Static Leakage Reduction Through Simultaneous V_t/T_{ox} and State Assignment

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

***Abstract*—Standby leakage current minimization is a pressing concern for mobile applications that rely on standby modes to extend battery life. In this paper, we propose new leakage current reduction methods in standby mode. First, we propose a combined approach of sleep-state assignment and threshold voltage (V_t) assignment in a dual-V_t process for subthreshold leakage (I_{sub}) reduction. Second, for the minimization of gate oxide leakage current (I_{gate}) which has become comparable to I_{sub} in 90-nm technologies, we extend the above method to a combined sleep-state, V_t and gate oxide thickness (T_{ox}) assignments approach in a dual-V_t and dual-T_{ox} process to minimize both I_{sub} and I_{gate}. By combining V_t or V_t/T_{ox} assignment with sleep-state assignment, leakage current can be dramatically reduced since the circuit is in a known state in standby mode and only certain transistors are responsible for leakage current and need to be considered for high-V_t or thick-T_{ox} assignment. A significant improvement in the leakage/performance tradeoff is therefore achievable using such combined methods. We formulate the optimization problem for simultaneous state/V_t and state/V_t/T_{ox} assignments under delay constraints and propose both an exact method for its optimal solution as well as two practical heuristics with reasonable run time. We implemented and tested the proposed methods on a set of synthesized benchmark circuits and show substantial leakage current reduction compared to the previous approaches using only state assignment or V_t assignment alone.**

***Index Terms*—Dual oxide thickness, dual threshold voltage, gate leakage, leakage currents, power optimization, state assignment, subthreshold leakage.**

I. Introduction

THERE is a growing need for high-performance and low-power systems, especially for portable and battery-powered applications. Since these applications often remain in stand-by mode significantly longer than in active mode, their stand-by (or leakage) current has a dominant impact on battery life. Standby mode leakage current reduction therefore has been a concern for some time and a number of such methods have been proposed to address this problem [1]–[7], [9]–[18]. However, with continued process scaling, lower supply voltages necessitate reduction of threshold voltages to meet performance goals and result in a dramatic increase in subthreshold leakage current. New methods for reducing the leakage current in standby mode are therefore critically needed.

Manuscript received March 17, 2004; revised June 16, 2004. This work was supported by the National Science Foundation, the Semiconductor Research Corporation (SRC), the Gigascale Systems Research Center/Defense Advanced Research Projects Agency (GSRC/DARPA), IBM, and Intel. This paper was recommended by Associate Editor F. N. Najm.

The authors are with the University of Michigan, Ann Arbor, MI 48109 USA (e-mail: dongwool@umich.edu; blaauw@umich.edu; dmcs@umich.edu).

Digital Object Identifier 10.1109/TCAD.2005.847906

In dual-V_t technology, the MTCMOS approach [1] was proposed where a high-V_t sleep transistor is inserted between the power supply and the circuit logic. In standby mode, this sleep transistor is turned off which dramatically reduces leakage due to its high-V_t. However, the method requires routing of an additional set of power supply lines in the layout as well as substantially sized sleep transistors to maintain good supply integrity and circuit performance [2]. Also, special latches that maintain state in standby mode need to be used [3]. In addition, the method does not scale well into sub-1 V technologies due to the increased delay penalty for the high-V_t sleep device [4].

A different approach to standby mode leakage reduction has been proposed that leverages the state dependence of a leakage current due to the so-called stack effect [5], [6]. In [7], the circuit input state that minimizes leakage current is determined and special flip-flops are inserted in the design to produce this state in standby mode. The flip-flops in the design are modified to produce a predetermined state in standby mode while also maintaining the previously latched state. The required modification to a flip-flop is minor and can be incorporated in the feedback path of the slave latch with minimal impact on performance [8]. The implementation of special flip-flops in [8] incurs only a small area overhead due to the change from an inverter to a NAND/NOR gate. Furthermore, the additional leakage current by this modification is also small. In general, determining the minimum sleep state is a difficult problem due to the inherent logic correlations in the circuit. However, a number of efficient heuristics for this problem have been proposed [9], [10]. The limitation of this approach is that for larger circuits, the reduction in leakage current is typically only in the range of 10% to 30% [9].

The above techniques are aimed primarily at subthreshold leakage current reduction which has been the dominant component of leakage in CMOS technologies to date. However, in 90-nm technologies the magnitude of gate tunneling leakage, I_{gate}, in a device is comparable to the subthreshold leakage, I_{sub}, at room temperature. With difficulties in achieving manufacturable high-k insulator solutions to address the gate leakage problem, the burden of addressing this problem is primarily on circuit designers and EDA tools. As a result, there has been recent work in the area of gate leakage analysis and reduction techniques including pin reordering, pMOS sleep transistors, and the use of NAND implementations rather than NOR implementations [11]–[13]. Also, the MTCMOS technique was extended to combat gate leakage by using a thick-oxide I/O device with a larger gate drive than the logic transistors as the inserted sleep transistor [14].

0278-0070/$20.00 © 2005 IEEE

Another previous approach to leakage reduction that targets only subthreshold leakage is to use individual assignment of transistor threshold voltages in a dual-V_t process [15]–[18]. In these approaches, the tradeoff between high-V_t transistors with low leakage/low performance and low-V_t transistors with high leakage/high performance is exploited. Circuit paths that are noncritical are assigned high-V_t while critical circuit portions are given low-V_t assignments. The method therefore provides a tradeoff between circuit performance and leakage reduction. It was demonstrated that with a modest performance reduction of 5–10%, significant reduction of 3–4 × in leakage could be obtained over a circuit with all low-V_t transistors [17]. In these approaches, high/low-V_t assignments are performed without knowledge of the states of the circuit. Therefore, in order to obtain sufficient leakage reduction under all possible circuit states, all or most of the transistors in a particular gate must be set to high-V_t and hence the gate incurs substantial performance degradation.

While such dual-V_t processes have been commonplace for several generations, the availability of multiple oxide thicknesses in a single process has only become relevant at the 90-nm node due to the rise of $I_{\rm gate}$ [19]. Given a process technology with dual oxide thicknesses for logic devices, the dual-V_t approach can be easily extended to also consider gate leakage by assigning thick-oxide transistors to noncritical paths as well. However, similar to the dual-V_t assignment approach, a simultaneous dual-V_t and dual oxide thickness assignment with unknown states of the circuit will set all or most of the transistors in a particular gate to both high-V_t and thick-oxide, to ensure that under all possible circuit states in standby mode leakage current is acceptable. However, transistors that are simultaneously assigned a high-V_t and a thick-oxide have a dramatic delay penalty compared to low-V_t transistors with thin-oxide. Therefore, this approach carries with it a significant delay penalty for process technologies where both $I_{\rm sub}$ and $I_{\rm gate}$ need to be addressed.

In this paper, we therefore propose new methods to reduce standby mode leakage current. We can divide our new methods into two categories: 1) simultaneous dual-V_t and sleep state assignment for $I_{\rm sub}$ reduction for technologies in which $I_{\rm sub}$ is dominant in standby mode and 2) simultaneous dual V_t, dual oxide thickness and sleep state assignment for both $I_{\rm sub}$ and $I_{\rm gate}$ minimization for technologies which have comparable amount of $I_{\rm gate}$ to $I_{\rm sub}$. First, we combine the concepts of V_t assignment and sleep state assignment. This approach is based on the key observation that, given a known input state for a gate, the leakage of that gate can be dramatically reduced by setting only a single OFF-transistor on each path from V_{dd} to Gnd to high-V_t. Since all other transistors in the gate are kept at low-V_t and continue to have high drive current, the performance degradation is limited while a significant gain in leakage current is obtained. Therefore, this approach provides a much better tradeoff between leakage and performance compared to V_t assignment with unknown input state where most or all of the transistors must be set to high-V_t before a significant improvement in the leakage current is observed. The link between the effectiveness of V_t assignment and state assignment was previously observed for Domino logic [8], since these circuits are by their own nature in a known state in standby mode. However, we extend this concept to general CMOS circuits by actively controlling the circuit state in standby mode, thereby dramatically increasing the effectiveness of leakage reduction.

The second proposed approach minimizes the total leakage current ($I_{\rm sub}$ and $I_{\rm gate}$) by simultaneous assignment of sleep state, high-V_t and thick-oxide transistors. In this approach, a key observation is that given a known input state, a transistor need not be assigned *both* a high-V_t and a thick-oxide since $I_{\rm sub}$ only occurs in transistors that are OFF while significant $I_{\rm gate}$ occurs only in transistors that are ON. Furthermore, depending on the input state of a circuit, only a subset of transistors needs to be considered for either high-V_t or thick-oxisde. Therefore, the impact on the delay of the gate is significantly reduced while obtaining leakage reductions comparable to when all transistors are assigned to both high-V_t and thick-oxide. The proposed method is compatible with existing library-based design flows, and we explore different tradeoffs between the number of V_t and $T_{\rm ox}$ variations for each library cell and the obtained leakage reduction. Note that only a few possible V_t and $T_{\rm ox}$ assignments are useful, depending on the input state. Therefore, we can reduce the number of needed library cells significantly. In addition, we compare the obtained leakage reduction when V_t (the first method) and V_t/$T_{\rm ox}$ (the second method) assignments can be made individually for transistors in a stack as opposed to when an entire stack is restricted to a uniform assignment due to manufacturing or area considerations.

Since the circuit state/V_t and the circuit state/V_t/$T_{\rm ox}$ assignments interact, it is necessary to consider their optimization simultaneously. The state/V_t and state/V_t/$T_{\rm ox}$ assignment task is to find a simultaneous assignment that minimizes the leakage current in standby mode while meeting a user specified delay constraint. We formulate this problem as an integer optimization problem under delay constraints. The search space consists of all input states/V_t and input states/V_t/$T_{\rm ox}$ assignments and hence is very large. Therefore, in addition to an exact solution, we also propose a number of heuristics. The proposed methods are implemented on benchmark circuits synthesized using an industrial cell library in 0.18 μm technology for $I_{\rm sub}$ minimization and in a predictive 65-nm technology for both $I_{\rm sub}$ and $I_{\rm gate}$ minimization. On average, the proposed $I_{\rm sub}$ minimization method by simultaneous state/V_t assignment approach improves leakage current by a factor of 5 × over the traditional approach using V_t assignment only. The second proposed method that minimizes both $I_{\rm sub}$ and $I_{\rm gate}$ by simultaneous state/V_t/$T_{\rm ox}$ assignment has an average leakage reduction of 5–6 × over an all low-V_t and thin-oxide design solution at a 5% delay range point and achieves more than a 2 × improvement over the first proposed approach using V_t and state assignment only (i.e., without dual-$T_{\rm ox}$).

