Runtime Leakage Minimization Through Probability-Aware Optimization

Dongwoo Lee, *Student Member, IEEE*, David Blaauw, *Member, IEEE*, and Dennis Sylvester, *Member, IEEE*

***Abstract*—Runtime leakage current, defined as circuit leakage during normal operation (i.e., nonstandby mode), has become a major concern in very advanced technologies along with traditional standby mode leakage. In this paper, we propose a new leakage reduction method that specifically targets runtime leakage current. We first observe that the state probabilities of nodes in a circuit tend to be skewed, meaning that they have either high or low values. We then propose a method that exploits these skewed state probabilities by setting only those transistors to high-V_t (thick-oxide) that have a high likelihood of being OFF (ON) and, hence, contributing significantly to the total runtime leakage. Accordingly, we also propose a library specifically tailored to the proposed approach, where V_t and T_{ox} assignment with favorable tradeoffs under skewed input probabilities is provided. For further leakage reduction, we also introduce circuit resynthesis using pin reordering, pin rewiring, mapping, and decomposition. The optimization algorithm shows substantial leakage improvement over probability unaware optimization using a traditional standard cell library.**

***Index Terms*—Circuit resynthesis, dual oxide thickness, dual threshold voltage, gate leakage, leakage current, power optimization, runtime mode, state assignment, state probability, subthreshold leakage.**

I. Introduction

In RECENT years, leakage power has become a significant concern as process dimensions and supply voltage continue to scale down. Up to 54% of the total power dissipation is projected to be subthreshold leakage power dissipation at the 65-nm node [1]. To address subthreshold leakage current ($I_{\rm sub}$) in standby mode, the multi-threshold CMOS (MTCMOS) approach was proposed where a high-V_t gating transistor is inserted in series with the power supply [2]. This method incurs routing overhead for virtual power supplies and requires special latches to preserve state in standby mode [3]. In a different approach, a dedicated sleep input vector that minimizes $I_{\rm sub}$ is assigned to a circuit in standby mode [4]. This approach uses modified flip-flops that force the output to the required state [5]. However, $I_{\rm sub}$ reduction is small in this case due to logical correlations—typically in the range of 10%–30% [6]. A substrate reverse body bias can be applied to control V_t for $I_{\rm sub}$ minimization when using a triple well technology. In addition to the overhead of substrate bias generators, the diminishing body coefficient with process scaling makes this approach less effective [7], [8].

The dual-V_t approach reduces $I_{\rm sub}$ by assigning transistor threshold voltages using a process where both high- and low-V_t transistors are available. To reduce leakage current, noncritical gates in the circuit are assigned to high-V_t, while critical circuit portions are assigned to low-V_t [9]–[11]. This approach was extended for standby mode operation by combining the V_t assignment with sleep state assignment using a branch and bound search method [12]. This method is based on the observation that, given a known input state for a gate, the subthreshold leakage current of that gate can be reduced by setting only OFF transistors on each path from $V_{\rm dd}$ to *Gnd* to high-V_t since only OFF transistors contribute to $I_{\rm sub}$. In this way, [12] improves the tradeoff between leakage and performance compared to V_t assignment with unknown input states where most or all of the transistors must be set to high-V_t before a significant improvement in $I_{\rm sub}$ is observed. However, while this approach significantly improves the leakage current in standby mode, it is not directly applicable to runtime leakage while the circuit is operating since the circuit state is not known in this case.

In recent technologies, the gate tunneling leakage current ($I_{\rm gate}$) has become comparable to $I_{\rm sub}$. While continued scaling of the gate oxide layer thickness ($T_{\rm ox}$) is necessary to provide substantial current drive at reduced supply voltage, it leads to significant gate tunneling leakage current. One technique to reduce $I_{\rm gate}$ is to apply pin reordering in standby mode [13]. Since $I_{\rm gate}$ depends strongly on the position of the ON/OFF transistors, $I_{\rm gate}$ can be greatly reduced by placing OFF transistors at the bottom of the stack. While this method is quite effective for $I_{\rm gate}$ reduction, it also cannot be applied to runtime leakage reduction, since it is unknown at design time which transistors are OFF in runtime. Furthermore, it is desirable that a leakage reduction technique act equally well on both $I_{\rm gate}$ and $I_{\rm sub}$ components since they can both be significant.

Other prior work has shown that pin rewiring as well as reordering, can reduce dynamic circuit power dissipation. While pin reordering swaps pins within a single gate, pin rewiring swaps functionally equivalent pins across gates. Dynamic power can be reduced by lowering the transition density at gates with high loading using pin rewiring [14]. However, this algorithm is not applicable to leakage power reduction.

Traditionally, runtime leakage power has been of less concern than standby mode leakage since in runtime dynamic power dissipation has been significantly greater than static power dissipation. This is no longer true in aggressively scaled processes such as 65 nm, particularly in high-performance processor designs [15]. Therefore, new approaches for reducing leakage power

Manuscript received June 28, 2005; revised January 6, 2006. This work was supported in part by the National Science Foundation, by the Semiconductor Research Corporation, and by the Gigascale Systems Research Center/Defence Advanced Research Projects Agency.

D. Lee is with the Memory Division, Samsung Electronics Company Ltd., Gyeonggi-Do 445-701, Korea (e-mail: dongw.lee@samsung.com).

D. Blaauw and D. Sylvester are with the Electrical Engineering and Computer Science Department, University of Michigan, Ann Arbor, MI 48109 USA (e-mail: blaauw@umich.edu; dmcs@umich.edu).

Digital Object Identifier 10.1109/TVLSI.2006.884149

in runtime mode are needed. In this paper, we propose a new method to reduce leakage current in runtime mode. Our approach leverages dual-V_t/dual-$T_{\rm ox}$ technology and performs simultaneous circuit sizing, $V_t/T_{\rm ox}$ assignment, and circuit resynthesis. In order to improve the leakage/performance tradeoff, we exploit the state probabilities of nodes in the circuit during runtime, combined with a specially tailored cell library that takes advantage of frequently occurring skewed gate input probabilities. In general, knowledge of expected state probabilities of nodes in a circuit allow leakage optimization techniques similar to the standby techniques described above (from [12], for instance) to be applied, with slightly smaller improvements expected depending on the node statistics. This will be discussed in more detail in the following sections. Node state probabilities are typically determined during the design phase based on extensive functional gate-level simulations of expected program loads. We also exploit circuit resynthesis techniques such as pin reordering, pin rewiring, mapping, and decomposing gates. Mapping and decomposition [16] may be applied in order to obtain increased stack effect [17], [18], where multiple transistors are turned OFF in series, thereby minimizing $I_{\rm sub}$. The combined effect of these runtime leakage optimization techniques is shown to be large, with savings of 57% on average for a range of benchmark circuits compared to traditional leakage reduction methods in a predictive 65-nm technology.

In this paper, we use existing dual-V_t and dual-$T_{\rm ox}$ processes, which are already widely available and used [19]. These processes require additional masks and process steps, which will vary depending on the exact process used. Our leakage optimization method does not require any additional modifications (except possibly having to space out series connected transistors - this will be discussed in detail in Section VII) and, hence, is transparent from a process point of view.

II. Overview of the Approach

A. Leakage Dependence on State Probability

In this section, we discuss the calculation of leakage current in runtime mode using state probabilities. It is well known that the leakage current of a gate depends on the input state of that gate. For example, the leakage currents (including both $I_{\rm sub}$ and $I_{\rm gate}$) of a simple NAND2 gate are shown in Fig. 1 for all input states. The minimum leakage current at "00" state is only 15% of the maximum leakage current at "11" state (Table I). In this paper, we use BSIM4 models of a predictive 65-nm process with typical process conditions. Experiments are performed at 1-V operating voltage and room temperature.

However, in runtime mode the input state of a gate is unknown. Therefore, we compute the leakage current of a gate using the state probabilities of the gate inputs. The state probability of a node is the probability of that node being in a high state. If we know the state probabilities of the input nodes in a gate, we can determine the probability that a gate will be in each state in runtime mode. For example, if two inputs, A and B, for the NAND2 gate in Fig. 1 have 0.8 and 0.2 as their state probabilities, respectively, then assuming that the state probabilities are independent the probability of the state AB = "10" is

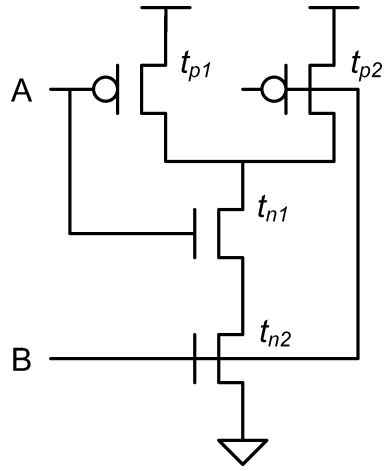


Fig. 1. Simple NAND2 gate.

TABLE I
Leakage Current of NAND2 Gate

Input state (AB)	Leakage current (nA)
00	41.2
01	146.6
10	91.8
11	270.4

$P_{\rm A}\times(1-P_{\rm B}) = 0.8\times(1-0.2) = 0.64$. While we assume independent input state probabilities for the purpose of illustration, the implemented computation can account for correlations between the gate input state probabilities using methods described in [20].

Based on the calculated probability of each state, the leakage current of a gate can be calculated using the following equation:

$$I_{\rm leak} = \sum_k P_k \times I_{{\rm state},k}.$$

In the previous equation, k is over all possible input states in the gate, P_k is the probability of state k and $I_{{\rm state},k}$ is the leakage current of state k. If the NAND2 gate in the previous example has the leakage current values in Table I with the given input state probabilities, the expected leakage current of this NAND2 gate is 114.5 nA.

