least one inter-transistor node.

A. Basic Logic Gates

Basic logic gates considered in this work include the inverter, 2 and 3-input NAND and NOR gates and the 4-input NAND gate. One inherent characteristic of CMOS topology is that in steady state, only one network (either pull-up or pull-down) maybe ON, whilst the complementary network is OFF.

An ON network has a serial path of ON transistors connecting the output to either ground in the case of a pull-down network or the voltage supply in the case of a pull-up network. At any one time subthreshold leakage will only occur in the OFF network.

For example, consider the 3-input NAND gate illustrated in Figure 6(a). It has an input vector of ‘001’, causing its PMOS network to be ON and its NMOS network to be OFF. In this case the total leakage (subthreshold, forward and backwards gate tunnelling and junction tunnelling) of the 3 transistor stack of the OFF pull-down network is superposed with only the gate tunnelling currents of the two ON single transistor stacks in the ON pull-up network. This is because both the output and the voltage supply are at $V_{DD}$ due to the two ON single transistor stacks in the pull-up network, causing the $V_{DS}$ of the single OFF transistor stack to be ‘0’ and therefore no subthreshold leakage current runs across it in accordance with Equation (1).
PMOS network is OFF. Consequently the total leakage of the 3-transistor stack of the OFF pull-up network is superposed with only the gate tunnelling currents of the ON single-transistor stack in the ON pull-down network. Again this is because the $V_{DS}$ across both of the parallel OFF transistors (in the pull-down network) is zero. Based on the above discussion, the subthreshold rule of estimating leakage current exhibited by a logic gate (basic or complex) from its constituent stacks is: If a network is ON, it does not exhibit subthreshold leakage.

B. Complex Logic Gates

The proposed methodology for estimating total leakage in complex gates will focus primarily on finding equivalent stacks for the pull-down/pull-up networks from which the total leakage exhibited by the entire network can be estimated accurately.

1) OAI/AOI Complex Logic Gates: Consider only the pull-down networks of the AOI32 and OAI23 complex gates shown in Figure 7 and Figure 8 respectively. The AOI pull-down network in Figure 7 is composed of two 3-transistor stacks. If no path of ON transistors exists between output and ground in either stack then both stacks will exhibit all leakage mechanisms considered in this work. As such, the stack estimator tool can be used to estimate leakage in both stacks independently as shown in Figure 7(a). These are then superposed to obtain total leakage in the pull-down network. If the input vector to one of the stacks is ‘111’ as shown in Figure 7(b), (i.e. an ON path exists between output and ground) then the subthreshold leakage must be omitted in accordance with the subthreshold rule.

The topology of an OAI pull-down network is illustrated in Figure 8. Consider the case when the inputs to all parallel transistors at each level of the stack are identical (Figure 8.(a)). In this case, the OAI network can be collapsed into a single equivalent stack with the same identical input vector. The normalized widths of each transistor in the equivalent stack is determined by summing the normalized widths of the two (or more) parallel transistors occupying the same location in the OAI network as shown in Figure 8(a). The stack estimator tool can then be used to estimate the leakage in this equivalent stack; which is ultimately the total static leakage of the pull-down network.

In the case when the inputs to all parallel transistors at each level of the stack are not identical, again an equivalent stack can be realized. However, this equivalent stack is now composed of the transistor path with the maximum number of ON transistors. Essentially the ON transistors act as “short-circuits” that provide current flow with an alternate path around any parallel OFF transistors. If two or more parallel transistors in the pull-down network have the same potential ($V_{DD}$ or ground) at their gates then the normalized width of the transistor representing them in the equivalent stack is the summation of their normalized widths. Examples of this are shown in Figure 8(b-d).

2) Hybrid Complex Gates: The pull-up and pull-down networks of hybrid complex gates can also be collapsed into one or more transistor stacks, depending on their topology and input vector. Deriving the equivalent transistor stack can be accomplished via three steps. The first one involves collapsing all transistors that are in parallel with each other in a similar way as was accomplished for the OAI complex gates.

After all parallel transistors have been collapsed; one must also consider parallel sub-stacks (See Fig. Figure 9(c)). If at least one of these sub-stacks has all ON transistors (i.e. the sub-stack is ON), then it will again short circuit all of the other parallel OFF sub-stacks. This allows for the removal of these OFF sub-stacks.
from the output to either ground (pull-down network) or the supply (pull-up network). Each one of these paths constitutes an equivalent stack. An example of this three step algorithm is shown in Figure 9. The case of Figure 9(a) illustrates the schematic of a rather large pull-down network. The first and second steps are shown in Figures 9(b) and (c); they illustrate the collapsing of all parallel transistors and sub-stacks depending on whether they are ON and/or OFF. The third step is shown in Figure 9(d) where the equivalent stacks are traced across all remaining transistor paths between output and ground. Once the equivalent stacks are extracted, subthreshold leakage is estimated across each one and summed. Junction tunnelling and gate tunnelling leakages are estimated on a per-transistor basis and added to the estimated subthreshold leakage to obtain the total leakage of the pull down network.