The remainder of this paper is organized as follows. In Section II, we discuss the leakage model used and the characteristics of $I_{\rm sub}$ and $I_{\rm gate}$ leakage current. In Section III, we present the approach using simultaneous V_t and state assignment for $I_{\rm sub}$ leakage reduction. In Section IV, we present the second approach that also addresses $I_{\rm gate}$ by performing simultaneous V_t, $T_{\rm ox}$, and state assignment. In Section V, we present our results on benchmark circuits and in Section VI we present our conclusions.

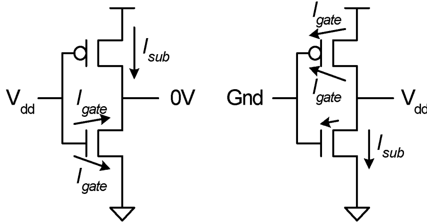

Fig. 1. Inverter circuit with nMOS oxide leakage current.

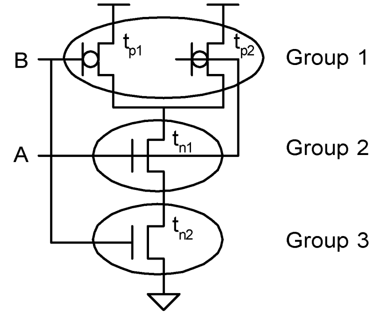

Fig. 2. Concept of groups for a NAND2 gate.

II. Leakage Model and Characteristics

In this section, we discuss our leakage current model and briefly review the general characteristics of gate leakage current in CMOS gates.

Since the proposed leakage optimization approach is library-based, we use precharacterized leakage current tables for each library cell, with specific leakage table entries for each possible input state of a library cell. The precharacterized tables were constructed using SPICE simulation with BSIM3 models from 0.18 μm technology for $I_{\rm sub}$ minimization approach. In order to represent both $I_{\rm sub}$ and $I_{\rm gate}$ components for the state/V_t/$T_{\rm ox}$ assignment approach, BSIM4 models were used to generate the precharacterization of tables. The device simulation parameters were obtained using leakage estimates from a predicted 65-nm processes [20], and had a gate leakage component that was approximately 36% of the total leakage at room temperature (at which all analysis is performed). Also, the delay and output slope as a function of cell input slope and output loading were stored in precharacterized tables.

The total gate leakage for a library cell consists of several different components, depending on the input state of the gate, as illustrated for the inverter cell in Fig. 1. The maximum gate tunneling current occurs when the input is at V_{dd} and $V_s = V_d = 0$ V for the nMOS device. In this case, $V_{gs} = V_{gd} = V_{dd}$ and the $I_{\rm gate}$ is at its maximum for the nMOS device. At the same time, the pMOS device exhibits substantial subthreshold leakage current. When the input is at Gnd, the output rises to V_{dd} and $V_{gs} = 0$ while V_{gd} will become $-V_{dd}$ for the nMOS device, resulting in a reverse gate tunneling current from the drain to the gate node. In this case, tunneling is restricted to the gate-to-drain overlap region, due to the absence of a channel. Since this overlap region is much smaller than the channel region, reverse tunneling current is significantly reduced compared to the forward tunneling current [21]. Note that BSIM4 intrinsically considers this reverse tunneling current so it is included in the precharacterized tables described above.

When the input voltage is Gnd, the pMOS device also exhibits gate current from the channel to the gate since its $V_{gs} = V_{gd} = -V_{dd}$. The relative magnitude of the pMOS gate current in comparison to the nMOS gate current differs for different process technologies. If standard SiO_2 is used as the gate oxide material, then the $I_{\rm gate}$ for a pMOS device is typically one order of magnitude smaller than that for an nMOS device with identical $T_{\rm ox}$ and V_{dd} [19], [22]. This is due to the much higher energy required for hole tunneling in SiO_2 compared to electron tunneling. However, in alternate dielectric materials, the energy required for electron and hole tunneling can be completely different. In the case of nitrided gate oxides, which are in use today in some processes, pMOS $I_{\rm gate}$ can actually exceed nMOS $I_{\rm gate}$ for higher nitrogen concentrations [23], [24]. In this paper, we assume that standard SiO_2 gate oxide material is used and the pMOS gate current is negligible. However, the presented methods can be easily extended to include appreciable pMOS gate leakage as well.

III. Subthreshold Leakage Reduction

A. Simultaneous V_t and State Assignment

Consider the leakage and performance of the simple NAND2 circuit shown in Fig. 2 under different input states and V_t assignments. It is clear that given a particular input state, only those transistors that are OFF need to be considered for high-V_t assignment as the ON-transistors are not leaking. For instance, in state $\rm AB = 01$, only transistor t_{n1} needs to be considered for high-V_t assignment. Assigning other transistors to high-V_t will only decrease the performance of the gate with no reduction in leakage current. For example, we do not need to assign high-V_t to t_{p2} in the 10 state or t_{p1} in the 01 state, since high-V_t nMOS t_{n2} or t_{n1} has already suppressed the current flow from V_{dd} to Gnd, reducing the subthreshold leakage current. On the other hand, in state 11 *both* t_{p1} and t_{p2} must be assigned high-V_t in order to reduce leakage, since they are parallel devices.

We can partition the transistors into so-called V_t-groups, corresponding to the minimum sets of transistors that need to be set to high-V_t to reduce leakage in a particular state assignment. For the two-input NAND gate in Fig. 2, three V_t-groups exist as shown. The concept of V_t-groups can be easily applied to more complex structures in which case it may be possible that a transistor belongs more than one V_t-group. It is clear that we can restrict ourselves to setting only entire V_t-groups to either high or low-V_t. By considering only V_t-groups, instead of individual transistors, we therefore significantly reduce the number of possible V_t assignments and the optimization complexity. For example, in the two-input NAND gate of Fig. 2 the number of all possible V_t assignments is $2^4 = 16$. However, based on input states, we have only three groups for high-V_t assignment and one assignment for all low-V_t transistors. In Table I, we show the leakage current for the NAND2 in Fig. 2 for different input states and V_t-group assignments. Column 3 shows the leakage current when we use high-V_t for one or more V_t-groups that are OFF in a particular input state. In column 4 and 5, the leakage current with all transistors assigned to, respectively, high-V_t and low-V_t is shown. We can see that in states 01, 10, and 11 only a single V_t-group is a candidate for high-V_t assignment. Also,

TABLE I
LEAKAGE CURRENT OF NAND2 GATE

Input state (AB)	Assigned high-V_t group	Leakage current (pA)		
		with group assignment	with all high-V_t	with all low-V_t
00	2	24.9	7.2	286.7
	3	9.8		
	2 and 3	7.2		
01	2	26.6	26.6	1054
10	3	25.7	24.4	922.6
11	1	14.2	14.2	357.2

setting only this one V_t-group to high-V_t results in equal or nearly equal leakage compared with the leakage when all transistors are assigned high-V_t demonstrating the effectiveness of the approach. In state 00, three high-V_t assignments are possible: group 2, group 3, and both group 2 and 3. However, the leakage current with both groups assigned to high-V_t is only slightly better than that with only one group set to high-V_t, and assigning group 3 to high-V_t reduces leakage somewhat more than assigning group 2 to high-V_t. Hence, it is clear that we need to only consider assignment of group 3 to high-V_t without significant loss in optimality.

Table I shows that the leakage current varies considerably as different groups associated with different input states are set to high-V_t. At the same time, the impact of different high-V_t group assignments on the performance of the circuit must be considered. By setting only a single group to high-V_t, the performance degradation is restricted to only a single signal transition direction and is also reduced compared to high-V_t assignments where most or all transistors are set to high-V_t. Therefore, the performance/power tradeoff of V_t assignment with known input state is much improved compared with that with unknown input state.

The input state of a gate effects which transition direction is degraded by a high-V_t group assignment to a gate. Also, the position of the high-V_t group in a stack of transistors changes the impact of a high-V_t group assignment on the different input to output gate delays. Therefore, the input state of a gate must be chosen such that its associated high-V_t group results in the least degradation of the critical paths in the circuit. However, only the input state of the circuit as a whole can be controlled and the logic correlations of the circuit restrict the possible assignments of gate input states. Therefore, selection of the circuit input state and of which gate is assigned a high-V_t group must be made simultaneously to obtain the maximum improvement in leakage current with minimum loss in performance.

B. Exact Solution to V_t and State Assignment

The size of the input state space is 2^n, where n is the number of circuit inputs. For each input state assignment, there are two possible V_t assignments for each gate (one high-V_t group which is pre-determined by its input state for leakage reduction, and all low-V_t for best performance). The total number of possible V_t assignments is therefore 2^m, where m is the number of gates in the circuit and the total size of the search space is 2^{n+m}.

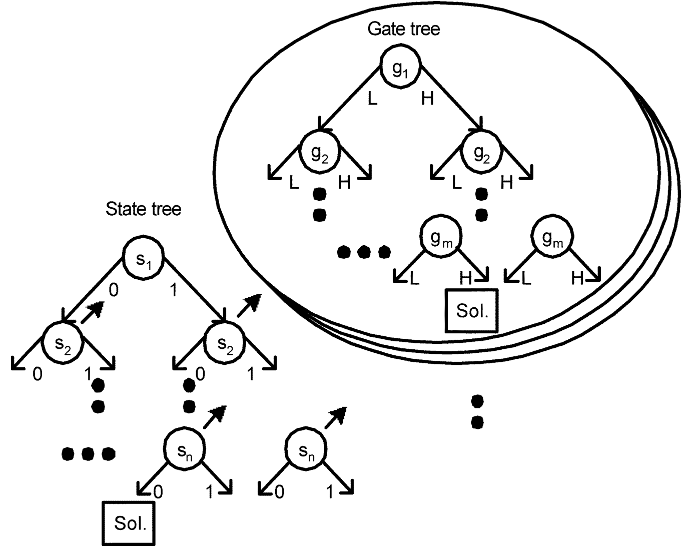

Fig. 3. State tree with gate tree at each node.