B. Input State Probability Distribution

In this section, we demonstrate that node state probabilities show a bi-modal distribution, meaning that some nodes have a high state probability while other nodes have a low state probability. This is intuitively clear when we consider the propagation of probabilities through simple logic gates. For instance, if we evaluate a three-input AND gate where all input have an input state probability of 0.5, assuming independence, the state probability of the gate output is $0.5^3 = 0.125$. Hence, state probabilities tend to diverge to high or low values as they propagate through the circuit. This is also illustrated in Fig. 2, which shows the state probabilities of the primary inputs (PIs), primary outputs (POs), and internal nodes of MCNC benchmark circuit i10.

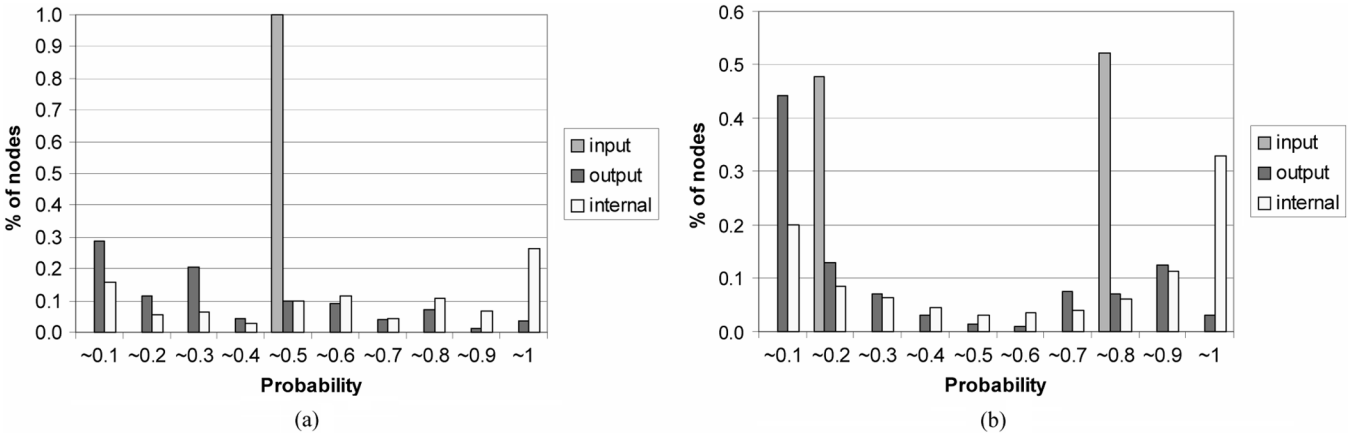


Fig. 2. State probabilities of i10 circuit. (a) Probabilities of $\mathrm{primary\ inputs} = 0.5$. (b) Probabilities of $\mathrm{primary\ inputs} = 0.2$ or 0.8.

In Fig. 2(a), all inputs have $P = 0.5$. However, the state probabilities of outputs and internal nodes are not centered at 0.5 but show a bi-modal distribution. In Fig. 2(b), where all inputs have a state probability of either 0.2 or 0.8, the state probabilities of outputs and internal nodes remain lower or higher than those of the inputs. Since the outputs of a circuit will act as inputs to another circuit, it is clear that for such a circuit block the typical state probability for the inputs can be expected to lie in the ranges of $P = 0.1$–0.2 or 0.8–0.9. In our analysis, we, therefore, use three state probabilities for primary inputs: 1) all PIs have $P = 0.5$; 2) half the PIs have a lower probability of 0.2 and the rest have a higher probability of 0.8; and 3) which is identical to 2) but uses probabilities $P = 0.9$ and $P = 0.1$.

C. Problem Formulation

In this section, we overview our proposed minimization method for both I_{sub} and I_{gate} in runtime mode. The objective of the proposed approach is to find the minimum leakage current of a circuit while meeting a specific delay criterion. Starting from the worst performance point (the lowest leakage current point), we move to the best performance point (the highest leakage current point). The worst performance point is achieved using all high-V_t and thick-T_{ox} transistors with minimum size. Circuit resynthesis is also performed to optimize the circuit for lowest leakage. Every optimization move is performed using a sensitivity-based algorithm where leakage current values are calculated based on gate input state probabilities. The best sensitivity move achieves the highest delay improvement with the lowest increased leakage current, and is chosen at each optimization step. Each chosen move can be one of four of the following possibilities: 1) selecting cells from the library with a more speed aggressive V_t/T_{ox} assignment; 2) rewiring functionally symmetric pins; 3) circuit modification by mapping or decomposition; and 4) increasing the gate size. Pin reordering is combined with the V_t/T_{ox} assignment step. A specially tailored cell library is used in our approach, and is discussed in Section IV. The optimization approach is described in more detail in the following section.

III. Leakage Reduction Techniques

A. Probability-Aware V_t/T_{ox} Assignment Algorithm

In this section, we review how V_t/T_{ox} assignment is performed for leakage minimization with a known input state and then show how V_t/T_{ox} assignment can be combined with state probabilities for runtime leakage reduction.

The V_t/T_{ox} assignment algorithm used in our leakage minimization approach is based on the key observation that given a known input state, a transistor need not be assigned both a high-V_t and a thick-oxide. If a transistor is OFF, gate leakage is significantly reduced since there is no gate-to-channel tunneling component and, hence, the transistor only needs to be considered for high-V_t assignment. Conversely, a transistor that is ON given a particular input state may exhibit significant I_{gate}, but does not impact I_{sub}. Hence, conducting transistors only need to be considered for thick-oxide assignment. If the input state is unknown, it cannot be predicted at design time which transistors will be ON or OFF and, therefore, all or most transistors must be assigned to both high-V_t and thick-oxide in order to significantly reduce the total average leakage. This degrades the obtained leakage/delay tradeoff relative to the case where input state is known.

Similar to our work in [12] and [21], we introduce so-called groups, which are the minimum sets of transistors that need to be set to high-V_t or thick-oxide to reduce leakage in a particular state. For instance, in a stack of several OFF transistors, only one transistor needs to be assigned to high-V_t to effectively reduce the total I_{sub}. Similarly, I_{gate} for transistors in a stack exhibits a strong dependence on their position. If a conducting transistor is positioned above a nonconducting transistor in a stack, its V_{gs} and V_{gd} will be small and gate leakage will be significantly reduced [13]. Hence, depending on the input state, only a small subset of all ON transistors needs to be assigned thick-oxide and only a subset of all OFF transistors need to be considered for high-V_t assignment.

We consider the leakage and performance of a simple NAND2 gate shown in Fig. 3. Fig. 3(a) is the NAND2 gate when all transistors are assigned to both low-V_t and thin-oxide for best performance. Fig. 3(d) shows the NAND2 gate with minimum leakage by assigning all transistors to high-V_t and thick-oxide. Fig. 3(b) and (c) are examples of V_t/T_{ox} assignment using the

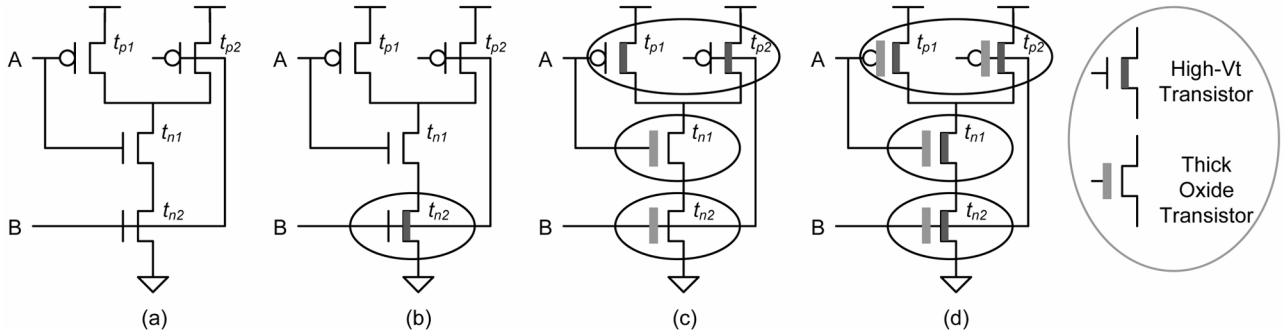


Fig. 3. Concept of group at NAND2 gate.