The methodology introduced here can easily be applied to pull-up networks as well. Once total leakage in the pull-up and pull-down networks of any complex gate (AOI, OAI or hybrid) are estimated, they can then be superposed to estimate total leakage of the entire gate.

IV. LOGIC BLOCK LEVEL LEAKAGE ESTIMATION

The total static leakage estimation of basic and complex logic gates can be extended to the logic block level by means of event driven simulation. Initially, the logic block described in HDL is mapped into a set of virtual library components as dictated by generic topological criteria. The outcome of the mapping is an optimized netlist of the logic block and a database for a soft version of a selected set of virtual library components. These library components, which consists of complex and/or basic gates, are sized based on the technology and design constraints [3], [21]. Input vectors to the logic block are then propagated through the gate netlist via event driven simulation. For every logic block input vector, a set of corresponding input vectors to each of the block’s constituent gates is determined. At this point the logic gate level leakage estimation approach proposed in the preceding section is applied; taking the transistor netlist and the input vector information of each synthesized gate as its inputs and creating a total static leakage estimate (using the algorithms presented earlier) that is a function of the input vector of the logic block. This process is depicted in Figure 10.

The fact that the virtual library based framework enables pattern dependent static power estimation of a logic block ‘on-the-fly’, can also be used to facilitate input vector control [14], [23], [24]. The input vector exhibiting the least leakage in a logic block or logic gate may be applied to that logic block or logic gate in standby mode to reduce static power dissipation.

V. RESULTS

The performance of the proposed static leakage estimation approaches at the transistor level, logic gate level and the logic block level are presented in this section. At each level the model was setup and coded in MatLab. The results produced by the model were then compared to the results produced by SPICE simulations using ultra deep submicron PTM based processes. The following sections explain the experimental setup and document the model accuracy with respect to SPICE at each level.

A. Transistor Level

Scripts for the two, three and four transistor stacks based on the models and framework described in this paper were coded in MatLab. The results of these models were then compared to that of SPICE simulations using the 65nm, 45nm and 32nm BSIM4-based PTM processes. For each stack length, 1000 different scenarios with randomly selected transistor widths (chosen within a range of normalized widths W = 2 to 10) were applied to the proposed model and to SPICE simulations. The supply voltage was fixed at 0.9V for all three technologies. The results are tabulated in Table II for NMOS and PMOS.

<table>
<thead>
<tr>
<th>Stack Length</th>
<th>Average % error</th>
<th>Variance in average % error</th>
</tr>
</thead>
<tbody>
<tr>
<td>32nm</td>
<td>N P N P N P N P</td>
<td>32nm 45nm 65nm</td>
</tr>
<tr>
<td>1</td>
<td>1.0 0.7 1.0 0.7</td>
<td>1.0 0.7 1.0 0.7</td>
</tr>
<tr>
<td>2</td>
<td>2.7 2.7 2.7 2.7</td>
<td>2.7 2.7 2.7 2.7</td>
</tr>
<tr>
<td>3</td>
<td>7.0 7.0 7.0 7.0</td>
<td>7.0 7.0 7.0 7.0</td>
</tr>
<tr>
<td>4</td>
<td>11.8 NA 11.8 NA</td>
<td>11.8 NA 11.8 NA</td>
</tr>
</tbody>
</table>

The runtime required by the model was also measured and compared to that of SPICE. The runtime values recorded in Table III are for a single stack estimation. It can be seen from the results that the proposed model is at least 89.70 times faster than SPICE (for the four transistor stack). For the two transistor stack the proposed model is 116 times faster than SPICE. Note that the model runtime increases significantly from the 2 transistor stack to 3 transistor stack. This is primarily because of increased processing requirements of the Bayesian classification approach utilized in the subthreshold leakage estimation.
The robustness of the NMOS transistor level leakage estimation model was evaluated against two process parameters: (i) supply voltage and (ii) temperature variations for all stack lengths. Unlike the experimental setup used to determine the values in Table II, the transistor widths were fixed and the average error was calculated across each stack’s 2^N input vectors. The results for the voltage variations and the temperature variations are provided in Tables IV and V.

