Simultaneous State, Vt and Tox Assignment for Total Standby Power Minimization

Dongwoo Lee, Harmander Deogun, David Blaauw, Dennis Sylvester
{dongwool,hdeogun,blaauw,dmcs}@umich.edu
University of Michigan, Ann Arbor, MI

Abstract

Standby leakage current minimization is a pressing concern for mobile applications that rely on standby modes to extend battery life. Also, gate oxide leakage current (I_{gate}) has become comparable to subthreshold leakage (I_{sub}) in 90nm technologies. In this paper, we propose a new method that uses a combined approach of sleep-state, threshold voltage (V_t) and gate oxide thickness (T_{ox}) assignments in a dual-V_t and dual-T_{ox} process to minimize both I_{sub} and I_{gate}. Using this method, total leakage current can be dramatically reduced since in a known state in standby mode, only certain transistors are responsible for leakage current and need to be considered for high-V_t or thick-T_{ox} assignment. We formulate the optimization problem for simultaneous state, V_t and T_{ox} assignments under delay constraints and propose two practical heuristics. We implemented and tested the proposed methods on a set of synthesized benchmark circuits. Results show an average leakage current reduction of 5-6X and 2-3X compared to previous approaches that only use state or state+V_t assignment, respectively, with small delay penalties.

1 Introduction

The use of low-power standby modes in modern integrated circuits has become a useful tool to improve battery life in wireless applications that may be idle for very long periods of time compared to their active modes. There are various ways to put a circuit into standby mode including the use of dedicated sleep vectors along with special sequential elements [1] and multi-threshold CMOS (MTCMOS) where a very high-V_t transistor (e.g., 0.6V) is inserted in series between ground and a generic combinational circuit to cut off leakage [2]. The latter approach requires careful device layout and sizing to avoid large area and delay penalties and does not scale well into sub-1V technologies. In the former technique of state assignment, the overhead caused by flip-flop modification is minimal [3] but the limitation is that the power reduction can be quite small (on the order of 10-30%) [4]. When the circuit is in standby mode, a larger amount of leakage reduction is desirable since this will translate directly to longer battery life (e.g., standby time for a cell phone).

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 90nm 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 is primarily on circuit designers and EDA tools to deal with this growing component of power consumption. 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 [5][6][7]. 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 [8].

Another traditional approach to leakage reduction that targets only subthreshold leakage is to assign high-V_t transistors to non-critical paths in a circuit [9][10]. 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 90nm node due to the rise of I_{gate} [11]. 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 non-critical paths as well. However, since the state of the circuit is unknown, when entering standby mode all or most of the transistors in a particular gate must be set to high-V_t and thick-oxide to ensure that under all possible standby mode states the total leakage current is acceptable. However, transistors that are simultaneously assigned a high-V_t and a thick-oxide have a substantial 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_{sub} and I_{gate} need to be addressed. If the circuit can be placed in a known standby state, as described earlier, this delay impact may be avoidable if high-V_t and thick-oxide devices are carefully assigned. In particular, a recent approach combining sleep state assignment and V_t assignment was proposed [12], however, this approach does not consider gate leakage current and as such cannot minimize total leakage current in 90nm technologies and beyond.

In this paper, we therefore propose a new method to reduce the total leakage current by simultaneous assignment of standby mode state and 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 since I_{sub} only occurs in transistors that are OFF while significant I_{gate} occurs only in transistors that are ON. Furthermore, depending on the input state of a circuit, only a subset of transistors need to be considered for high-V_t or thick-oxide, thereby significantly reducing the impact on the delay of the gate while obtaining leakage reductions comparable to when all transistors are assigned high-V_t and thick-oxides. The proposed approach thus provides a much better trade-off between leakage and performance compared to V_t and oxide thickness assignment with an unknown/arbitrary input state. The proposed method is compatible with existing library-based design flows, and we explore different trade-offs between the number of V_t and T_{ox} variations for each library cell and the obtained leakage reduction. Also, we compare the obtained leakage reduction when V_t and T_{ox} 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 and the V_t / T_{ox} assignments interact, it is necessary to consider their optimization simultaneously. The state, V_t, and T_{ox} assignment task is to find a simultaneous assignment that minimizes the total 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 T_{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 a predictive 65nm technology. On average, the proposed method improved leakage current by a factor of 5-6X over an all low-V_t and thin-oxide design solution with a 5% delay penalty and achieves more than a 2X improvement over an approach using V_t and state assignment only (i.e., without dual-T_{ox}).