TABLE II
LEAKAGE CURRENT OF NAND2 GATE

Input state (AB)	Leakage current (nA)			
	Best performance	Minimized leakage		Lowest leakage
	with all low-V_t and thin-oxide assignment - Figure 3 (a)	Group assignment		with all high-V_t and thick-oxide assignment - Figure 3 (d)
		Figure 3 (b)	Figure 3 (c)	
00	41.2	14.0	26.9	2.4
01	146.6	137.1	75.2	8.1
10	91.8	13.3	64.1	4.6
11	270.4	260.9	19.5	10.7

TABLE III
NORMALIZED DELAY OF NAND2 GATE

	NAND2 gate in Figure 3	Rise delay		Fall delay	
		Pin A	Pin B	Pin A	Pin B
Best performance	(a)	1.00	1.00	1.00	1.00
Group assignment	(b)	1.00	1.00	1.12	1.16
	(c)	1.36	1.37	1.27	1.27
Lowest leakage	(d)	1.92	1.93	1.78	1.79

group concept with knowledge of the input state. For instance, when both inputs of the NAND2 gate are "1," both PMOS transistors are OFF and, hence, contribute to $I_{\rm sub}$. At the same time, since both NMOS transistors are conducting, they contribute to significant $I_{\rm gate}$. A group-assigned NAND2 gate in Fig. 3(c) improves the leakage/performance tradeoff for the "11" state. $I_{\rm sub}$ through the PMOS devices is reduced by high-V_t assignment, and $I_{\rm gate}$ of NMOS devices is minimized by thick-oxide assignment. With an "11" input state, the leakage current of the group-assigned NAND2 gate is nearly as low as that of an all high-V_t and thick-oxide implementation [19.5 nA in Fig. 3(c) versus 10.7 nA in Fig. 3(d)], as shown in Table II, and is much reduced compared to the all low-V_t and thin-oxide case [270.4 nA in Fig. 3(a)]. At the same time, the group-assigned NAND2 gate does not incur as large a delay impact as the minimum leakage assignment. Table III shows the normalized rise/fall 50% delays of each NAND2 gate in Fig. 3. The lowest leakage NAND2 gate has a 92% (78%) impact on rise (fall) delay, whereas the group assigned gate in Fig. 3(c) has only a 37% (27%) impact compared to the best performance case. With input state "AB" = "00" or "10," our $V_t/T_{\rm ox}$ assignment algorithm selects the NAND2 gate in Fig. 3(b). $I_{\rm sub}$ in the NMOS transistor stack is reduced using high-V_t assignment on transistor t_{n2}. In this case, since all other transistors (t_{p1}, t_{p2}, and t_{n1}) remain as low-V_t and thin-oxide, delay impact is minimized (Table III).

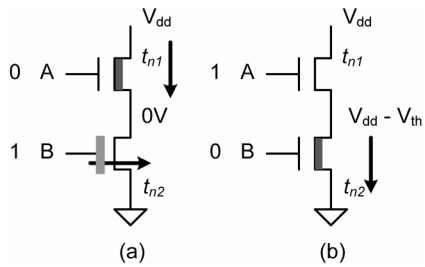


Fig. 4. Pin reordering.

The magnitude of $I_{\rm sub}$ primarily depends of the number of OFF versus ON transistors in a stack, while $I_{\rm gate}$ also depends strongly on the position of the ON/OFF transistors. Using this property, $I_{\rm gate}$ can be reduced by changing the position of the ON/OFF transistors. In Fig. 4(a), when an NMOS stack of a

TABLE IV
LEAKAGE CURRENT WITH DIFFERENT $V_T/T_{\rm OX}$ ASSIGNMENT FOR NAND2 GATE WITH $P_A = 0.8$ AND $P_B = 0.8$

Leakage current [nA]			
Best performance	Minimized leakage		Lowest leakage
with all low-V_t and thin-oxide assignment - Figure 3(a)	Group assignment		with all high-V_t and thick-oxide assignment - Figure 3(d)
	Figure 3(b)	Figure 3(c)	
212.8	191.6	35.8	9.0

NAND2 gate has "01" values at its input pins "AB," it has large $I_{\rm gate}$ due to the large $V_{\rm gd}$ and $V_{\rm gs}$ of the ON transistor t_{n2} as well as a large $I_{\rm sub}$ due to the OFF transistor t_{n1}. In order to effectively reduce the leakage with this input state, the NMOS transistor t_{n1} must be assigned to high-V_t and the NMOS transistor t_{n2} must be assigned to thick-oxide. However, if the two symmetric input pins A and B are swapped (reordered), the NMOS stack will exhibit very small $I_{\rm gate}$ since the ON transistor t_{n1} experiences a small $V_{\rm gd}$ and $V_{\rm gs}$ as shown in Fig. 4(b). After reordering the input pins, it is necessary to set only the NMOS transistor t_{n2} to high-V_t without any thick-oxide assignment. In general, if we place OFF transistors at the bottom of a stack by reordering input pins we can minimize $I_{\rm gate}$ of a gate in standby mode. Note that pin reordering will impact the delay of the circuit and, hence, some performance penalty may be incurred. However, this penalty will be offset by the elimination of the thick-oxide assignment in the pull-down stack. In this paper, we, therefore, consider pin reordering combined with $V_t/T_{\rm ox}$ assignment. By using pin reordering, we can minimize leakage current and also reduce the number of needed library cells, since a cell for the "01" state is no longer necessary. For the leakage reduction in runtime mode, rather than input states, we exploit state probabilities for pin reordering. Similar to the state probability-aware $V_t/T_{\rm ox}$ assignment, with state probability information $I_{\rm gate}$ can be reduced by placing an NMOS transistor with a very high state probability at the bottom of the stack.

In runtime mode, we can also improve the leakage/performance tradeoff with knowledge of state probabilities. Instead of a fixed-input state, we exploit the state probability of a gate and combine it with $V_t/T_{\rm ox}$ assignment. We first determine the probability of a gate being in a particular state using the input state probabilities of the gate, as explained in Section II-A. For instance, if both inputs A and B of NAND2 gate have 0.8 as their state probabilities, this NAND2 gate will have a "11" state with the highest probability of 0.64. Since a NAND2 gate in this state has $I_{\rm sub}$ through the nonconducting PMOS transistors and $I_{\rm gate}$ through the conducting NMOS devices, high-V_t assignment to PMOS transistors and thick-oxide assignment to NMOS transistors influences leakage current more than any other $V_t/T_{\rm ox}$ assignment. This means that if, in probability-aware optimization, we assign high-V_t to t_p1/t_p2 and thick-oxide to t_n1/t_n2 [Fig. 3(c)], we can reduce leakage current more effectively than when we assign high-V_t/thick-oxide to other transistors. On the other hand, if $V_t/T_{\rm ox}$ assignment is performed without knowledge of the state probability, each input state of the gate appears to have equal probability and it is likely that a different group, or possibly the whole gate, is chosen for high-V_t/thick-oxide

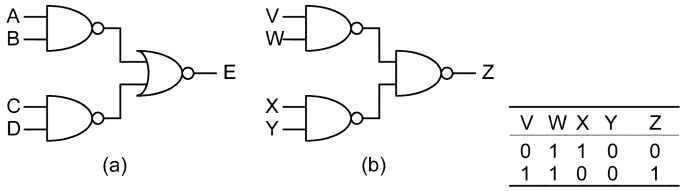


Fig. 5. Functional symmetry.

assignment, resulting in a worsened leakage/delay tradeoff. In the previous example, if high-V_t is assigned to t_p1/t_p2 and thick-oxide to t_n1/t_n2 Fig. 3(c) with the given input state probabilities and using the data shown in Table II, the leakage current of this NAND2 gate is 35.8 nA. If high-V_t is assigned to t_n2 Fig. 3(b) due to a lack of state probability information, the leakage becomes 191.6 nA, or $\sim 5X$ larger in this example. As shown in Table IV, with a given input state probability the leakage current with group $V_t/T_{\rm ox}$ assignment in Fig. 3(c) (35.8 nA) is relatively close to that with $V_t/T_{\rm ox}$ assignment for the lowest leakage current [9 nA in Fig. 3(d)]. However, the case with group $V_t/T_{\rm ox}$ assignment in Fig. 3(b) (191.6 nA) shows leakage that is close to that with the maximum performance $V_t/T_{\rm ox}$ assignment (212.8 nA). This indicates that without consideration of the input state probabilities, $V_t/T_{\rm ox}$ assignment will not improve the leakage/performance tradeoff significantly. Hence, it has been common in traditional probability-unaware optimizations to simply assign the entire gate to high or low V_t. On the other hand, with state probability information a group based $V_t/T_{\rm ox}$ assignment can significantly improve the leakage current and reduce the performance penalty in runtime mode.

B. Probability-Aware Pin Rewiring at Supergates

Two pins A and B of the NAND2 in Fig. 3 can be swapped with no change in the function of the gate. We call these two pins functionally symmetric. We can find functionally symmetric pins across multiple gates as well as within a single gate. For instance, Fig. 5(a) is an example of functionally symmetric pins. In Fig. 5(a), all pins A, B, C, and D are functionally symmetric, therefore, all these pins can be swapped (rewired) with no change in the function of output pin E. However, Fig. 5(b) is not fully symmetric. If pins V, W, X, and Y are rewired, the function of output pin Z may be changed. For example, if input pins V, W, X, and Y have "$0, 1, 1,$ and 0" as their input values, respectively, output pin Z is low. On the other hand, if these pins have "$1, 1, 0,$ and 0" after pin rewiring, the value of output pin Z becomes high. As in Fig. 5(a), when a number of gates have functionally symmetric input pins and there is no functional change in the output pin, we call this group of gates a supergate

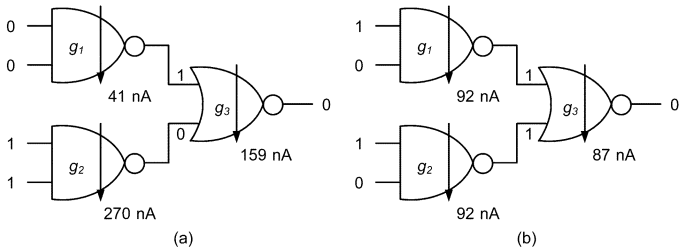


Fig. 6. Pin rewiring.

[22]. In order to find symmetric pins (supergates) in a circuit, we use a linear-time algorithm for symmetry identification in a multilevel netlist [14].