### TABLE III
**Runtime of Proposed Model vs SPICE (45nm)**

<table>
<thead>
<tr>
<th>Stack Length</th>
<th>Model (ms)</th>
<th>SPICE (ms)</th>
<th>Speedup</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.28</td>
<td>26.02</td>
<td>92.93</td>
</tr>
<tr>
<td>2</td>
<td>0.33</td>
<td>38.29</td>
<td>116.03</td>
</tr>
<tr>
<td>3</td>
<td>0.43</td>
<td>40.97</td>
<td>95.28</td>
</tr>
<tr>
<td>4</td>
<td>0.55</td>
<td>47.34</td>
<td>98.00</td>
</tr>
</tbody>
</table>

### TABLE IV
**Percentage error as a function of Stack length and temperature (V_{DD} = 0.9V and using the 45nm process)**

<table>
<thead>
<tr>
<th>Stack Length</th>
<th>Temp = 0°C</th>
<th>Var in %Err</th>
<th>Mean %Err</th>
<th>Var in %Err</th>
<th>Mean %Err</th>
<th>Var in %Err</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3.0</td>
<td>0.3</td>
<td>0.3</td>
<td>0.0</td>
<td>3.5</td>
<td>0.4</td>
</tr>
<tr>
<td>2</td>
<td>2.4</td>
<td>1.8</td>
<td>0.0</td>
<td>0.0</td>
<td>5.3</td>
<td>14.2</td>
</tr>
<tr>
<td>3</td>
<td>1.4</td>
<td>1.0</td>
<td>1.5</td>
<td>1.8</td>
<td>6.2</td>
<td>19.4</td>
</tr>
<tr>
<td>4</td>
<td>3.4</td>
<td>10.5</td>
<td>3.4</td>
<td>19.3</td>
<td>8.8</td>
<td>86.7</td>
</tr>
</tbody>
</table>

### TABLE V
**Percentage error as a function of Stack length and V_{DD} (For the 45nm process at nominal temperature)**

<table>
<thead>
<tr>
<th>Stack Length</th>
<th>V_{DD} = 0.7V</th>
<th>V_{DD} = 0.9V</th>
<th>V_{DD} = 1.1V</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean %Err</td>
<td>Var in %Err</td>
<td>Mean %Err</td>
<td>Var in %Err</td>
</tr>
<tr>
<td>1</td>
<td>3.8</td>
<td>10.2</td>
<td>1.1</td>
</tr>
<tr>
<td>2</td>
<td>3.6</td>
<td>3.0</td>
<td>0.9</td>
</tr>
<tr>
<td>3</td>
<td>7.1</td>
<td>51.1</td>
<td>3.5</td>
</tr>
<tr>
<td>4</td>
<td>8.3</td>
<td>20.5</td>
<td>2.9</td>
</tr>
</tbody>
</table>

The results in Table IV indicate that even with a variation of up to 0.2V in VDD, the accuracy of the model is still reasonable. The accuracy is obviously compromised but under the principle of graceful degradation. This is also true in the case of temperature variations.

It is important to note that the pre-characterization process (fitting of the equations of each leakage mechanism to SPICE data and collection of SPICE data for the Bayesian approach to determine the V_{DS}/v_t region of each transistor in the stack) was performed under a fixed set of conditions for each process, i.e. V_{DD} = 0.9V, and Temperature = 27°C. This is why the smallest errors exhibited by the voltage variation analysis and the temperature variation analysis happened at V_{DD} = 0.9V and Temperature = 27°C respectively. For optimized results under another particular set of conditions, pre-characterization must be based on SPICE simulations performed under those set of voltage and temperature conditions.

The traditional approach utilized for modeling process variations in the ASIC design paradigm involves the model analysis and verification at all four process corners. Process variations can also be incorporated into the model by deriving an optimized version of the model for each corner, through performing pre-characterization for each corner.

### B. Logic Gate Level

Evaluation of static leakage estimation with the logic gate level methodology was broken down into three major sections; i) evaluation of the basic logic gates, ii) complex OAI/AOI gates and iii) complex hybrid gates. In all three cases the transistors were sized to maintain equal transition times following the logical effort paradigm [25].

![Fig. 11. Percentage error statistics exhibited by the basic logic gate estimation model w.r.t SPICE. (a) 32nm, (b) 45nm, and (c) 65nm.](image)

For basic conventional CMOS logic gates, models for the inverter, 2-input NOR and NAND, 3-input NOR and NAND and the 4-input NAND were coded in MatLab in accordance with the proposed methodology. The results provided by the model were then compared to SPICE simulations based on the 32nm, 45nm and 65nm process nodes. The average percentage error was recorded as a function of all 2^N input vectors is provided in Figure 11.

For the performance evaluation of the proposed methodology applied to complex OAI/AOI and hybrid complex gates, several circuits were coded in MatLab. The results provided by the model were then compared to SPICE simulations. For each logic gate, the percentage error was recorded as a function of all 2^N input vectors; where N is the number of logic gate inputs. The results for AOI/OAI complex gates are provided in Figure 12. The reported average percentage error for OAI complex gates was slightly larger than for their respective AOI complements.