Proceedings of the Design, Automation and Test in Europe Conference and Exhibition (DATE'04)
1530-1591/04 $20.00 © 2004 IEEE

2 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 BSIM4 models and accurately represent both subthreshold and gate leakage components. The device simulation parameters were obtained using leakage estimates from expected 65nm processes [13], and had a gate leakage component that was approximately 36% of the total leakage at room temperature (at which all analysis is performed).[1] Different T_{ox} and V_t versions as well as high- and low-V_t versions of a cell, as will be explained further in Section 4, were characterized. Also, the delay and output slope as a function of cell input slope and output loading were stored in precharacterized tables. The difference in I_{gate} for the thick-oxide NMOS devices vs. the thin-oxide device is 11X whereas I_{sub} is reduced by 17.8X (16.7X) when replacing a low-V_t NMOS (PMOS) device with a high-V_t version.

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 Figure 1. The maximum gate tunneling current occurs when the input is at V_{dd} and $V_s = V_d = 0V$ for the NMOS device. In this case, $V_{gs} = V_{gd} = V_{dd}$ and the I_{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 [14]. 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_{gate} for a PMOS device is typically one order of magnitude smaller than that for an NMOS device with identical T_{ox} and V_{dd} [11][15]. 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, in use today in a few processes, PMOS I_{gate} can actually exceed NMOS I_{gate} for higher nitrogen concentrations [16]. 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.

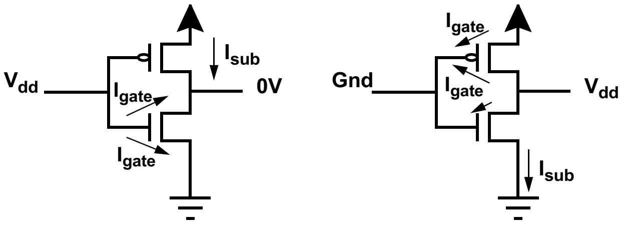

Figure 1. Inverter circuit with NMOS oxide leakage current.

3 Leakage Reduction Approach

The proposed leakage optimization method performs simultaneous assignment of standby mode state and 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 trade-off.

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. 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 non-conducting 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 2-input NAND and NOR gate in Figure 2. In Figure 2(a) a 2-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

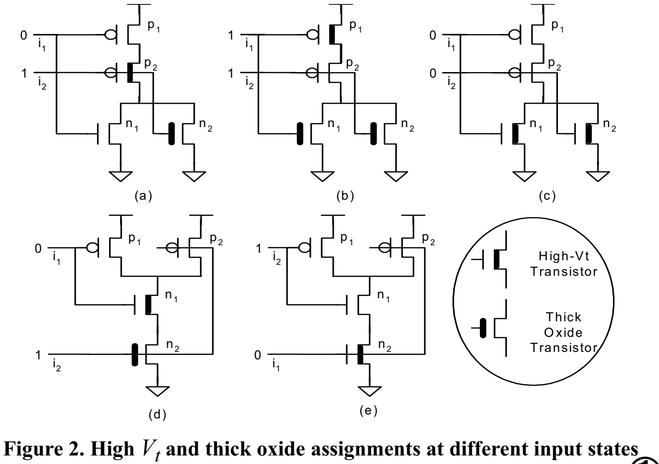

Figure 2. High V_t and thick oxide assignments at different input states

[1] Since this work aims at standby mode leakage, we expect junction temperatures during these idle periods to be lower than under normal operating conditions, making room temperature analysis more valid.

is unaffected by the high-V_t and thick-oxide assignments, while the other transitions are affected only moderately.