$I_{\rm gate}$ can be reduced by pin reordering as discussed in Section III-A. However, when NAND2 gates have the same input values such as "00" or "11," as shown in Fig. 6(a), pin reordering cannot be applied. On the other hand, if we introduce pin rewiring at supergates, we can obtain additional leakage current reduction. In Fig. 6(a), gate g_1 has 41 nA as its leakage current mainly due to $I_{\rm sub}$, while gate g_2 has 270 nA mainly due to $I_{\rm gate}$. Because these two gates are within the same supergate, i.e., the four pins of two NAND2 gates are functional symmetric, we can rewire these pins to obtain a leakage current reduction. For example, if both NAND2 gates have "10" input values then by rewiring as shown in Fig. 6(b) both NAND2 gates will have little $I_{\rm gate}$ and, hence, their leakage is 92 nA each for a total leakage reduction of $\sim 40\%$ using rewiring. In this example, the leakage current of the NOR2 gate is also reduced from 159 to 87 nA. We seek to apply pin rewiring for runtime leakage reduction by exploiting state probability information. It is noted that since both pin reordering and rewiring impact the delay, they will be applied considering delay constraints in our leakage optimization approach.

C. Circuit Modification by Mapping and Decomposition

Further optimization for leakage current reduction can be obtained using circuit modification by mapping and decomposition. Circuit modification may be applied by changing gate types in order to obtain an increased stack effect, where multiple OFF transistors in series result in significantly reduced $I_{\rm sub}$. For instance, the circuit in Fig. 7(a) can be mapped into a NAND3 gate in Fig. 7(b). The NAND3 gate in Fig. 7(b) may exhibit less leakage current than the circuit in Fig. 7(a) due to the presence of a taller stack, depending on the input state. Leakage reduction will be obtained if the NAND3 stack has a high probability of having multiple inputs with a zero state. In Fig. 7, with given input states, the stack of gate g_1 in Fig. 7(a) has only one OFF transistor, and, consequently, the NAND2 gate g_1 has a leakage current of 92 nA mainly due to $I_{\rm sub}$. On the other hand, since gate g_4 in Fig. 7(b) has two OFF transistors in its stack with the given input states, the NAND3 gate g_4 has only 45 nA as its leakage current, which is less than half that of the NAND2 gate g_1. Moreover, because Fig. 7(a) has two more gates, the total leakage current of Fig. 7(a) is 259 nA. Therefore, if we resynthesize the circuit from Fig. 7(a) to Fig. 7(b) by mapping, we can obtain an 83% leakage current reduction.

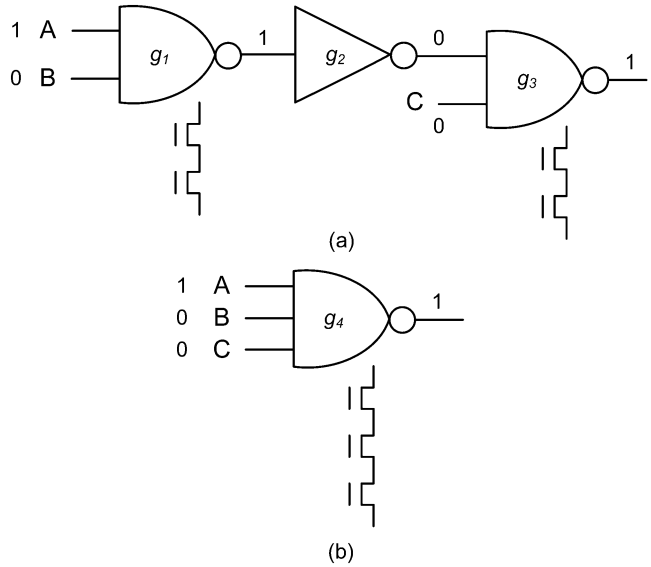


Fig. 7. Circuit modification by mapping.

Again, this circuit modification using mapping or decomposition will alter delay and must be applied while considering the impact on circuit performance. In our leakage current minimization approach, circuit modification by mapping or decomposition will be combined with other leakage reduction techniques under the given delay constraint. Mapping or decomposition will also provide more possibilities for $V_t/T_{\rm ox}$ assignment or pin reordering/rewiring as well as its own leakage current reduction.

IV. Cell Library Construction

In this section, we discuss the construction of needed library cells for dual-V_t/dual-$T_{\rm ox}$ assignment with consideration of state probabilities for runtime leakage reduction. In order to perform leakage current minimization using $V_t/T_{\rm ox}$ assignment with knowledge of state probabilities, it is necessary to construct a library in which all needed $V_t/T_{\rm ox}$ versions for each cell are available. Given such a library, the process of assigning $V_t/T_{\rm ox}$ can be performed by simply swapping cells within the library.

For $V_t/T_{\rm ox}$ assignment, we consider each input state and find the group that is responsible for $I_{\rm sub}$ or $I_{\rm gate}$. For simultaneous $I_{\rm sub}$ and $I_{\rm gate}$ minimization a number of different V_t and $T_{\rm ox}$ assignments are possible that provide different leakage/performance tradeoff points at different input states. A number of $V_t/T_{\rm ox}$ assignments are available for the NOR2 gate, as shown in Fig. 8. For the fastest delay and highest leakage design point, all transistors are assigned to low-V_t and thin-$T_{\rm ox}$, Fig. 8(a). On the other hand, in the slowest delay and lowest leakage point over all possible input states, all transistors are assigned to high-V_t and thick-$T_{\rm ox}$ as shown in Fig. 8(b).

In addition to the fastest and minimum leakage versions of the cell, several intermediate tradeoff points can be constructed for a cell by assigning only some of the transistors (groups) that contribute to leakage to high-V_t or thick-$T_{\rm ox}$. Based on the different library options discussed in more detail in [21], we construct a four-cell version library for each gate. The four $V_t/T_{\rm ox}$

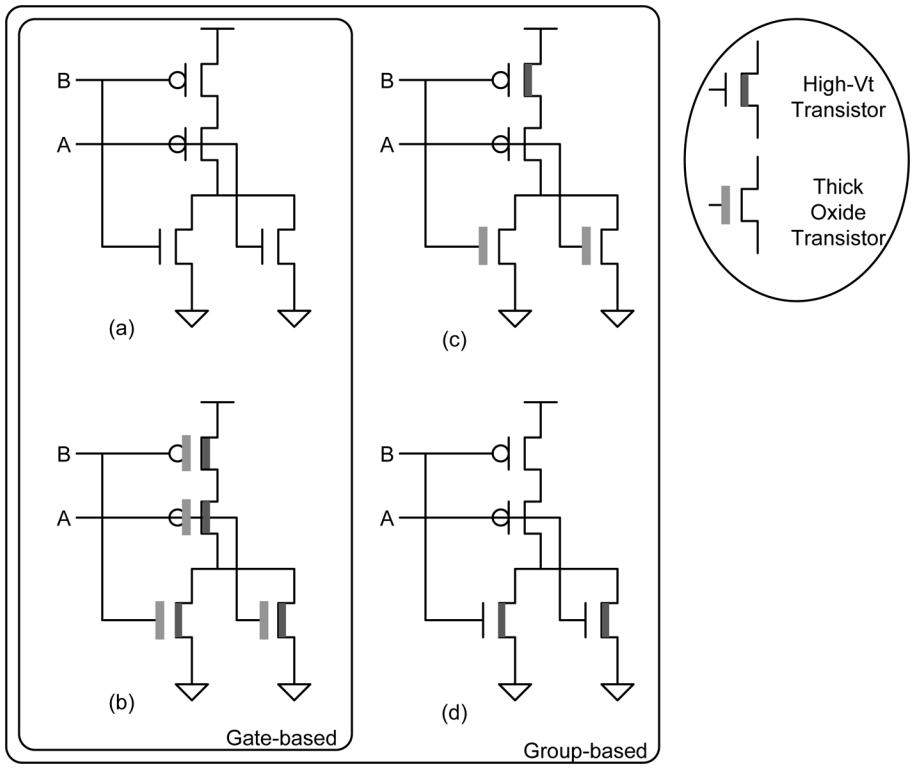


Fig. 8. Four-cell V_t-$T_{\rm ox}$ versions of NOR2 gate.

cell versions shown in Fig. 8(a)–(d) represent the four-cell version library with group-based option for NOR2 gates.

In addition to the above group-based library option, we consider a gate-based library option, where no group $V_t/T_{\rm ox}$ assignment is allowed and gates must consist of either all high-V_t /thick-$T_{\rm ox}$ or all low-V_t/thin-$T_{\rm ox}$ transistors. Therefore, this library option has only two cell versions for each gate. This cell library version is useful in two scenarios: 1) when gate input state probabilities are not highly skewed; 2) for probability-unaware optimization. For a NOR2 gate, the two cells in Fig. 8(a) and 8(b) constitute the gate-based library option. Note that the gate-based library is a subset of the group-based option.