In order to find an exact solution to the problem, we developed an efficient branch-and-bound method that simultaneously explores the state and V_t assignments and that exploits the characteristics of the problem to obtain efficient pruning of the search space to improve the run time. Due to the exponential nature of the problem, an exact solution is only possible for very small circuits. However, the exact approach is still useful as the proposed heuristics are based on it.

We use two types of branch and bound trees. The first branch-and-bound tree determines the input state of the circuit and is referred to as the *state tree*. The nodes of the state tree correspond to the input variables of the circuit inputs. Each node of the state tree is associated with a so-called *gate tree* which is searched to determine the group V_t assignment. In other words, for a state tree with k nodes, there exist k copies of the gate tree. Each node in a particular gate tree corresponds to a gate in the circuit, as shown in Fig. 3. Each node has two fanout edges, representing the assignment of that gate with all low-V_t groups (left branch) or with one high-V_t group, as determined by the input state of the gate (right branch).

At the root of the state tree, the state of all input variables is unknown. As the algorithm proceeds down the tree, the state of one input variable becomes defined with each level that is traversed. At each node in the state tree, a solution of leakage current can be obtained by traversing the gate tree. Note that the gate

tree may be traversed both with a completely known input state at the bottom of the state tree as well as with a partially or completely unknown input state, at higher levels of the state tree.

For each node in the state and gate tree, upper and low bounds on the leakage current are computed incrementally. Note that early in the state tree the bounds on leakage will be very loose since the state of the circuit is only partly defined. As the algorithm traverses down the state tree, the input state becomes more defined and the leakage bounds become closer. Similarly, the leakage bounds are very wide at the top of each gate tree, as the V_t assignment of all gates are unknown, and becomes progressively tighter as the algorithm traverses down the tree. Only at the bottom of *both* the state tree and its associated gate tree do the upper and low bounds on leakage coincide. The algorithm first traverses down to the bottom of the tree and then returns back up, to traverse down unvisited branches in DFS manner. During the search, a tree branch is pruned when it has a lower bound on leakage that is worse than the best upper bound on leakage that has been observed so-far. In addition to pruning based on leakage bounds, we also compute a lower bound on the circuit delay at each node in the gate tree traversal and prune all branches whose lower bound exceeds the specified delay constraint. Computation of the delay bounds is also performed incrementally.

Also, early in the state tree, computation of the exact minimum V_t assignment by traversing the gate tree is not meaningful since even at that bottom of the gate tree there is considerable uncertainty in the leakage current due to the unknown input state. Therefore, the gate tree is searched only partially at the higher levels of the state-tree which results in slightly more conservative bounds, but an overall improvement in the run time of the algorithm.

The gate tree is also searched in DFS manner and edges are pruned based on the computed leakage bounds. During the downward traversal of the gate tree, the high-V_t branch is always selected, provide it meets the delay constraint. This is due to the fact that the high-V_t branch always has less leakage current than the low-V_t branch. Only if the lower bound on the delay of the high-V_t branch exceeds the delay constraint, is the low-V_t branch selected and is the high-V_t branch pruned.

Finally, the gates in the circuit are assigned to nodes in the gate tree in topological order to enable incremental delay computation. Gates of equal topological level are further sorted by decreasing leakage to improve the pruning of the search space. The input signals of the circuit are also assigned to nodes in the state tree in specific order. We want to place inputs whose state assignment strongly influences the total leakage of the circuit near the top of the state tree. We estimate the influence of each input signal on the circuit leakage by taking the sum of the leakage current of all gates connected to the input signal. This input variable ordering is similar to that used in [25]. Fig. 4 shows a simple circuit example of state tree and gate tree searching.

1) Incremental Leakage Bound Computation: During the traversal of the gate tree, some of the gates will have a known V_t assignment and others, which have not been visited, will have an unknown V_t assignment. As shown in Fig. 5, a lower bound on the leakage is computed by assuming all unknown

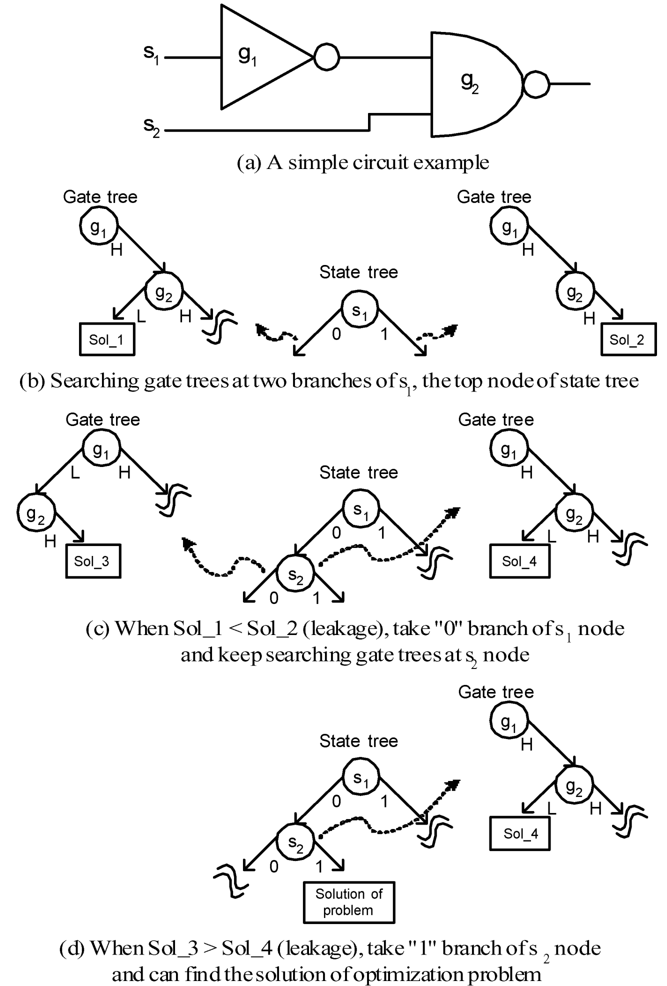

Fig. 4. Example of state tree and gate tree searching.

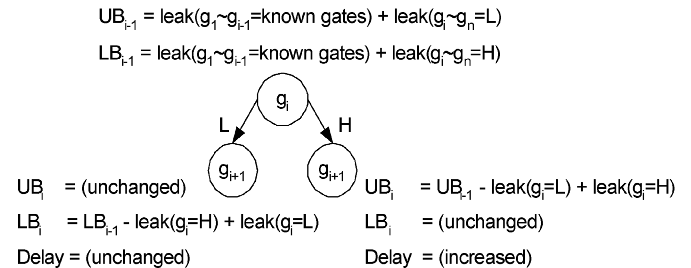

Fig. 5. Incremental leakage bound computation.

gates have a high-V_t group assignment and an upper bound is computed by assuming all unknown gates have a low-V_t group assignment. As the high branch is taken in the downward traversal, only the upper bound is updated (decreased) whereas when a low branch is taken, only the lower bound must be updated and is increased.

2) Incremental Delay Bound Computation: Similar to the leakage current bounds, a lower bound on the delay is computed assuming all unknown gates have low-V_t group assignments. Delay is changed only when a high branch is taken in the traversal and is computed incrementally. We first compute the slack of the circuit for all circuit nodes at the start of the tree traversal with all V_t assignments assumed to be low-V_t. When a group changes from a low to a high-V_t group assignment during

the traversal, the slack of that gate will be updated. However, the V_t change of the gate will affect not only the gate itself but also the delays of fanout gates due to the slope change at the output of the changed gate. Since the slope at the output of the changed gate will become slower due to its high-V_t assignment, the delay of all fanout gates will increase, resulting an overall increased circuit delay. Ignoring the effect of slope change on fanout gates will therefore result in the computation of an *optimistic* lower bound which ensures that the optimal solution is not accidentally pruned. It also enables incremental delay computation, given that the gates are visited in topological ordering. As gates are visited, the changed input slope, due to high-V_t assignments of a fanin gate, is processed to ensure that an exact delay bound is computed at the bottom of the gate tree.

C. Heuristic Solution to V_t and State Assignment

We propose two fast heuristics that can be applied to large circuits and that produce high quality solutions. The proposed heuristic are based on the exact method described in Section III-B, and are discussed below.

1) Heuristic 1: In this heuristic, the state and gate tree search is limited to only one downward traversal. Note that while only a single traversal of the state tree is performed, at each node of the state tree the decision to follow the left or right child node is based on the computed bounds of the leakage using the gate tree. Each downward traversal of the gate tree visits m nodes, where m is the number of gates in the circuit. We perform exactly two such traversals at each state tree node, leading to a total run time complexity that is $O(nm)$, where n is the number of circuit inputs.

2) Heuristic 2: In the second heuristic, the state tree is searched more extensively, subject to a fixed run time constraint, while the gate tree search is kept to a single downward traversal for each state tree node. Experimentally, it was found that the quality of the first bottom node reached in the gate tree search is near the optimal V_t assignment. This is due to the fact that the gate tree always chooses the high-V_t child in its downward traversal which tends to produce a high quality result. This is in contrast to the state tree, where choosing the correct child during the downward traversal was found to be much more difficult. Therefore, the solution quality was found to improve most by searching the state tree more extensively, subject to a run time constraint, while limiting the gate tree search to a single downward traversal.

D. V_t Assignment Control Within Stacks

We assume the ability to assign V_t on an individual basis within stacks of transistors. Although it is generally possible to assign the V_t of each transistor in a stack individually, this may result in the need for increased spacing between the transistors in order not to violate design rules and ensure manufacturability [26]. Hence, at times it may be desirable to restrict the assignment of V_t such that all transistors in a stack are uniform. In this case, less flexibility exists in the assignment of V_t, and hence the obtained tradeoff in delay and leakage will degrade to some extent. In Section V-A, we present results showing the impact on the leakage optimization when uniform stack assignments are enforced in the library.