The five hybrid complex gates shown in Figure 13 were also evaluated. Average percentage error results compared to
SPICE for these gates are presented in Figure 14. Note that the adder without carry component (ADDWOC) and carry component (CC) complex gates constitute a complete CMOS full adder [26] and are typically used together.

C. Logic Block Level

For the evaluation of the logic block level scheme, five MCNC benchmark circuits were utilized: C17, CM138a, B1, CM42a, DECOD. Three versions of the B1 MCNC benchmark circuit were synthesized: (i) B1-basic consisting of only basic logic gates, (ii) B1-complex consisting of only complex gates and (iii) B1-combined consisting of a combination of complex and basic logic gates. Results are shown in Figure 15. Static leakage for each circuit was modelled in SPICE and according to our proposed model using PTM’s 45nm process. In our model each possible input vector to the circuit was allowed to propagate through the circuit. The inputs to each of the constituent gates were then determined. These were then used, along with transistor/stack topology information, to estimate the total leakage of each gate on-the-fly. The leakage of all the constituent gates of the circuit were then summed to provide an estimate of the total static leakage exhibited by the circuit, for a particular circuit input vector. The largest maximum error (1.55%) was exhibited by the complex gate version of the B1 circuit while the smallest maximum error (0.25%) was exhibited by the DECOD circuit. A runtime analysis was also performed and the results shown in Table VI, demonstrate that the proposed model provides significant runtime savings. Note that the time for event-driven simulation is inherent in the model runtimes presented.

VI. CONCLUSION

This paper presented a complete bottom-up approach to static leakage estimation in UDSM transistors with the ultimate goal of estimating total static leakage in logic blocks. The approach can be easily extended to IP blocks. Frameworks for the transistor level, basic and complex gate levels and for high level static leakage estimation were presented. Each of these frameworks relies on the existence of the framework of the model preceding it. At the logic gate level, both basic and complex gates are considered and their leakage can be estimated ‘on-the-fly’ based on their input vectors and transistor/stack topology. This facilitates static leakage estimation in a library free synthesis based design flow. It also offers good accuracy and significant runtime savings in comparison to SPICE.

Each framework was modelled in MatLab and evaluated with respect to SPICE using PTM model files for UDSM process nodes. The approach exhibited a worst case mean error of 1.1% with respect to SPICE simulations across seven MCNC benchmark circuits. It also provided significant runtime savings over SPICE.

Finally, it is also worth mentioning that the analytical models presented here can also be used for circuit optimization.

<table>
<thead>
<tr>
<th>MCNC Circuit</th>
<th>SPICE (ms)</th>
<th>Model (ms)</th>
<th>Speedup</th>
</tr>
</thead>
<tbody>
<tr>
<td>C17</td>
<td>222.82</td>
<td>3.86</td>
<td>57.73</td>
</tr>
<tr>
<td>CM138a</td>
<td>794.36</td>
<td>20.19</td>
<td>39.34</td>
</tr>
<tr>
<td>B1-basic</td>
<td>347.50</td>
<td>9.78</td>
<td>35.53</td>
</tr>
<tr>
<td>B1-combined</td>
<td>410.16</td>
<td>9.55</td>
<td>42.95</td>
</tr>
<tr>
<td>B1-complex</td>
<td>204.40</td>
<td>5.65</td>
<td>36.18</td>
</tr>
<tr>
<td>CM42a</td>
<td>317.98</td>
<td>5.56</td>
<td>57.19</td>
</tr>
<tr>
<td>DECOD</td>
<td>995.67</td>
<td>27.34</td>
<td>36.42</td>
</tr>
</tbody>
</table>
while the estimation methodology for logic blocks can be used effectively to determine the input vector which results in minimum leakage current during the standby mode of operation.

REFERENCES


Côme Rozon received the M.Sc. degree (June 1977) in solid state physics from Sherbrooke University, Sherbrooke, Quebec, Canada, and the Ph.D. degree (August 1987) in Electrical Engineering from Queen’s University, Kingston, Ontario, Canada. From 1975 to 1983 he served as a Combat Systems Engineer in the Royal Canadian Navy and retired at the rank of Naval Lieutenant. In 1983 he joined the teaching staff of the Electrical & Computer Engineering Department at the Royal Military College of Canada, Kingston, Ontario. He has worked as a consultant for Newbridge Networks and Nortel Networks. He held the position of Director of Computing Services at the Royal Military College of Canada and also served as Head of the Department of Electrical and Computer Engineering. He is now a retired emeritus professor who still conducts research works in nanotechnologies, circuit design and design for testability.