V. Optimization Approach

In this section, we describe the complete state probability-aware leakage optimization method for runtime leakage minimization. Leakage current is optimized using the four different techniques discussed in Section III: 1) $V_t/T_{\rm ox}$ assignment; 2) rewiring functionally symmetric pins; 3) circuit modification by mapping or decomposition; and 4) circuit sizing. We include circuit sizing to supplement the other techniques since it is a well-understood and standard technique for power/delay optimization. Pin reordering is combined with $V_t/T_{\rm ox}$ assignment. All possible $V_t/T_{\rm ox}$ assignments with and without pin reordering are considered for delay/leakage optimization. The objective of the optimization approach is to achieve the minimum leakage current at a specific delay criterion. Starting from the slowest delay and lowest leakage design point, the optimization improves the circuit delay using the four different leakage optimization techniques. The slowest delay and lowest leakage point is obtained using all leakage optimization techniques: high-V_t and thick-$T_{\rm ox}$ assignment at minimum size and circuit resynthesis by pin reordering, pin rewiring, mapping, and decomposition. From this initial point, the optimization moves to the fastest delay and highest leakage point. Each optimization step employs only one technique from among the previous four. At every optimization step, we evaluate the improvements of all possible moves by the four individual techniques and make the single move providing the largest improvement (i.e., maximum sensitivity value). Since the direction of optimization is improving delay, the optimization method uses up-sizing and low-V_t/thin-$T_{\rm ox}$ assignment. For circuit resynthesis, the approach takes the move with the maximum sensitivity value among all possible moves for each technique; pin reordering, pin rewiring, mapping, and decomposition. For example, all possible pin rewiring cases should be considered in order to find the best delay/leakage tradeoff. However, in practice, this enumeration is impractical since a supergate with n symmetric pins would require $n!$ cases, resulting in a very large number of cases for $n > 10$. In our approach, after we divide a supergate into subunits whose number of symmetric pins is under 10, we perform pin rewiring within those sub-units. This heuristic approach is found to maintain the performance of pin rewiring, since most supergates having a large number of symmetric pins consist of repeated patterns and can be easily subdivided. For the mapping and decomposition, we exhaustively explore all possible mapping and decomposition possibilities. However, if we use a high-quality existing mapping and decomposition algorithm, the runtime of our optimization process will improve.

The performance of a circuit has to be evaluated after every optimization move. We use static timing analysis to do this

1. Set the circuit at the initial point - the slowest delay and lowest leakage point using all leakage optimization techniques: high- V_t and thick- T_{ox} assignment at minimum size and circuit resynthesis by pin reordering, pin rewiring, mapping and decomposition.
2. While more improvement of delay can be obtained:
   2.1. Calculate the sensitivity values (improved delay divided by increased leakage current) of all possible optimization techniques for each gate:
      -. V_t / T_{ox} assignment
      -. Rewiring functionally symmetric pins
      -. Circuit modification by mapping or decomposition
      -. Circuit sizing
   2.2. Select the move with the maximum sensitivity value (more improved delay with less increased leakage current) among all moves by 2.1.
   2.3. Update the circuit with the move which is selected at 2.2.
   2.4. Update the delay and leakage value of the circuit.

Fig. 9. Algorithm for the state probability-aware leakage optimization method for runtime leakage minimization.

TABLE V
COMPARISON OF LEAKAGE AND DELAY BETWEEN FOUR POSSIBLE $V_T - T_{\rm OX}$ ASSIGNMENT FOR NMOS

Assignment		Normalized values			
		Leakage			Delay
V_t	Oxide thickness	I_{sub}	Forward I_{gate}	Reverse I_{gate}	
Low	Thin	1.00	0.41	0.22	1.00
High	Thin	0.06	0.31	0.22	1.33
Low	Thick	0.73	0.04	0	1.26
High	Thick	0.05	0.03	0.02	1.69

where delay is calculated using a delay table with input transition time and output capacitance load indices for each gate. For fast delay evaluation, we use local delay calculation. For the every possible optimization move, we update only the local delay of a gate being considered for an optimization move. In the present implementation, the performance of the entire circuit is updated after the best move is selected among all possible moves. Runtime can be further improved by updating the delay of only those paths (gates) that are related to the taken move, rather than the entire circuit. The leakage current values used in calculating the sensitivity values are based on knowledge of the state probability as described earlier. The algorithm for our state probability-aware leakage optimization method for runtime leakage minimization is shown in Fig. 9.

During $V_t/T_{\rm ox}$ assignment with group-based library options multiple moves are possible for each gate, while in the gate-based library only one move needs to be considered per gate. With a gate-based library, making a $V_t/T_{\rm ox}$ assignment is a fairly coarse-grained move compared to the group-based library which provides intermediate steps in the leakage/delay space. From the all high-V_t and thick-$T_{\rm ox}$ version of a gate, high-V_t or thick-$T_{\rm ox}$ is first assigned to some of the groups in a gate. After some groups are selected for high-V_t or thick-$T_{\rm ox}$ assignment, the next move considers setting all transistors to low-V_t and thin-$T_{\rm ox}$. In the example of a NOR2 gate with gate-based and group-based libraries shown in Fig. 8, the optimization with gate-based library option moves only from Fig. 8(b) to Fig. 8(a). However, if the optimization uses the group-based library, the first step of the optimization is a move from Fig. 8(b) to either Fig. 8(c) or Fig. 8(d), and the second move would be from Fig. 8(c) or Fig. 8(d) to Fig. 8(a).

VI. LEAKAGE MODEL AND CHARACTERISTICS

Since the proposed leakage minimization approach is a library-based method, precharacterized leakage current tables for each library cell are used. Each table has specific leakage current values for each possible input state of a library cell. Based on the current values for each input state, the leakage current value in runtime mode is calculated. The precharacterized tables for $I_{\rm sub}$ and $I_{\rm gate}$ minimization were constructed based on SPICE simulations with BSIM4 models using a predictive 65-nm process with a gate leakage component that is approximately 36% of the total leakage at room temperature. For performance characterization, precharacterized delay and output slope tables were stored as a function of cell input slope and output loading. The difference in $I_{\rm gate}$ for the thick-$T_{\rm ox}$ NMOS transistor versus thin-$T_{\rm ox}$ transistor is 11X whereas $I_{\rm sub}$ is reduced by 17.8X (16.7X) when replacing a low-V_t NMOS (PMOS) transistor with a high-V_t version. The high-to-low 50% delay difference for an all low-V_t/thin-$T_{\rm ox}$ inverter versus high-V_t/thick-$T_{\rm ox}$ inverter is 70%. Table V shows relative leakage and delay values at the four possible V_t and $T_{\rm ox}$ assignments for NMOS devices in this technology. Note that in addition to $I_{\rm gate}, I_{\rm sub}$ is also reduced for the low-V_t/thick-$T_{\rm ox}$ device. This is the result of a slight V_t rises as $T_{\rm ox}$ is thickened. In addition, we also increase transistor gate length slightly in thick oxide devices to maintain good short channel effects, which further increases V_t in these devices. The dependence between V_t and $T_{\rm ox}$ is automatically captured in our SPICE-based characterization process.

VII. RESULT

The proposed probability-aware leakage minimization method using simultaneous $V_t/T_{\rm ox}$ assignment, circuit sizing,

TABLE VI
LEAKAGE CURRENT COMPARISON BETWEEN PROBABILITY-UNAWARE AND -AWARE MINIMIZATION BY $V_T/T_{\rm OX}$ ASSIGNMENT AND SIZING METHOD. ALL METHODS USE THE GROUP-BASED LIBRARY AND $P = 0.1/0.9$

	Number of gates	10% delay penalty			20% delay penalty			30% delay penalty		
		$I_{\rm leak}$ [μA]		Diff. %	$I_{\rm leak}$ [μA]		Diff. %	$I_{\rm leak}$ [μA]		Diff. %
		Unaware	Aware		Unaware	Aware		Unaware	Aware	
c432	381	112.82	76.64	32.07	77.08	45.58	40.86	55.11	30.70	44.30
c499	948	392.14	277.11	29.33	235.00	170.05	27.64	155.85	114.54	26.51
c880	648	164.70	122.10	25.87	124.84	74.51	40.31	88.92	50.12	43.64
c1355	864	273.64	209.55	23.42	190.25	149.76	21.28	120.50	103.17	14.38
c1908	783	214.18	164.93	22.99	137.97	103.14	25.25	95.12	67.04	29.52
c2670	1147	217.15	138.25	36.34	148.96	84.38	43.35	91.42	60.40	33.93
c3540	1595	391.91	287.46	26.65	279.05	184.07	34.04	183.59	122.85	33.09
c5315	2392	431.95	317.00	26.61	296.63	230.87	22.17	218.60	156.18	28.55
c6288	4241	1287.40	1032.50	19.80	895.22	661.72	26.08	610.56	453.89	25.66
c7552	2828	567.68	450.52	20.64	381.20	284.51	25.37	273.06	190.20	30.34
alu64	4458	557.88	378.09	32.23	407.02	247.24	39.26	289.71	158.85	45.17
i1	59	17.84	13.10	26.61	11.96	11.28	5.69	8.09	7.76	4.08
i2	208	70.23	55.46	21.03	50.38	32.24	36.00	28.54	21.80	23.61
i3	258	174.97	99.44	43.17	94.58	59.04	37.58	65.49	44.62	31.87
i4	220	87.04	63.66	26.86	67.70	40.39	40.34	43.84	27.32	37.68
i5	536	117.38	75.84	35.39	81.45	48.01	41.05	54.29	32.40	40.32
i6	682	215.18	133.12	38.14	163.04	83.77	48.62	125.87	49.88	60.37
i7	724	242.81	144.75	40.39	143.35	82.52	42.44	106.22	52.70	50.39
i8	1280	395.18	267.38	32.34	275.83	161.80	41.34	195.41	93.37	52.22
i9	768	271.24	164.68	39.29	173.92	94.83	45.48	113.13	67.84	40.04
i10	2853	410.49	294.52	28.25	298.54	186.17	37.64	203.24	125.16	38.42
AVG				29.88			34.37			34.96

and resynthesis by pin reordering, pin rewiring, mapping, and decomposition was implemented on a number of benchmark circuits (10 ISCAS85 circuits [23], 10 MCNC benchmark circuits,[1] and one 64-bit ALU benchmark circuit) synthesized using an industrial cell library. Based on the given state probabilities of the primary inputs, we compute the state probability of each node in the circuit using the method described in [20]. Our proposed state probability-aware method is compared with the state probability-unaware method where all nodes have equal probability of 0.5. First, in order to show the effectiveness of the state probability-aware method, we compare our proposed method with a traditional leakage optimization approach. For the previous approach, we use state probability-unaware $V_t/T_{\rm ox}$ assignment and simultaneous circuit sizing. Leakage current using this previous method is compared with the proposed state probability-aware $V_t/T_{\rm ox}$ assignment and simultaneous circuit sizing method. After that, in order to achieve further leakage reduction, we combine circuit resynthesis with $V_t/T_{\rm ox}$ assignment and circuit sizing. Our proposed approach is tested with a predictive 65-nm technology for both $I_{\rm sub}$ and $I_{\rm gate}$ minimization, as discussed in Section VI.