In Figure 2(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 Figure 2(a), or the state 00, shown in Figure 2(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-oxides without degrading the 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 [5]. This is illustrated in Figure 2(d), for a 2-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 Figure 2(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.

4 Cell Library Construction

In order to perform simultaneous V_t, T_{ox} and state assignment, it is necessary to develop a library where for each cell the necessary V_t and T_{ox} version are available. 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 is possible, providing different delay / leakage trade-off points. For the fastest and highest leakage trade-off point, all transistors are assigned to low-V_t and thin oxides, such as the NAND2 gate shown in Figure 3(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 Figure 3(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 trade-off point.

In addition to the fastest version and minimum leakage version of the cell, a number of other intermediate trade-off 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 trade-off 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 4 different trade-off points for each input state of a library cell, which are: 1) the minimum delay, shown in Figure 3(a), 2) minimum leakage, shown in Figure 3(b), 3) fast falling transition but slow rising transition, with intermediate leakage, shown in Figure 3(c), and 4) fast rising transition but slow falling transition with intermediate leakage, shown in Figure 3(d). Although other possible trade-off points could be considered, we 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 trade-off points for each input combination could result in as many as 16 (4x4) cell versions for a 2 input gate. However, in practice, many of the cell versions are shared between different input states. Also, in some cases not all 4 trade-off 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 Figure 3(a) and is shared for all input combinations, and the minimum leakage version is shown in Figure 3(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 trade-off 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 Figure 3(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 trade-off 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 trade-off point, thereby resulting in no additional cell version. The NAND2 gate therefore requires a total of 5 cell versions to provide up to 4 trade-off points for each input state. In Table 1, we show the delay / leakage trade-offs 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 2, shows the number of cell version required for sev-

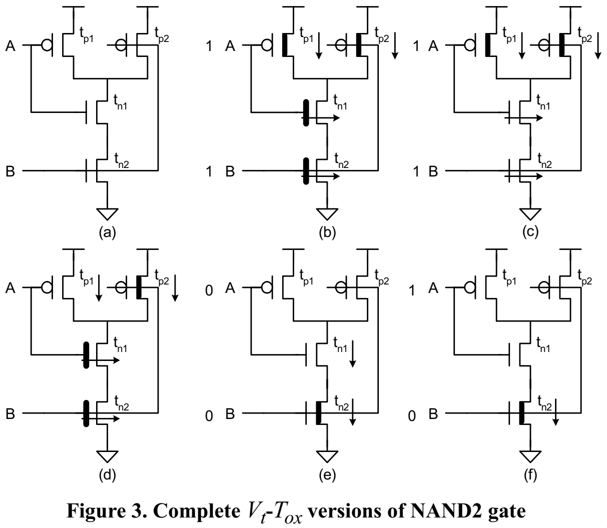

Figure 3. Complete V_t-T_{ox} versions of NAND2 gate

Table 1. Trade-offs for different V_t-T_{ox} versions of NAND2 gate

State	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	1.00	1.00	1.00	1.00
	Fast rise delay (d)	109.1	1.00	1.36	1.27	1.27
	Fast fall delay (c)	91.4	1.36	1.36	1.00	1.00
	Minimum leakage (b)	19.5	1.36	1.37	1.27	1.27
00	Minimum delay (a)	41.2	1.00	1.00	1.00	1.00
	Minimum leakage (e)	14.0	1.00	1.00	1.12	1.16
10	Minimum delay (a)	91.8	1.00	1.00	1.00	1.00
	Minimum leakage (f)	13.3	1.00	1.00	1.12	1.16

eral common gates. Note that the number of cell version is higher for NOR gates than NAND gates. Since for a library the total number of cells would increase significantly, we also explored reducing the number of cells by allowing only two trade-off points for each cell (minimum delay, and minimum leakage), instead of 4 trade-off points. In this case, the number of cells for the NAND2 gate reduces to only 3 versions. The number of cell version required for two trade-off points for different cell types is shown in Table 2, column 3. In Section 6 we compare the final leakage results using the full library with 4 trade-off points and the reduced library with only two trade-off points.