IV. Leakage Reduction Method for Both Subthreshold and Gate Leakage Current

A. Leakage Reduction Approach

The proposed leakage optimization method performs simultaneous assignment of standby mode state, high-V_t, and thick-oxide transistors. The proposed method 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. This is due to the fact that if a transistor that is OFF, gate leakage is significantly reduced and hence the transistor only needs to be considered for high-V_t assignment. Conversely, a transistor that, given a particular input state, is ON 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 in standby mode, 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. However, given a known input state, we can avoid assignment of transistors to both high-V_t and thick-oxide, thereby significantly improving the obtained leakage/delay tradeoff.

Furthermore, depending on the input state of a circuit, only a subset of transistors needs to be considered for high-V_t or thick-oxide, as discussed in Section III-A. For instance, in a stack of several transistors that are OFF, 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 also has 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 reduced. 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 illustrate the advantage of high-V_t and thick-oxide assignment with a known input state for a two-input NAND and NOR gate in Fig. 6. In Fig. 6(a) a two-input NOR gate is shown with input state 01. Since only pMOS transistors p_2 is OFF in the pull-up stack, it is the only transistor that needs to be set to high-V_t to reduce the subthreshold leakage of the gate. Similarly, only nMOS transistor n_2 exhibits gate leakage and needs to be assigned thick-oxide to reduce I_{gate}. Hence only two out of four transistors are affected while the total leakage current is reduced by nearly the same amount as when all transistors in the gate are set to high-V_t *and* thick-oxide simultaneously. As a result, the delay of the rising input transition at input i_1 is unaffected by the high-V_t and thick-oxide assignments, while the other transitions are affected only moderately.

In Fig. 6(b), the worst-case input state for a NOR2 gate is shown, which is when both inputs are 1. In this case, both nMOS devices must be assigned to thick-oxide to reduce I_{gate}, while at least one pMOS device is set to high-V_t. Depending on the delay requirements, the best input state is either the state 01 shown in Fig. 6(a), or the state 00, shown in Fig. 6(c), which requires only two transistors to be set to high-V_t. Hence, it is clear that the input state significantly impacts the ability to effectively assign high-V_t and thick-oxide without degrading the

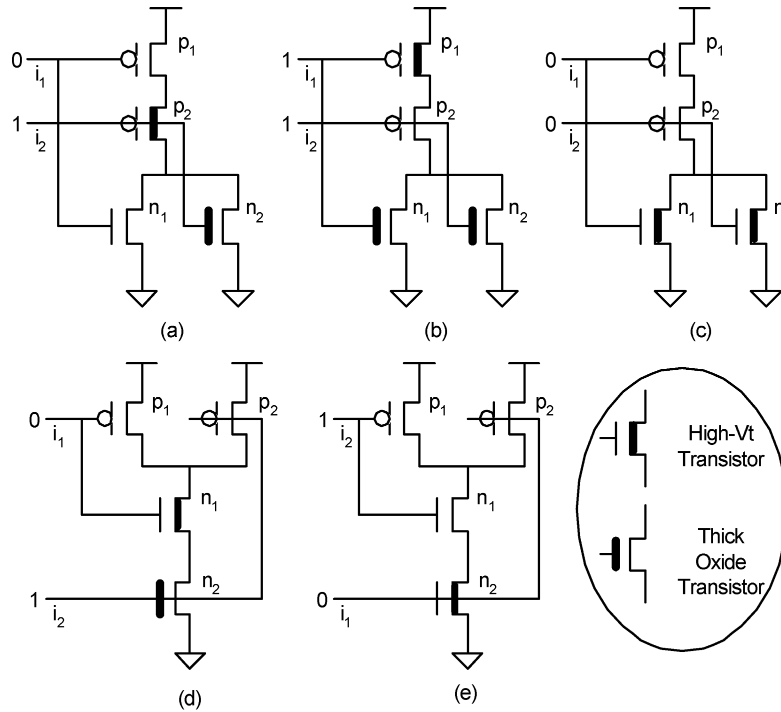

Fig. 6. High-V_t and thick-oxide assignments at different input states.

performance of the circuit. This leads to the need for a simultaneous optimization approach where both the input state and the high-V_t and thick-oxide assignments are considered simultaneously under delay constraints.

In addition to high-V_t and thick-oxide assignment, we also take advantage of the I_{gate} dependence on input pin ordering to reduce leakage current [11]. This is illustrated in Fig. 6(d), for a two-input NAND gate with input state 01. In order to effectively reduce the leakage under this input state, nMOS transistor n_1 must be assigned to high-V_t and nMOS transistor n_2 must be assigned to thick-oxide. However, if input pins i_1 and i_2 are reordered, with i_1 positioned at the bottom of the stack, as shown in Fig. 6(e), the V_{gs} and V_{gd} voltage of nMOS transistor n_1 will be reduced from V_{dd} to approximately one V_t drop. Hence, the gate leakage current of n_1 will be substantially reduced and can be ignored. After reordering the input pins, it is necessary to only set nMOS transistor n_2 to high-V_t without further assignments of thick-oxide transistors. It should be noted that pin reordering will impact the delay of the circuit and hence some performance penalty might be incurred. However, this penalty will be readily offset by the elimination of the thick-oxide assignment in the pull-down stack. In this paper, we therefore consider combined input state assignment with pin-reordering and V_t/T_{ox} assignment.

B. Cell Library Construction

In order to perform simultaneous V_t, T_{ox} and state assignment, it is necessary to develop a library in which the necessary V_t and T_{ox} versions are available for each cell type. After such a library has been constructed, the process of assigning V_t and T_{ox} assignments can be performed by simply swapping cells from the library. Since different V_t and T_{ox} variations do not alter the footprint of a cell, the leakage optimization can be performed either before or after final placement and routing.

For each gate and input state, a number of different T_{ox} and V_t assignments are possible, providing different delay/leakage tradeoff points. For the fastest and highest leakage tradeoff point, all transistors are assigned to low-V_t and thin-oxide, such as the NAND2 gate shown in Fig. 7(a). On the other hand, for the slowest and lowest leakage version of the cell all transistors contributing to leakage are assigned either high-V_t or thick-oxide. For instance, for the NAND2 gate with input state 11, shown in Fig. 7(b), all transistors affect the leakage current and both nMOS transistors are assigned thick-T_{ox} while both pMOS transistors are assigned high-V_t to obtain the minimum leakage/maximum delay tradeoff point.

In addition to the fastest version and minimum leakage version of the cell, a number of other intermediate tradeoff points can be constructed for a cell by assigning only some of the transistors that contribute to leakage to high-V_t or thick-T_{ox}. These cell versions would have lower leakage than the fastest cell version but would be faster than the lowest leakage version. It is clear that a large number of possible cell versions can be constructed if all possible tradeoff points are considered for each possible input state. While a larger set of cell versions provides the optimization algorithm with more flexibility, and hence a more optimal leakage result, it also increases the size of the library, which is undesirable. Therefore, we initially restrict our library to at most four different tradeoff points for each input state of a library cell, which are: 1) the minimum delay, shown in Fig. 7(a), 2) minimum leakage, shown in Fig. 7(b), 3) fast falling transition but slow rising transition, with intermediate leakage, shown in Fig. 7(c), and 4) fast rising transition but slow falling transition with intermediate leakage, shown in Fig. 7(d). Although other possible tradeoff points could be considered, we

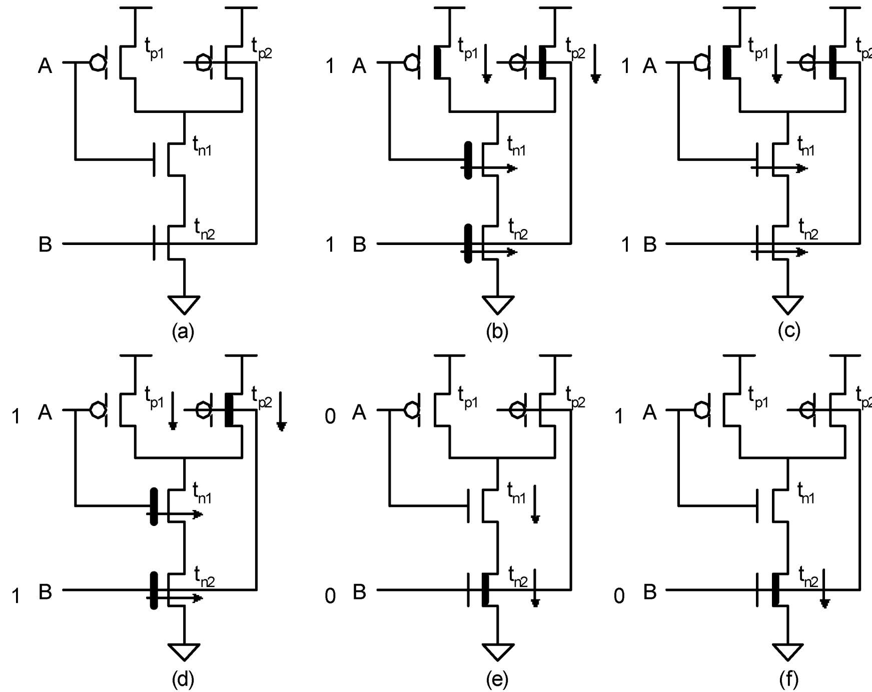

Fig. 7. Complete V_t-T_{ox} versions of NAND2 gate.

TABLE II
TRADEOFFS FOR DIFFERENT V_t-T_{ox} VERSIONS OF NAND2 GATE

State (AB)	Cell	Total leakage current (nA)	Normalized rise delay		Normalized fall delay	
			Pin A	Pin B	Pin A	Pin B
11	Minimum delay (a)	270.4	1	1	1	1
	Fast rise delay (d)	109.1	1	1.36	1.27	1.27
	Fast fall delay (c)	91.4	1.36	1.36	1	1
	Minimum leakage (b)	19.5	1.36	1.37	1.27	1.27
00	Minimum delay (a)	41.2	1	1	1	1
	Minimum leakage (e)	14.0	1	1	1.12	1.16
10	Minimum delay (a)	91.8	1	1	1	1
	Minimum leakage (f)	13.3	1	1	1.12	1.16

empirically found that these four points yield good optimization results and provide a systematic approach for constructing all versions of a cell.