A comparison between state probability-unaware and -aware leakage optimization methods is shown in Table VI. This optimization method performs $V_t/T_{\rm ox}$ assignment and sizing with a group-based library. The state probabilities of the primary inputs are $P = 0.1/0.9$. At three different delay backoff points (10%, 20%, and 30% larger than the minimum achievable delay) the leakage current values with state probability-unaware and -aware methods are shown. The leakage reduction percentages of the probability-aware method versus probability-unaware method are also shown. Across the three delay penalty points, the probability-aware method shows approximately 30%–35% lower leakage current on average than the probability-unaware method, with a 60% maximum improvement.

In Table VII, we compare the leakage current reduction achieved using different cell library options. The results using the state probability-aware method with gate-based and group-based libraries are compared with that of the state probability-unaware method with a gate-based library option. Table VII shows that with the same gate-based library, the probability-aware method has 7% lower leakage current on average than the probability-unaware method (Column 6). When the probability-aware method uses the group-based library, this leakage reduction improves to 48%. This clearly shows that the probability-aware method benefits significantly from the group-based library option that was specifically tailored for skewed input probabilities.

It is possible that performing individual $V_t/T_{\rm ox}$ assignment of series connected transistors requires these transistors to be spaced out slightly to meet design rules [24]. Hence, using a group-based library may incur a layout area overhead. However, it is also possible to perform a P/N based $V_t/T_{\rm ox}$ assignment where stacks of transistors are given a uniform assignment. In our previous work [25], we have shown that such an assignment approach results in leakage currents that are only slightly higher than the group-based approach. Such a P/N base assignment may, therefore, strike a favorable tradeoff if individual $V_t/T_{\rm ox}$ assignment in series connected transistors requires additional spacing for a particular technology.

Table VIII shows the comparison between different state probabilities of the primary inputs. When the primary inputs show moderate state probabilities of 0.5, the probability-aware optimization performs at its worst but still enables 12% leakage reduction compared to probability-unaware techniques. This

[1][Online]. Available: http://www.cbl.ncsu.edu

TABLE VII
LEAKAGE CURRENT COMPARISON BETWEEN CELL LIBRARY OPTIONS BY V_T/T_{OX} ASSIGNMENT AND SIZING METHOD. ALL METHODS USE 10% DELAY PENALTY POINT AND $P = 0.1/0.9$ (CURRENT IN MICROAMPERES, OPTIMIZATION RUNTIME IN SECONDS)

	Probability-unaware		Probability-aware method (Diff% vs. unaware & gate-based)					
	Gate-based library		Gate-based library			Group-based library		
	I_{leak}	Runtime	I_{leak}	Runtime	Diff. %	I_{leak}	Runtime	Diff. %
c432	180.51	4	173.72	4	3.76	76.64	5	57.54
c499	462.58	35	451.52	32	2.39	277.11	42	40.09
c880	232.58	14	229.20	13	1.45	122.10	16	47.50
c1355	393.98	30	340.25	29	13.64	209.55	36	46.81
c1908	286.17	21	274.50	19	4.08	164.93	24	42.36
c2670	281.84	37	278.60	35	1.15	138.25	42	50.95
c3540	502.02	95	495.82	91	1.23	287.46	111	42.74
c5315	617.65	186	559.94	175	9.34	317.00	208	48.68
c6288	1923.05	1037	1706.46	1007	11.26	1032.50	1200	46.31
c7552	828.55	311	783.25	296	5.47	450.52	356	45.63
alu64	834.66	678	782.46	653	6.25	378.09	729	54.70
i1	19.71	0	17.92	0	9.06	13.10	0	33.55
i2	103.73	1	80.55	1	22.35	55.46	1	46.54
i3	141.43	2	133.06	3	5.92	99.44	3	29.69
i4	126.11	1	111.51	1	11.58	63.66	2	49.52
i5	165.00	8	153.07	8	7.23	75.84	10	54.04
i6	294.37	15	268.26	14	8.87	133.12	18	54.78
i7	347.20	19	303.19	17	12.68	144.75	22	58.31
i8	654.67	64	566.52	85	13.46	267.38	75	59.16
i9	348.13	20	343.95	19	1.20	164.68	24	52.70
i10	545.52	241	520.17	235	4.65	294.52	263	46.01
AVG					7.48			47.98

TABLE VIII
LEAKAGE CURRENT COMPARISON BETWEEN STATE PROBABILITIES OF THE PIS BY V_T/T_{OX} ASSIGNMENT AND SIZING METHOD WITH GROUP-BASED LIBRARY. ALL METHODS USE 10% DELAY PENALTY POINT

	Leakage current difference % between probability unaware and aware optimization		
	P=0.1/0.9	P=0.2/0.8	P=0.5
c432	32.07	21.38	23.15
c499	29.33	22.78	15.26
c880	25.87	19.73	18.21
c1355	23.42	11.04	0.94
c1908	22.99	15.87	4.42
c2670	36.34	26.18	11.69
c3540	26.65	19.69	13.06
c5315	26.61	18.05	7.29
c6288	19.80	14.50	11.18
c7552	20.64	15.71	9.67
alu64	32.23	24.30	10.92
i1	26.61	16.67	7.16
i2	21.03	15.38	3.77
i3	43.17	31.45	0.75
i4	26.86	22.08	15.73
i5	35.39	23.45	8.83
i6	38.14	29.50	22.44
i7	40.39	31.68	21.75
i8	32.34	24.97	17.81
i9	39.29	29.47	21.89
i10	28.25	24.52	14.88
AVG	29.88	21.83	12.42

indicates that the proposed techniques are widely applicable and can be useful even when node state probabilities are not divergent as described earlier.

We now compare the total transistor size (width) of the circuits optimized using both state probability-unaware and -aware methods with a group-based library and PI state probabilities of $P = 0.1/0.9$ in Table IX. The results show that the proposed method results in 3%–8% smaller circuit size than the probability-unaware method on average. Since dynamic power is proportional to total transistor width, the probability-aware optimization method results in lower dynamic power as well as reduced static power compared to traditional techniques.

Table X shows the additional leakage reduction by combining circuit resynthesis with state probability-aware V_t/T_{ox} assignment and sizing using a group-based library. The state probabilities of the PIs are $P = 0.1/0.9$. Table X shows

TABLE IX
TOTAL TRANSISTOR SIZE COMPARISON BETWEEN STATE PROBABILITY-UNAWARE AND -AWARE METHODS BY V_T/T_{OX} ASSIGNMENT AND SIZING WITH GROUP-BASED LIBRARY. ALL METHODS USE $P = 0.1/0.9$

	10% delay penalty			20% delay penalty			30% delay penalty		
	Width [mm]		Diff. %	Width [mm]		Diff. %	Width [mm]		Diff. %
	Unaware	Aware		Unaware	Aware		Unaware	Aware	
c432	6.58	6.33	3.82	6.36	6.06	4.73	6.14	5.54	9.76
c499	18.51	18.43	0.40	17.94	17.61	1.84	17.50	16.71	4.49
c880	11.00	10.65	3.17	10.33	9.85	4.71	9.74	9.14	6.19
c1355	17.71	17.34	2.06	17.19	16.56	3.63	16.52	15.82	4.21
c1908	12.67	12.32	2.76	12.11	11.75	3.02	11.57	10.96	5.27
c2670	16.38	15.87	3.15	15.72	14.48	7.88	14.87	13.25	10.87
c3540	26.43	25.05	5.21	24.98	23.32	6.66	23.20	21.55	7.14
c5315	31.43	30.58	2.69	29.70	28.86	2.84	28.31	27.01	4.60
c6288	69.04	68.38	0.95	67.19	65.96	1.82	65.04	63.36	2.58
c7552	42.52	42.12	0.94	40.30	38.76	3.82	38.23	35.14	8.07
alu64	60.12	53.82	10.48	56.34	49.43	12.26	53.04	44.90	15.33
i1	1.03	0.98	4.33	0.98	0.94	4.52	0.96	0.91	5.56
i2	4.35	4.27	1.96	4.17	4.15	0.46	4.15	4.07	2.10
i3	5.97	5.84	2.17	5.87	5.61	4.51	5.81	5.22	10.28
i4	3.95	3.91	1.09	3.93	3.72	5.18	3.80	3.54	7.01
i5	8.41	8.04	4.40	8.01	7.39	7.75	7.49	6.89	8.08
i6	12.53	11.69	6.73	12.12	10.93	9.83	11.84	9.99	15.60
i7	14.37	13.68	4.77	13.89	12.92	6.97	13.57	11.73	13.53
i8	24.39	23.71	2.79	23.74	22.26	6.24	23.02	20.03	12.98
i9	14.04	13.44	4.23	13.30	12.54	5.72	12.83	11.88	7.40
i10	33.32	30.98	7.04	30.80	28.69	6.86	28.97	26.60	8.17
AVG			3.58			5.30			8.06