In principle, using four possible tradeoff points for each input combination could result in as many as 16 (4×4) cell versions for a two-input gate. However, in practice, many of the cell versions are shared between different input states. Also, in some cases not all four tradeoff points are realizable and hence the total number of cell versions is significantly less. We illustrate this for the NAND2 gate for input state 00. The fastest cell version is again shown in Fig. 7(a) and is shared for all input combinations, and the minimum leakage version is shown in Fig. 7(e). Note that only one transistor needs to be set to high-V_t to achieve minimum leakage for this input state. This results from the fact that pMOS devices have negligible gate leakage in the target technology and only one transistor in a stack needs to be set to high-V_t to reduce the leakage through the entire stack. Hence, for the input state 00, only two tradeoff points are needed and only one additional cell version is added to the library.

Input state 10 again requires the assignment of only a single transistor to high-V_t for the minimum leakage version, as shown in Fig. 7(f). This is due to the fact that the gate leakage through the top nMOS transistor n_1 is negligible since its V_{gs} and V_{gd} is reduced to approximately one V_t drop. Only two tradeoff points are therefore required for this input state and both versions are shared with the 00 state. Finally, if the 01 state occurs in the circuit, the optimization will automatically perform input pin swapping for all but the fastest tradeoff point, thereby resulting in no additional cell version. The NAND2 gate therefore requires a total of five cell versions to provide up to four tradeoff points for each input state. In Table II, we show the delay/leakage tradeoffs obtained for each input state using the described approach for the NAND2 gate.

The same process can be applied to each cell in the library to construct the full set of cell versions for the leakage characterization method. Table III shows the number of cell version required for several common gates. Note that the number of cell versions is higher for NOR gates than NAND gates. We

TABLE III
THE NUMBER OF NEEDED LIBRARY CELLS

	4 trade-off points	2 trade-off points	2 trade-off points with 3 cell versions only
Inverter	5	3	3
NAND2	5	3	3
NAND3	5	3	3
NOR2	8	4	3
NOR3	9	5	3

also explored reducing the number of cells by allowing only two tradeoff points for each cell (minimum delay, and minimum leakage), instead of four tradeoff points. In this case, the number of cells for the NAND2 gate reduces to only three versions. The number of cell version required for two tradeoff points for different cell types is shown in Table III, column 3. In column 4, we add one more cell library version – two tradeoff points with only three cell versions. In order to minimize the number of needed library cells, one or two cells of NOR2 or NOR3, respectively, are removed from library with small degradation of leakage/delay tradeoff. Therefore, all gates have only three cells in this option. In Section V-B we compare the final leakage results using the full library with four tradeoff points, the reduced library with only two tradeoff points, and the most restrictive library with only three options per cell.

Finally, we consider V_t and $T_{\rm ox}$ assignment control within stacks similar to the discussion for V_t stack control in Section III-D. However, $T_{\rm ox}$ assignment differs from V_t assignment in that the assignment of $T_{\rm ox}$ to transistors in a stack is already uniform due to use of pin-swapping. This is evident from the five added cell versions for the NAND2 in Fig. 7, and can be easily shown to be true for all cell versions generated under the proposed approach. This is a significant advantage since spacing design rules for different $T_{\rm ox}$ assignments are expected to be more severe than those for spacing between different V_t assignments [26]. However, the V_t assignment is not always uniform as shown in Fig. 7(e), where only a single transistor in a stack is assigned to high-V_t. In the event that a uniform stack is required, both transistors in the stack need to be set to high-V_t, resulting in a slightly worsened delay/leakage tradeoff. Leakage current comparison results between individual versus uniform stack assignment control will be shown in Section V-B.

C. Optimization – Approach and Heuristics

In this section, we present an exact solution and two heuristics to the problem of finding a simultaneous input state, high-V_t and thick-$T_{\rm ox}$ assignments for a circuit under delay constraints. As mentioned, the leakage minimization problem can be formulated as an integer optimization problem under delay constraints. The size of the input state space is 2^n, where n is the number of circuit inputs. As discussed in Section IV-B, for each input state assignment, there are up to four possible V_t-$T_{\rm ox}$ assignments for each gate. Note that while the total number of cell versions can be larger than 4, only four of them need to be considered for each specific input state. For instance, for the NAND2 gate in Fig. 7, only versions (a)-(d) are considered for a 11 input state. Therefore, the total number of possible V_t-$T_{\rm ox}$ assignments is 4^m, where m is the number of gates in the circuit and the total size of the search space is 2^{n+2m}.

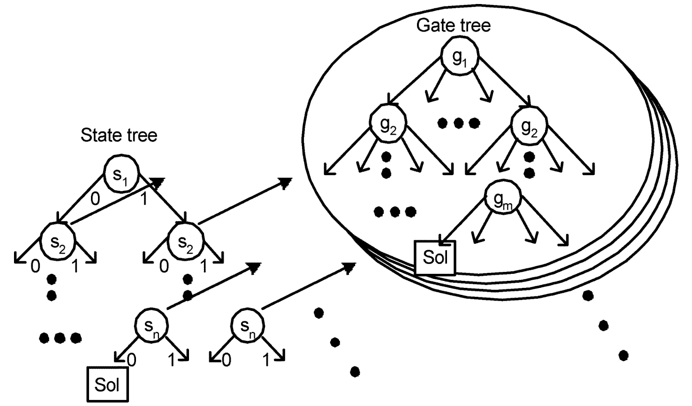

Fig. 8. State tree with gate tree at each node.

In order to find an exact solution to the problem, we extend the branch-and-bound method with Section III-B. The branch and bound algorithm for V_t-$T_{\rm ox}$ and state assignment uses two interdependent search trees: *state tree* and *gate tree*. The state tree is searched to determine the input state of the circuit and the gate tree is searched to determine the V_t-$T_{\rm ox}$ assignment of the circuit, as shown in Fig. 8. The only difference from Section III-B is the gate tree. Each node in a particular gate tree corresponds to a gate in the circuit. Since there are four possible V_t-$T_{\rm ox}$ assignments for a gate, each node of the gate tree has four edges: minimum delay, minimum leakage, fast fall delay with intermediate leakage, and fast rise delay with intermediate leakage. The exponential nature of the problem makes it impossible to obtain an exact solution for substantial circuits, such as $I_{\rm sub}$ minimization approach in Section III-B. Therefore, we also use the two heuristics discussed in Section III-C. In the first heuristic, we perform only one downward traversal search in the state and gate tree. On the other hand, in the second heuristic we continue searching the state tree until a predetermined runtime constraint, however we perform only a single downward search in the gate tree for each state tree node. As discussed in Section III-C, since the quality of the first bottom node of the gate tree is near the optimal V_t assignment, the quality of the solution can be improved by searching the state tree more extensively.

V. RESULTS

A. Subthreshold Leakage Reduction

The proposed methods for simultaneous state and V_t assignment were tested on the ISCAS benchmark circuits [27] and a 64-bit ALU circuit, synthesized using a 0.18 μm industrial

TABLE IV
BENCHMARK CIRCUIT SPECIFICATIONS

Circuit	Number of			Max. level
	Input	Output	Gate	
C432	36	7	177	20
C499	41	32	519	23
C880	60	26	364	25
C1355	41	32	528	27
C1908	33	25	432	40
C2670	233	140	825	21
C3540	50	22	940	41
C5315	178	123	1627	40
C6288	32	32	2470	119
C7552	207	108	1994	44
alu64	131	64	1803	204

TABLE V
LEAKAGE CURRENT COMPARISON BETWEEN HEURISTICS

	Minimized leakage current (nA) (reduction factor: vs. average leakage current, run time unit: sec.)															
	Avg. I_{leak} by random (10K) vectors	0% in low-V_t/high-V_t delay range					5% in low-V_t/high-V_t delay range					10% in low-V_t/high-V_t delay range				
		Heuristic 1			Heuristic 2		Heuristic 1			Heuristic 2		Heuristic 1			Heuristic 2	
		I_{leak}	X	Time	I_{leak}	X	I_{leak}	X	Time	I_{leak}	X	I_{leak}	X	Time	I_{leak}	X
C432	32.9	7.7	4.3	1	4.3	7.7	4.9	6.7	1	3.6	9.2	4.7	7.0	1	3.6	9.1
C499	94.0	13.2	7.1	3	11.3	8.3	13.1	7.2	2	11.6	8.1	9.7	9.6	2	9.7	9.6
C880	73.4	9.7	7.5	4	8.9	8.3	8.9	8.2	3	8.3	8.8	8.9	8.3	4	8.3	8.8
C1355	85.1	19.0	4.5	3	12.7	6.7	14.6	5.8	3	11.7	7.3	12.0	7.1	3	11.0	7.7
C1908	82.8	19.0	4.3	2	15.1	5.5	15.5	5.3	2	12.2	6.8	13.4	6.2	2	10.3	8.0
C2670	162.5	12.7	12.8	58	12.5	13.0	12.7	12.8	55	12.4	13.1	14.3	11.3	55	12.2	13.3
C3540	173.1	20.1	8.6	10	16.4	10.6	20.5	8.4	10	14.6	11.8	17.4	10.0	9	14.5	11.9
C5315	309.1	26.4	11.7	169	25.9	11.9	27.5	11.2	164	25.2	12.3	28.5	10.9	165	25.2	12.2
C6288	451.5	157.5	2.9	47	153.9	2.9	145.5	3.1	44	141.4	3.2	135.8	3.3	43	128.4	3.5
C7552	385.8	31.0	12.4	330	30.6	12.6	30.8	12.5	330	30.1	12.8	30.7	12.6	328	29.6	13.0
alu64	332.3	46.0	7.2	405	43.6	7.6	47.2	7.0	408	44.5	7.5	43.0	7.7	406	42.0	7.9
AVG			7.6			8.6		8.0			9.2		8.5			9.6

Heuristic 2 was limited to a runtime of 1800 seconds.

library with Synopsys. (Table IV shows specifications of the benchmark circuits.) This technology has a difference of $14\times$ ($10\times$) in I_{sub} and 16% (15%) in delay between low-V_t and high-V_t nMOS (pMOS) devices. The leakage current for each V_t version of a cell was computed using SPICE simulation and stored in precharacterized tables. Delay computation was performed based on the Synopsys table delay model and was verified to match with Synopsys timing analysis delay reports. In addition to the proposed methods, traditional methods using only state or V_t assignment were also implemented for comparison. The state-only assignment was implemented using the approach discussed in [7], [25] while for V_t-only assignment a method similar to the sensitivity-based approach of [17] was used. Experiments were performed on a 1.8-GHz Pentium 4 machine.