TABLE X
LEAKAGE CURRENT COMPARISON BETWEEN V_T/T_{OX} ASSIGNMENT—SIZING METHOD AND COMPLETE METHOD (V_T/T_{OX} ASSIGNMENT—SIZING WITH CIRCUIT RESYNTHESIS). ALL METHODS USE THE STATE PROBABILITY-AWARE METHOD WITH THE GROUP-BASED LIBRARY AND $P = 0.1/0.9$

	10% delay penalty			20% delay penalty			30% delay penalty		
	I_{leak} [μA]		Diff. %	I_{leak} [μA]		Diff. %	I_{leak} [μA]		Diff. %
	V_t/T_{ox} + Sizing	Complete (V_t/T_{ox} + Sizing + resynthesis)		V_t/T_{ox} + Sizing	Complete (V_t/T_{ox} + Sizing + resynthesis)		V_t/T_{ox} + Sizing	Complete (V_t/T_{ox} + Sizing + resynthesis)	
c432	76.64	51.85	32.35	45.58	31.02	31.94	30.70	21.87	28.76
c499	277.11	228.28	17.62	170.05	150.35	11.58	114.54	97.46	14.91
c880	122.10	119.88	1.81	74.51	53.99	27.55	50.12	41.24	17.72
c1355	209.55	197.72	5.65	149.76	141.09	5.79	103.17	96.71	6.26
c1908	164.93	148.20	10.14	103.14	91.37	11.41	67.04	66.27	1.15
c2670	138.25	117.50	15.00	84.38	72.28	14.35	60.40	51.73	14.36
c3540	287.46	244.40	14.98	184.07	142.56	22.55	122.85	102.40	16.64
c5315	317.00	294.25	7.18	230.87	196.64	14.83	156.18	140.25	10.20
c6288	1032.50	862.63	16.45	661.72	555.93	15.99	453.89	385.06	15.16
c7552	450.52	382.87	15.02	284.51	246.84	13.24	190.20	171.80	9.68
alu64	378.09	267.25	29.31	247.24	176.55	28.59	158.85	130.88	17.61
i1	13.10	9.11	30.43	11.28	4.93	56.35	7.76	3.63	53.27
i2	55.46	43.66	21.27	32.24	25.97	19.45	21.80	20.54	5.79
i3	99.44	88.08	11.42	59.04	53.60	9.22	44.62	37.40	16.18
i4	63.66	52.34	17.78	40.39	35.38	12.41	27.32	24.43	10.59
i5	75.84	69.26	8.67	48.01	40.90	14.82	32.40	28.70	11.41
i6	133.12	112.26	15.67	83.77	75.08	10.38	49.88	36.59	26.64
i7	144.75	83.74	42.15	82.52	51.80	37.23	52.70	38.31	27.31
i8	267.38	195.86	26.75	161.80	122.19	24.48	93.37	77.05	17.48
i9	164.68	160.53	2.52	94.83	91.40	3.62	67.84	49.85	26.52
i10	294.52	252.72	14.19	186.17	154.45	17.04	125.16	107.98	13.72
AVG			16.97			19.18			17.21

leakage current values with the complete method (state probability-aware V_t/T_{ox} assignment, circuit sizing, and resynthesis with the group-based library) in Column 3, 6, and 9 at three different delay backoff points. These results are compared with those obtained by the probability-aware V_t/T_{ox} assignment and sizing method with the group-based library (Columns

TABLE XI
LEAKAGE CURRENT COMPARISON BETWEEN THE TRADITIONAL METHOD (STATE PROBABILITY-UNAWARE V_T/T_{OX} ASSIGNMENT—SIZING WITH THE GATE-BASED LIBRARY) AND THE PROPOSED COMPLETE METHOD (STATE PROBABILITY-AWARE V_T/T_{OX} ASSIGNMENT, SIZING, AND CIRCUIT RESYNTHESIS WITH THE GROUP-BASED LIBRARY). ALL METHODS USE 10% DELAY PENALTY POINT AND $P = 0.1/0.9$ (CURRENT IN MICROAMPERES, OPTIMIZATION RUNTIME IN SECONDS)

	Probability-unaware		Probability-aware method (Diff% vs. unaware & gate-based)					
	Gate-based library		Group-based library					
	V_t/T_{ox} + Sizing		V_t/T_{ox} + Sizing			Complete (V_t/T_{ox} + Sizing + resynthesis)		
	I_{leak}	Runtime	I_{leak}	Runtime	Diff. %	I_{leak}	Runtime	Diff. %
c432	180.51	4	76.64	5	57.54	51.85	3645	71.28
c499	462.58	35	277.11	42	40.09	228.28	23454	50.65
c880	232.58	14	122.10	16	47.50	119.88	30534	48.46
c1355	393.98	30	209.55	36	46.81	197.72	16095	49.81
c1908	286.17	21	164.93	24	42.36	148.20	10590	48.21
c2670	281.84	37	138.25	42	50.95	117.50	33353	58.31
c3540	502.02	95	287.46	111	42.74	244.40	62246	51.32
c5315	617.65	186	317.00	208	48.68	294.25	75659	52.36
c6288	1923.05	1037	1032.50	1200	46.31	862.63	393284	55.14
c7552	828.55	311	450.52	356	45.63	382.87	151979	53.79
alu64	834.66	678	378.09	729	54.70	267.25	135709	67.98
i1	19.71	0	13.10	0	33.55	9.11	30	53.77
i2	103.73	1	55.46	1	46.54	43.66	410	57.91
i3	141.43	2	99.44	3	29.69	88.08	280	37.72
i4	126.11	1	63.66	2	49.52	52.34	490	58.50
i5	165.00	8	75.84	10	54.04	69.26	1076	58.02
i6	294.37	15	133.12	18	54.78	112.26	8156	61.87
i7	347.20	19	144.75	22	58.31	83.74	1432	75.88
i8	654.67	64	267.38	75	59.16	195.86	42937	70.08
i9	348.13	20	164.68	24	52.70	160.53	42438	53.89
i10	545.52	241	294.52	263	46.01	252.72	301077	53.67
AVG					47.98			56.60

2, 5, and 8) which are equivalent to Columns 3, 6, and 9 of Table VI. When circuit resynthesis is combined with state probability-aware V_t/T_{ox} assignment and sizing, an average 17%–19% further leakage reduction over the state probability-aware V_t/T_{ox} assignment and sizing method can be achieved.

Results from the complete optimization with a group-based library are now compared with those obtained using the traditional leakage optimization method, i.e., state probability-unaware V_t/T_{ox} assignment and sizing with a gate-based library, in Table XI. Table XI shows that the proposed complete leakage optimization approach obtains 57% leakage reduction on average over the traditional leakage optimization approach. Table XI also shows optimization runtimes for each leakage optimization method. Note that runtime using the complete approach is much larger than those of the other approaches. This results from the additional circuit resynthesis which is not performed in other approaches, and particularly from pin rewiring since it requires a large number of iterations to find the optimum delay/leakage tradeoff.

Finally, Fig. 10 plots the leakage current results for the proposed method as well as the probability-unaware method, with different library options as a function of the delay for circuit c6288. As shown in Fig. 10, the proposed complete method has the lowest leakage current. Among the other four curves using V_t/T_{ox} assignment and sizing approaches, the probability-aware optimization with the group-based library achieves the best result. Since the gate-based libraries consistently show worsened leakage for a given delay (top two curves), we can conclude that the use of group-based libraries is critical to leakage current optimization, along with the use of state probabilities. In Fig. 10, we also see that the group-based library option shows a bigger difference between the probability-aware and -unaware methods than the gate-based library as expected.

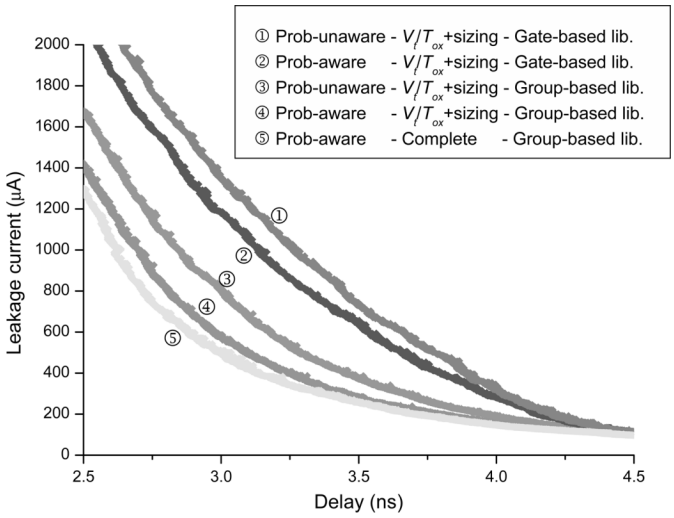


Fig. 10. Leakage current comparison for c6288.

VIII. CONCLUSION

In this paper, we have proposed a new leakage optimization method that specifically targets runtime leakage current. The method uses the skewed gate input state probabilities by setting

only those transistors in a gate to high-V_t and thick-$T_{\rm ox}$ that are most likely to contribute significantly to the total leakage current. The technique uses a sensitivity-based approach where leakage current is computed using the gate input state probabilities. A library where transistor-level V_t and $T_{\rm ox}$ assignments are selected based on expected skewed input probabilities was developed and results in significant leakage reduction for the probability-aware optimization approach. For further leakage reduction, we incorporate circuit resynthesis, consisting of pin reordering, pin rewiring, mapping, or decomposition, to the state probability-aware $V_t/T_{\rm ox}$ assignment and circuit sizing approach. The proposed state probability-aware method improves leakage current by an average of 30% over a state probability-unaware method. The complete proposed method, including circuit resynthesis and the specially tailored cell library, achieves an average of 57% runtime leakage reduction over the traditional state probability-unaware method with a basic standard cell library option.