Table V compares the leakage results obtained by the three proposed heuristics for three delay constraints to the average leakage computed using 10 000 random input vectors. The columns marked 0%, 5%, and 10% refer to leakage minimization results when the delay constraints were set at 0%, 5%, and 10% respectively, of the full delay range between all low-V_t and all high-V_t circuit delay, as illustrated in Fig. 9. The 0% column is therefore the most stringently constrained optimization as it corresponds to the best obtainable delay for the circuit (no performance penalty). Note that a simple replacement of all low-V_t devices with all high-V_t ones would yield a $\sim$ 20% circuit delay increase. Since the average leakage current with 10 000 random input vectors is computed with all low-V_t transistors, it also corresponds to a 0% delay criterion. Runtimes for heuristic 1 are given in Table V in seconds. Heuristic 2 was limited to a runtime of 1800 s (30 min). We report the reduction factor relative to the average leakage current over the 10 000 random vectors. Heuristic 2 has $\sim$ 10% lower leakage results than heuristic 1 at 5% delay point across the benchmark circuits. However, heuristic 2 has a 4–5 $\times$ runtime overhead for large circuits ($\sim 1000\times$ for small circuits) over heuristic 1.

Delay with all low-V_t
0%
Delay with all high-V_t
100%
5% 10% 50%

Fig. 9. Delay point from all low-V_t to all high-V_t range.

TABLE VI
LEAKAGE CURRENT COMPARISON WITH TRADITIONAL TECHNIQUES

	Minimized leakage current (nA)														
	Avg. I_{leak} by random (10K) vectors	State only assignment		V_t only & proposed heuristic (reduction factor: vs. average leakage current)											
				0% in low-V_t/high-V_t delay range				5% in low-V_t/high-V_t delay range				10% in low-V_t/high-V_t delay range			
				V_t only		Heuristic 1		V_t only		Heuristic 1		V_t only		Heuristic 1	
		I_{leak}	X	I_{leak}	X	I_{leak}	X	I_{leak}	X	I_{leak}	X	I_{leak}	X	I_{leak}	X
C432	32.9	26.3	1.25	30.8	1.1	7.7	4.3	29.5	1.1	4.9	6.7	29.2	1.1	4.7	7.0
C499	94.0	86.1	1.09	85.0	1.1	13.2	7.1	57.2	1.6	13.1	7.2	40.4	2.3	9.7	9.6
C880	73.4	63.7	1.15	64.6	1.1	9.7	7.5	63.9	1.1	8.9	8.2	20.2	3.6	8.9	8.3
C1355	85.1	81.4	1.04	94.0	0.9	19.0	4.5	65.1	1.3	14.6	5.8	53.4	1.6	12.0	7.1
C1908	82.8	74.6	1.11	67.0	1.2	19.0	4.3	46.5	1.8	15.5	5.3	30.3	2.7	13.4	6.2
C2670	162.5	146.2	1.11	44.7	3.6	12.7	12.8	39.7	4.1	12.7	12.8	27.8	5.8	14.3	11.3
C3540	173.1	155.7	1.11	161.9	1.1	20.1	8.6	148.4	1.2	20.5	8.4	82.4	2.1	17.4	10.0
C5315	309.1	283.1	1.09	290.6	1.1	26.4	11.7	289.7	1.1	27.5	11.2	108.3	2.9	28.5	10.9
C6288	451.5	412.4	1.09	417.0	1.1	157.5	2.9	259.5	1.7	145.5	3.1	233.0	1.9	135.8	3.3
C7552	385.8	352.3	1.10	360.2	1.1	31.0	12.4	353.5	1.1	30.8	12.5	350.9	1.1	30.7	12.6
alu64	332.3	294.5	1.13	312.8	1.1	46.0	7.2	288.5	1.2	47.2	7.0	230.1	1.4	43.0	7.7
AVG			1.12		1.3		7.6		1.6		8.0		2.4		8.5

In Table VI, we compare the proposed approach with traditional techniques, including state-only and V_t-only assignment methods. The state-only assignment method was limited to a runtime of 1800 s (30 min). The results demonstrate that substantial improvement in standby leakage current can be obtained using the proposed methods, with an average improvement of $\sim 80\%$ (5–6 $\times$) for the 0% and 5% delay constraints over V_t-only assignment.

Table VII compares leakage current results for both individual and uniform stack control. Since uniform stack control degrades the delay/leakage tradeoff as discussed in Section III-D, the results for uniform stack assignment exhibit less leakage reduction than those of individual stack control. It is interesting to note, however, that the leakage current degradation by moving to a less fine-grained threshold voltage assignment scheme is not overly large. This implies that even with manufacturing constraints, the proposed technique provides significant leakage savings. Fig. 10 further illustrates the normalized total leakage current in both individual and uniform stack control scenarios.

Finally, Fig. 11 plots the leakage results for the proposed method and the two traditional methods as a function of the delay constraint for circuit c6288. The optimization was performed for a range of delay constraints. The proposed method provides its largest improvements at tight delay constraints. This is due to the fact that, as the delay constraint becomes looser, more transistors can be set to high-V_t in both approaches, and the relative advantage of the proposed approach is reduced. However, leakage reduction is most challenging under tight performance constraints for which the proposed technique holds promise.

B. Leakage Reduction for Both Subthreshold and Gate Leakage

The proposed methods for simultaneous state, V_t and $T_{\rm ox}$ assignment were implemented on a number of benchmark circuits [27] synthesized using a library based on a predictive 65-nm process [20]. In this technology, the difference in $I_{\rm gate}$ for the thick-oxide nMOS devices versus the thin-oxide device is 11 $\times$, whereas $I_{\rm sub}$ is reduced by 17.8 $\times$ (16.7 $\times$) when replacing a low-V_t nMOS (PMOS) device with a high-V_t version. Table VIII shows relative leakage and delay values at the four possible V_t and $T_{\rm ox}$ assignments for nMOS devices in this technology. A comparison of our first and second heuristics along with average leakage computed using 10 000 random input vectors is shown in Table IX. The total leakage current value is given in mA and runtime is given in seconds. In heuristic 2, we set the runtime limit as 1800 s (30 min). The average leakage computed using the random vectors can be used to approximate the standby mode leakage if state assignment as well as dual-V_t and dual-$T_{\rm ox}$ techniques were not employed. Again, the delay

TABLE VII
LEAKAGE CURRENT COMPARISON BETWEEN INDIVIDUAL AND UNIFORM STACK CONTROL

	Minimized leakage current (nA)						
	Avg. I_{leak} by random (10K) vectors	5% in low-V_t/high-V_t delay range (reduction factor: vs. average leakage current)					
		V_t only assignment		Heuristic 1			
				Individual control		Uniform control	
		I_{leak}	X	I_{leak}	X	I_{leak}	X
C432	32.9	29.5	1.1	4.9	6.7	6.8	4.8
C499	94.0	57.2	1.6	13.1	7.2	12.5	7.5
C880	73.4	63.9	1.1	8.9	8.2	9.1	8.1
C1355	85.1	65.1	1.3	14.6	5.8	23.7	3.6
C1908	82.8	46.5	1.8	15.5	5.3	15.7	5.3
C2670	162.5	39.7	4.1	12.7	12.8	12.9	12.6
C3540	173.1	148.4	1.2	20.5	8.4	24.1	7.2
C5315	309.1	289.7	1.1	27.5	11.2	28.5	10.9
C6288	451.5	259.5	1.7	145.5	3.1	163.1	2.8
C7552	385.8	353.5	1.1	30.8	12.5	31.3	12.3
alu64	332.3	288.5	1.2	47.2	7.0	44.6	7.5
AVG			1.6		8.0		7.5

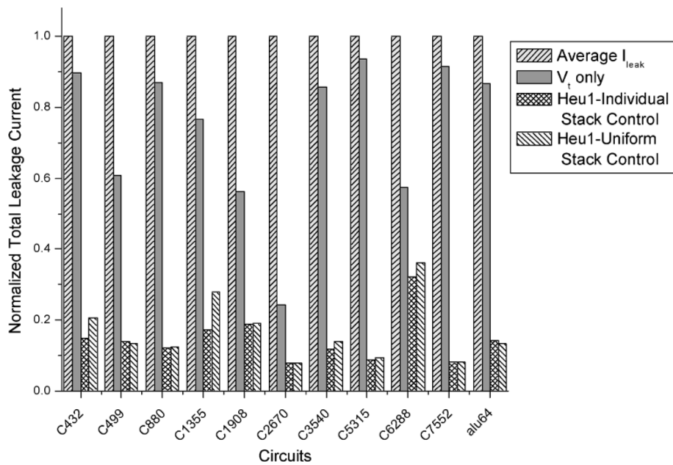

Fig. 10. Leakage current comparison between individual and uniform stack control.

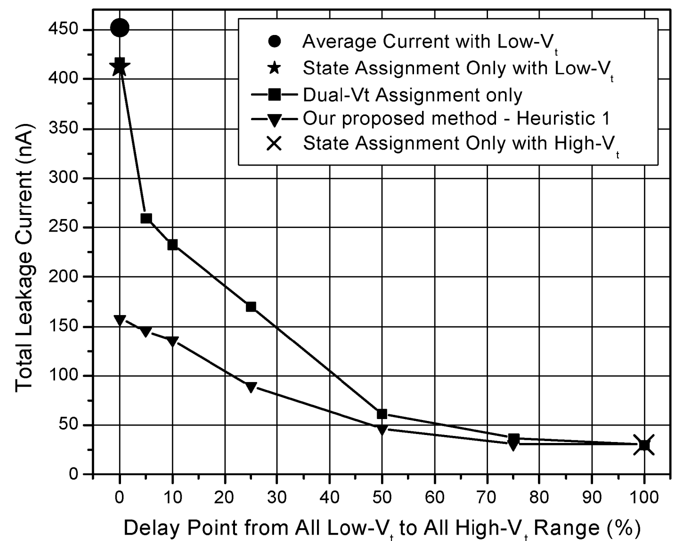

Fig. 11. Leakage current comparison for c6288.

range points used in all results are defined by a percentage of the *maximum possible* delay that is associated with moving from an all low-V_t and thin-oxide design to an all high-V_t and thick-oxide implementation. Note that a simple replacement of all fast devices with their slowest counterparts would yield a $\sim$ 70% circuit delay increase. Thus, when interpreting the results in this section, a 5% delay point indicates that the circuit after V_t and T_{ox} assignment has a delay that is approximately 4% larger than the original fastest implementation.

As shown in Table IX, heuristic 2 generally provides somewhat better results but at much greater runtimes. On average, heuristic 2 provides $\sim$ 10% lower leakage current than heuristic 1 across these benchmarks at the 5% delay point, similar to the results in Section V-A. The improvement of the two proposed heuristics compared to the average leakage without state, V_t or T_{ox} assignment is dramatic and approaches 7 × at the 10% delay point in the best-worst delay range. At just a 5% delay range point, the reduction in total standby leakage is 5.3–6 × with a maximum improvement of 8.6 × for heuristic 2 in circuit c2670.

In Table X we compare our results to other standby mode techniques, including state assignment alone and simultaneous state and V_t assignment (as in the previous section). The total leakage current value is given in mA. Again, we report the reduction factor in relation to the average leakage current with 10 000 random vectors for consistency. We first point out that state assignment alone, which we accomplish by searching the state tree only, achieves very little improvement in standby mode leakage; about 6% on average. By adding V_t assignment, the algorithm of the first proposed method shows an average reduction of 58% beyond state assignment alone at a 5%

TABLE VIII
COMPARISON OF LEAKAGE AND DELAY BETWEEN FOUR POSSIBLE V_t-T_{ox} ASSIGNMENTS 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	0.00	1.26
High	Thick	0.05	0.03	0.02	1.69

TABLE IX
LEAKAGE CURRENT COMPARISON BETWEEN HEURISTICS WITH FOUR-OPTION, INDIVIDUAL STACK CONTROL LIBRARY

	Minimized leakage current (μA) (reduction factor: vs. average leakage current, run time unit: sec.)															
	Avg. I_{leak} by random (10K) vectors	0% in the best-worst delay range					5% in the best-worst delay range					10% in the best-worst delay range				
		Heuristic 1			Heuristic 2		Heuristic 1			Heuristic 2		Heuristic 1			Heuristic 2	
		I_{leak}	X	Time	I_{leak}	X	I_{leak}	X	Time	I_{leak}	X	I_{leak}	X	Time	I_{leak}	X
C432	24.5	8.2	3.0	3	5.4	4.6	7.7	3.2	2	3.2	7.6	5.5	4.5	2	3.0	8.2
C499	65.8	32.2	2.0	7	31.1	2.1	26.1	2.5	7	24.6	2.7	22.7	2.9	6	20.8	3.2
C880	50.1	10.3	4.9	8	9.2	5.5	8.5	5.9	7	8.3	6.1	8.5	5.9	7	7.0	7.1
C1355	70.8	20.4	3.5	8	20.4	3.5	15.8	4.5	6	13.1	5.4	9.9	7.1	6	9.9	7.1
C1908	56.7	17.4	3.3	5	16.9	3.4	14.8	3.8	4	13.6	4.2	13.2	4.3	5	10.5	5.4
C2670	104.7	14.9	7.0	82	14.7	7.1	12.3	8.5	78	12.2	8.6	13.5	7.8	78	11.3	9.3
C3540	128.5	27.7	4.6	20	23.7	5.4	22.1	5.8	18	19.9	6.4	18.6	6.9	17	17.4	7.4
C5315	221.2	36.6	6.0	219	35.9	6.2	30.0	7.4	213	30.0	7.4	28.4	7.8	202	27.6	8.0
C6288	346.8	153.6	2.3	75	146.0	2.4	112.2	3.1	64	101.4	3.4	84.1	4.1	59	75.6	4.6
C7552	270.0	34.9	7.7	410	33.4	8.1	32.2	8.4	404	31.8	8.5	30.3	8.9	399	30.2	8.9
alu64	260.0	48.7	5.3	468	46.8	5.6	43.4	6.0	464	41.6	6.3	34.3	7.6	458	33.1	7.9
AVG			4.5			4.9		5.4			6.0		6.2			7.0

Heuristic 2 was limited to a runtime of 1800 seconds.

delay point. The full V_t, T_{ox} and state assignment approach provides an additional 53% reduction in current *beyond* state and V_t assignment for the 5% delay point. Note that the amount of leakage current reduction achieved by V_t, T_{ox}, and state assignment for total leakage ($I_{\mathrm{sub}} + I_{\mathrm{gate}}$) is smaller than that achieved by V_t and state assignment when considering only I_{sub} (Table VI of Section V-A). This is due to the worsened performance-leakage current tradeoff when considering both I_{sub} and I_{gate} compared to I_{sub} alone. Introducing thick-oxide thickness transistors for gate leakage current minimization yields a large delay impact. For example, the 180-nm technology under study has $\sim 15\%$ delay spread across V_t choices (Section V-A) while the projected 65-nm technology has $\sim 70\%$ delay difference across V_t and T_{ox} choices for roughly comparable normalized leakage reductions (Table VIII). Note, however, that the V_t, T_{ox} and state assignment approach still achieves much better leakage current reductions (additional 53% reduction) than the V_t and state assignment approach in the same 65-nm technology.

Table XI provides a comparison of results using the various cell library options; four and two tradeoff points with individual stack control, and also with uniform stacks. The main result in Table XI is that there is very little leakage current penalty when moving from a full four-option library to a simpler two-option library. There are several cases where the smaller library outperforms the larger library due to the heuristic nature of the algorithm used (heuristic 1 is used in this table). Since the library size required in the two-option scenario is roughly half that of the four-option, we conclude that the use of two-option represents a very good tradeoff between library complexity and potential leakage reduction. Moreover, we can see that the simplest cell library of the two-option with a reduced number of cells provides good leakage reduction results. In general, a reduced number of cells degrades the leakage/delay tradeoff as discussed in Section IV-B. However, we find that only complex, and infrequently used cells, such as three-input NORs require appreciable reductions in cell variants which limits the impact on total leakage reduction. Therefore, very good leakage current minimization can be obtained even with libraries with three cell versions for each cell. Also, the restriction that each stack of transistors must use the same V_t and T_{ox} is shown in Table XI to have only a minor impact on leakage. For instance, the uniform stack four-option case shows a 10.6% average power increase compared to the individual stack four-option case; this still represents a nearly $5\times$ reduction in standby leakage compared to the average case. Note that library complexity is not reduced in moving from in-

TABLE X
LEAKAGE CURRENT COMPARISON WITH TRADITIONAL TECHNIQUES (WITH FOUR-OPTION, INDIVIDUAL STACK CONTROL LIBRARY)

	Minimized leakage current (μA)													
	Avg. I_{leak} by random (10K) vectors	State only assignment		V_t-state & proposed heuristic (reduction factor: vs. average leakage current)										
				0% in the best-worst delay range				5% in the best-worst delay range				10% in the best-worst delay range			
				V_t & state		Heuristic 1		V_t & state		Heuristic 1		V_t & state		Heuristic 1	
		I_{leak}	X	I_{leak}	X	I_{leak}	X	I_{leak}	X	I_{leak}	X	I_{leak}	X	I_{leak}	X
C432	24.5	22.7	1.08	13.3	1.8	8.2	3.0	12.5	2.0	7.7	3.2	12.7	1.9	5.5	4.5
C499	65.8	63.9	1.03	41.9	1.6	32.2	2.0	35.7	1.8	26.1	2.5	32.2	2.0	22.7	2.9
C880	50.1	46	1.09	18.9	2.6	10.3	4.9	17.5	2.9	8.5	5.9	16.9	3.0	8.5	5.9
C1355	70.8	67.4	1.05	39.9	1.8	20.4	3.5	33.0	2.1	15.8	4.5	29.8	2.4	9.9	7.1
C1908	56.7	54.8	1.04	27.6	2.1	17.4	3.3	25.8	2.2	14.8	3.8	22.9	2.5	13.2	4.3
C2670	104.7	101.4	1.03	33.3	3.1	14.9	7.0	32.7	3.2	12.3	8.5	31.9	3.3	13.5	7.8
C3540	128.5	121.8	1.05	54.5	2.4	27.7	4.6	51.5	2.5	22.1	5.8	48.5	2.7	18.6	6.9
C5315	221.2	215.1	1.03	81.2	2.7	36.6	6.0	77.1	2.9	30.0	7.4	73.7	3.0	28.4	7.8
C6288	346.8	306.7	1.13	209.3	1.7	153.6	2.3	180.4	1.9	112.2	3.1	153.7	2.3	84.1	4.1
C7552	270.0	262.6	1.03	88.9	3.0	34.9	7.7	86.6	3.1	32.2	8.4	86.1	3.1	30.3	8.9
alu64	260.0	237.2	1.10	90.7	2.9	48.7	5.3	86.1	3.0	43.4	6.0	81.1	3.2	34.3	7.6
AVG			1.06		2.3		4.5		2.5		5.4		2.7		6.2