ACKNOWLEDGMENT

The authors would like to thank Mr. K. Chopra for his help and useful discussions.

REFERENCES

[1] S. Narendra, D. Blaauw, A. Devgan, and F. Najm, "Leakage issues in IC design: Trends, estimation and avoidance," in *Proc. ICCAD (Tutorial)*, 2003, p. xi.

[2] S. Mutoh, T. Douseki, Y. Matsuya, T. Aoki, S. Shigematsu, and J. Yamada, "1-V power supply high-speed digital circuit technology with multithreshold-voltage CMOS," *IEEE J. Solid-State Circuits*, vol. 30, no. 8, pp. 847–854, Aug. 1995.

[3] S. Shigematsu, S. Mutoh, Y. Matsuya, Y. Tanabe, and J. Yamada, "A 1-V high-speed MTCMOS circuit scheme for power-down application circuits," *IEEE J. Solid-State Circuits*, vol. 32, no. 6, pp. 861–869, Jun. 1997.

[4] J. Halter and F. Najm, "A gate-level leakage power reduction method for ultra-low-power CMOS circuits," in *Proc. CICC*, 1997, pp. 475–478.

[5] V. De, Y. Ye, A. Keshavarzi, S. Narendra, J. Kao, D. Somasekhar, R. Nair, and S. Borkar, "Techniques for leakage power reduction," in *Design of High-Performance Microprocessor Circuits*. Piscataway, NJ: IEEE Press, 2001.

[6] M. C. Johnson, D. Somasekhar, and K. Roy, "Models and algorithms for bounds on leakage in CMOS circuits," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 18, no. 6, pp. 714–725, Jun. 1999.

[7] T. Kuroda, T. Fujita, S. Mita, T. Nagamatsu, S. Yoshioka, K. Suzuki, F. Sano, M. Norishima, M. Murota, M. Kako, M. Kinugawa, M. Kakumu, and T. Sakurai, "A 0.9 V, 150-MHz, 10-mW, 4 mm2, 2-D discrete cosine transform core processor with variable threshold-voltage (VT) scheme," *IEEE J. Solid-State Circuits*, vol. 31, no. 11, pp. 1770–1779, Nov. 1996.

[8] S. Narendra, D. Antoniadis, and V. De, "Impact of using adaptive body bias to compensate die-to-die Vt variation on within-die Vt variation," in *Proc. Int. Symp. Low Power Electron. Des.*, 1999, pp. 229–232.

[9] L. Wei, Z. Chen, M. C. Johnson, K. Roy, and V. De, "Design and optimization of low voltage high performance dual threshold CMOS circuits," in *Proc. DAC*, 1998, pp. 489–494.

[10] S. Sirichotiyakul, T. Edwards, C. Oh, R. Panda, and D. Blaauw, "Duet: An accurate leakage estimation and optimization tool for dual V_t circuits," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 10, no. 2, pp. 79–90, Apr. 2002.

[11] M. Ketkar and S. Sapatnekar, "Standby power optimization via transistor sizing and dual threshold voltage assignment," in *Proc. ICCAD*, 2002, pp. 375–378.

[12] D. Lee and D. Blaauw, "Static leakage reduction through simultaneous threshold voltage and state assignment," in *Proc. DAC*, 2003, pp. 191–194.

[13] D. Lee, W. Kwong, D. Blaauw, and D. Sylvester, "Analysis and minimization techniques for total leakage considering gate oxide leakage," in *Proc. Des. Autom. Conf.*, 2003, pp. 175–180.

[14] C. Chang, M. Hsiao, B. Hu, K. Wang, M. Marek-Sadowska, C. Cheng, and S. Chen, "Fast postplacement optimization using functional symmetries," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 23, no. 1, pp. 102–118, Jan. 2004.

[15] G. Sery, S. Borkar, and V. De, "Life is CMOS: Why chase the life after?," in *Proc. DAC*, 2002, pp. 78–83.

[16] G. Hachtel and F. Somenzi, *Logic Synthesis and Verification Algorithms*. Norwell, MA: Kluwer, 2000.

[17] R. X. Gu and M. I. Elmasry, "Power dissipation analysis and optimization of deep submicron CMOS digital circuits," *IEEE J. Solid-State Circuits*, vol. 31, no. 5, pp. 707–713, May 1996.

[18] Z. Chen, M. C. Johnson, L. Wei, and K. Roy, "Estimation of standby leakage power in CMOS circuit considering accurate modeling of transistor stacks," in *Proc. Int. Symp. Low Power Electron. Des.*, 1998, pp. 239–244.

[19] T. Fukai, "A 65 nm-node CMOS technology with highly reliable triple gate oxide suitable for power-constrained system-on-a-chip," in *Proc. IEEE Symp. VLSI Technol.*, 2003, pp. 83–84.

[20] S. Ercolani, M. Favalli, M. Damiani, P. Olivo, and B. Ricco, "Estimate of signal probability in combinational logic networks," in *Proc. Eur. Test Conf.*, 1989, pp. 132–138.

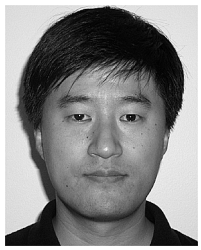
[21] D. Lee, H. Deogun, D. Blaauw, and D. Sylvester, "Simultaneous state, Vt and Tox assignment for total standby power minimization," in *Proc. Des., Autom. Test Eur. Conf. Exhibition*, 2004, pp. 494–499.

[22] K. Tsai, R. Tompson, J. Rajski, and M. Marek-Sadowska, "STAR-ATPG: A high speed test pattern generator for large scan designs," in *Proc. Int. Test Conf.*, 1999, pp. 1021–1030.

[23] F. Brglez and H. Fujiwara, "A neutral netlist of 10 combinatorial benchmark circuits," in *Proc. Int. Symp. Circuit Syst.*, 1985, pp. 695–698.

[24] R. Puri, personal communication, Oct. 2002.

[25] D. Lee, H. Deogun, D. Blaauw, and D. Sylvester, "Runtime leakage minimization through probability-aware dual-V_t or dual-$T_{\rm ox}$ assignment," in *Proc. ASP-DAC*, 2005, pp. 399–404.



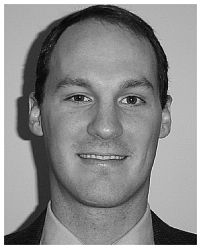
Dongwoo Lee (S'03) received the B.S. and M.S. degrees in electronics engineering from Korea University, Seoul, Korea, in 1994 and 1996, and the Ph.D. degree in electrical engineering from the University of Michigan, Ann Arbor, in 2005.

From May 1996 through June 2001, and since September 2005, he has been with the Flash Memory Design Team, Samsung Electronics Company, Ltd., Gyeonggi-Do, Korea. His research interests include circuit analysis and optimization problems for low-power VLSI systems.

David Blaauw (M'93) received the B.S. degree in physics and computer science from Duke University, Durham, NC, in 1986, and the M.S. and Ph.D. degrees in computer science from the University of Illinois, Urbana, in 1988 and 1991, respectively.

He was a Development Staff Member at the Engineering Accelerator Technology Division, IBM Corporation, Endicott, NY, until August 1993. From 1993 till August 2001, he was with Motorola, Inc., Austin, TX, where he was the Manager of the High Performance Design Technology Group. Since August 2001, he has been an Associate Professor at the University of Michigan, Ann Arbor. His work has focused on VLSI design and CAD with particular emphasis on circuit analysis and optimization problems for high-performance and low-power designs.

Dr. Blaauw was the Technical Program Chair and General Chair for the International Symposium on Low Power Electronic and Design in 1999 and 2000, respectively, and was the Technical Program Co-Chair and member of the Executive Committee for the ACM/IEEE Design Automation Conference in 2000 and 2001.



Dennis Sylvester (S'95–M'00) received the B.S. degree (*summa cum laude*) from the University of Michigan, Ann Arbor, in 1995, and the M.S. and Ph.D. degrees from the University of California, Berkeley, in 1997 and 1999, respectively, all in electrical engineering.

He was with Hewlett-Packard Laboratories, Palo Alto, CA, from 1996 to 1998. After working as a Senior Research and Development Engineer in the Advanced Technology Group of Synopsys, Mountain View, CA. He is currently an Assistant Professor of Electrical Engineering at the University of Michigan, Ann Arbor. He has published numerous papers in his field of research, which includes the modeling, characterization, and analysis of on-chip interconnect, low-power circuit design techniques, and variability-aware circuit approaches.

Dr. Sylvester received a National Science Foundation CAREER Award, the 2000 Beatrice Winner Award at ISSCC, two outstanding Research Presentation Awards from the Semiconductor Research Corporation, and a Best Student Paper Award at the 1997 International Semiconductor Device Research Symposium. He is also a recipient of the 2003 Ruth and Joel Spira Outstanding Teaching Award from the University of Michigan College of Engineering. His dissertation research was recognized with the 2000 David J. Sakrison Memorial Prize as the most outstanding research in the Electrical Engineering and Computer Science Department of the University of California, Berkeley. He is on the technical program committee of several design automation and circuit design conferences and was the general chair for the 2003 ACM/IEEE System-Level Interconnect Prediction (SLIP) Workshop. In addition, he is part of the International Technology Roadmap for Semiconductors (ITRS) U.S. Design Technology Working Group and made significant modeling contributions to the Design and System Drivers chapters of the 2001 ITRS. He is a Member of the Association for Computing Machinery, American Society of Engineering Education, and Eta Kappa Nu.