TABLE XI
LEAKAGE CURRENT COMPARISON BETWEEN CELL LIBRARY OPTIONS

	Minimized leakage current (μA) (reduction factor: vs. average leakage current, runtime unit: sec.)																			
	Avg. I_{leak} by random (10K) vectors	5% in the best-worst delay range																		
		Individual stack control									Uniform stack control									
		4-option			2-option			2-option 3 cell versions only			4-option			2-option			2-option 3 cell versions only			
		I_{leak}	X	time	I_{leak}	X	time	I_{leak}	X	time	I_{leak}	X	time	I_{leak}	X	time	I_{leak}	X	time
C432	24.5	7.7	3.2	2	7.4	3.3	2	7.1	3.4	2	7.3	3.4	4	7.9	3.1	3	8.6	2.8	2
C499	65.8	26.1	2.5	7	26.7	2.5	5	27.8	2.4	5	26.0	2.5	6	28.0	2.3	5	28.9	2.3	6
C880	50.1	8.5	5.9	7	9.7	5.2	6	8.0	6.3	7	10.0	5.0	7	10.7	4.7	6	10.8	4.6	8
C1355	70.8	15.8	4.5	6	16.2	4.4	5	14.1	5.0	7	23.4	3.0	7	25.2	2.8	6	23.9	3.0	7
C1908	56.7	14.8	3.8	4	14.9	3.8	4	14.3	4.0	4	15.9	3.6	4	15.3	3.7	4	16.8	3.4	5
C2670	104.7	12.3	8.5	78	12.1	8.7	75	12.4	8.4	79	16.1	6.5	79	15.4	6.8	76	16.5	6.3	78
C3540	128.5	22.1	5.8	18	24.2	5.3	17	25.3	5.1	18	27.1	4.7	18	25.8	5.0	17	29.2	4.4	19
C5315	221.2	30.0	7.4	213	30.9	7.2	208	30.7	7.2	216	32.1	6.9	205	32.9	6.7	202	33.8	6.6	209
C6288	346.8	112.2	3.1	64	114.2	3.0	58	114.2	3.0	61	134	2.6	65	147.8	2.3	59	145.4	2.4	62
C7552	270.0	32.2	8.4	404	31.4	8.6	397	30.6	8.8	412	31.8	8.5	403	31.1	8.7	402	31.1	8.7	410
alu64	260.0	43.4	6.0	464	44.0	5.9	458	43.2	6.0	470	42.0	6.2	468	47.0	5.5	452	46.1	5.6	468
AVG			5.4			5.3			5.4			4.8			4.7			4.6	

dividual to stack-based control; such a change would be dictated by manufacturing issues as well as the tradeoff between standby power (lower for individual control) and cell area (expected to be slightly lower for stack-based control). In addition, Table XI shows that the runtime differences of heuristics with different libraries are very small.

Finally, Fig. 12 plots the leakage current results for the proposed method and traditional methods as a function of the delay constraint for circuit c6288. Here, a 100% delay point implies a complete replacement of low-V_t and thin-oxide devices with high-V_t and thick-oxide. This is clearly the lowest leakage solution but is also very slow. The key point in Fig. 12 is that the proposed approaches (heuristic 2 results are not shown but are nearly identical to heuristic 1) provide substantial improvement beyond the average leakage or the use of state assignment alone and that these gains are achievable with very small and even zero delay penalties. The rapid saturation of the gains as the delay point increases beyond 10% implies that the new approach is best suited for achieving low-leakage standby states with very little performance overhead (e.g., 5% or even less). Note that the leakage current achieved by our proposed method does not converge to that by state assignment using all high-V_t and thick-oxide devices. The reason is that the selected library cells include only a limited number of thick-oxide assignments in order to simplify the library. Many additional library cells would be needed to achieve convergence to the minimal leakage

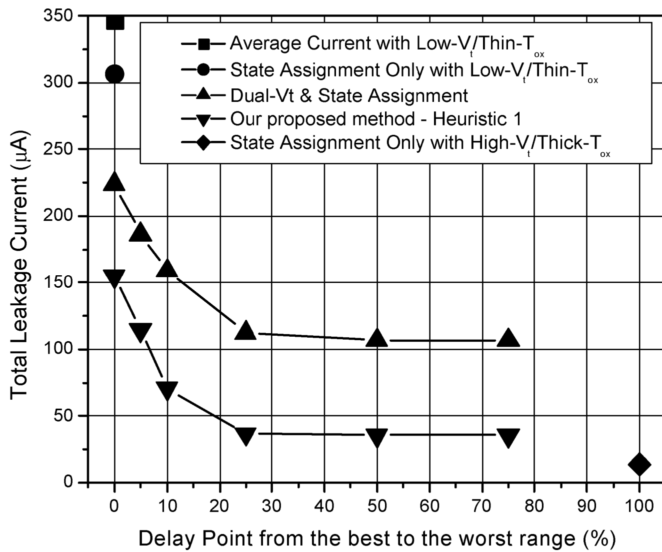

Fig. 12. Leakage current comparison for c6288.

solution; instead the bulk of this leakage savings can be achieved with very little performance penalty.

VI. Conclusion

In this paper, we propose new approaches for standby leakage current minimization under delay constraints. Our approaches use simultaneous state assignment and V_t or $V_t/T_{\rm ox}$ assignment. Efficient methods for computing the simultaneous state and V_t or $V_t/T_{\rm ox}$ assignments leading to the minimum standby mode leakage current were presented. The proposed methods were implemented and tested on a set of synthesized benchmark circuits. Using the new state and V_t assignment technique demonstrates $5\times$ lower leakage than previous V_t-only assignment approaches and $7\times$ lower than state assignment alone (at 5% delay point). In cases where gate leakage is prominent, as in 90-nm CMOS technologies, these improvements are increased by an additional factor of 2 using state and $V_t/T_{\rm ox}$ assignment. This second proposed approach was shown to reduce the total leakage current by more than $5\times$ on average compared to the state assignment only approach (at 5% delay point) and by over $2\times$ compared to the previously presented state and V_t assignment approach. We also investigate the leakage/complexity tradeoff for various cell library configurations and demonstrate that results are still very good even when only two additional variants are used for each cell type.

Acknowledgment

The authors would like to thank H. Deogun, Y. Kim, and B. Zhai for their work and help.

References

[1] 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.

[2] J. Kao, A. Chandrakasan, and D. Antoniadis, "Transistor sizing issues and tool for multithreshold CMOS technology," in *Proc. Design Automation Conf.*, 1997, pp. 409–414.

[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] H. Kawaguchi, K. Nose, and T. Sakurai, "A super cut-off CMOS (SCCMOS) scheme for 0.5 V supply voltage with picoampere standby current," *IEEE J. Solid-State Circuits*, vol. 35, no. 10, pp. 1498–1501, Oct. 2000.

[5] 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.

[6] 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 Electronics Design*, 1998, pp. 239–244.

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

[8] 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*. New York: IEEE Press, 2001.

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

[10] A. Fadi, S. Hassoun, K. A. Sakallaha, and D. Blaauw, "Robust SAT-based search algorithm for leakage power reduction," in *Proc. Int. Workshop on Power and Timing Modeling, Optimization and Simulation*, 2002, pp. 167–177.

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

[12] R. S. Guindi and F. N. Najm, "Design techniques for gate-leakage reduction in CMOS circuits," in *Proc. ISQED*, 2003, pp. 61–65.

[13] F. Hamzaoglu and M. R. Stan, "Circuit-level techniques to control gate leakage for sub-100 nm CMOS," in *Proc. Int. Symp. Low Power Electronics and Design*, 2002, pp. 60–63.

[14] T. Inukai, M. Takamiya, K. Nose, H. Kawaguchi, T. Hiramoto, and T. Sakurai, "Boosted gate MOS (BGMOS): Device/circuit cooperation scheme to achieve leakage-free giga-scale integration," in *Proc. Custom Integrated Circuits Conf.*, 2000, pp. 409–412.

[15] Q. Wang and S. B. K. Vrudhula, "Static power optimization of deep submicron CMOS circuits for dual V_t technology," in *Proc. Int. Conf. Computer-Aided Design*, 1998, pp. 490–496.

[16] 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. Design Automation Conf.*, 1998, pp. 489–494.

[17] 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 Integrat. (VLSI) Systems*, vol. 10, no. 4, pp. 79–90, Apr. 2002.

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

[19] S. Stiffler, "Optimizing performance and power for 130 nm and beyond," in *IBM Technology Group New England Forum*, 2003.

[20] *International Technology Roadmap for Semiconductors*, 2002.

[21] N. Yang, W. K. Henson, and J. J. Wortman, "A comparative study of gate direct tunneling and drain leakage currents in N-MOSFET's with sub-2 nm gate oxides," *IEEE Trans. Electron Devices*, vol. 47, no. 8, pp. 1636–1644, Aug. 2000.

[22] B. Yu, H. Wang, C. Riccobene, Q. Xiang, and M.-R. Lin, "Limits of gate oxide scaling in nano-transistors," in *Proc. Symp. VLSI Tech.*, 2000, pp. 90–91.

[23] Y.-C. Yeo, Q. Lu, W.-C. Lee, T.-J. King, C. Hu, X. Wang, X. Guo, and T. P. Ma, "Direct tunneling gate leakage current in transistors with ultra thin silicon nitride gate dielectric," *IEEE Electron Device Lett.*, vol. 21, no. 11, pp. 540–542, Nov. 2000.

[24] Q. Xiang, J. Jeon, P. Sachdey, B. Yu, K. C. Saraswat, and M.-R. Lin, "Very high performance 40 nm CMOS with ultrathin nitride/oxynitride stack gate dielectric and pre-doped dual poly-Si gate electrodes," in *Proc. Int. Electron Devices Meeting*, 2000, pp. 860–862.

[25] H. Kriplani, F. N. Najm, and I. N. Hajj, "Pattern independent maximum current estimation in power and ground buses of CMOS VLSI circuits: Algorithms, signal correlations, and their resolution," *IEEE Trans. Computer-Aided Design Integrat. Circuits Syst.*, vol. 14, no. 8, pp. 998–1012, Aug. 1995.

[26] R. Puri, private communication.
[27] F. Brglez and H. Fujiwara, "A neutral netlist of 10 combinatorial benchmark circuits," in *Proc. ISCAS*, 1985, pp. 695–698.

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

From May 1996 through June 2001, he was with the Non Volatile Memory Design Team, Samsung Electronics Company, Ltd., Kyungki-Do, Korea. His current 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 in 1986, 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 R&D Engineer in the Advanced Technology Group of Synopsys, Mountain View, CA, he is currently an Associate 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 an NSF 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 the recipient of the 2003 Ruth and Joel Spira Outstanding Teaching Award in 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.