Finally, we assume the ability to assign V_t and T_{ox} on an individual basis within stacks of transistors. Although it is generally possible to assign the V_t or T_{ox} 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 [17]. Hence, at times it may be desirable to restrict the assignment of V_t and T_{ox} such that all transistors in a stack are uniform. In this case, less flexibility exists in the assignment of V_t and T_{ox}, and hence the obtained trade-off in delay and leakage will degrade to some extent. First, it is important to note that due to the use of pin-swapping, the assignment of T_{ox} to transistors in a stack is already uniform in the proposed approach. This is evident from the 5 added cell versions for the NAND2 in Figure 3, 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_{ox} assignments are expected to be more severe that those for spacing between different V_t assignments [17]. However, the V_t-assignment is not always uniform as shown in Figure 3(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 trade-off. In Section 6, we present results showing the impact on the leakage optimization when uniform stack assignments are enforced in the library.

5 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_{ox} assignments for a circuit under delay constraints. As mentioned, the leakage minimization problem can be formulated as a 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 4, for each input state assignment, there are up to four possible V_t-T_{ox} assignments for each gate. Note that while the total number of cell versions can be larger than 4, only 4 of them need to be considered for each specific input state. For instance, for the nand2 gate in Figure 3, only versions (a)-(d) are considered for a 11 input state. Therefore, the total number of possible V_t-T_{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}.

Table 2. The number of needed library cells

	4 trade-off points	2 trade-off points
Inverter	5	3
NAND2	5	3
NAND3	5	3
NOR2	8	4
NOR3	9	5

In order to find an exact solution to the problem, we developed a branch-and-bound method with efficient pruning of the search space. The approach is similar to that presented in [12], which was limited to V_t and state assignment only. The branch and bound algorithm for V_t-T_{ox} and state assignment uses two interdependent search trees. The first and primary search tree is the *state tree* which is searched to determine the input state of the circuit. The nodes of the state tree correspond to the input variables of the inputs of the circuit. Each node has two edges corresponding to high and low state assignment of the associated variables. Each node of the state tree is associated with a second search tree, the *gate tree*, which is searched to determine the V_t-T_{ox} assignment of the circuit. Hence, for a state tree with k nodes, there implicitly exist k copies of the gate tree, as shown in Figure 4. Each node in a particular gate tree corresponds to a gate in the circuit. Since there are four possible V_t-T_{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. These four edges are sorted by their leakage current before the search. During a downward traversal of the gate tree, the edges associated with a gate are selected in order of increasing leakage current, subject to the delay constraint being met. This greedy approach attempts to select the lowest possible leakage assignment for each gate as they are encountered in the search tree and results in the establishment of a good lower bound to during the first downward traversal through the tree. This lower bound is then used for pruning the search tree in subsequent traversals and improves the search efficiency.

To improve the runtime, we exploit a number of methods, including incremental computation of the delay and leakage bounds as the search traverses through the gate tree. Also, bounds on the leakage with partial input state information are computed during the traversal of the state tree and are used to order the state tree branches.

The exponential nature of the problem makes it impossible to obtain an exact solution for substantial circuits. Therefore, we also propose two heuristics. In the first heuristics (heuristic 1), the gate and state tree search is limited to only a single traversal. It was found that a single downward traversal of the gate tree tends to produce a high quality leakage solution because the gate tree is searched in a pre-sorted order. In the second heuristic (heuristic 2), a single downward traversal of the state tree is still performed. However, the state tree is search for a preset time limit to further improve the result over that of heuristic 1.

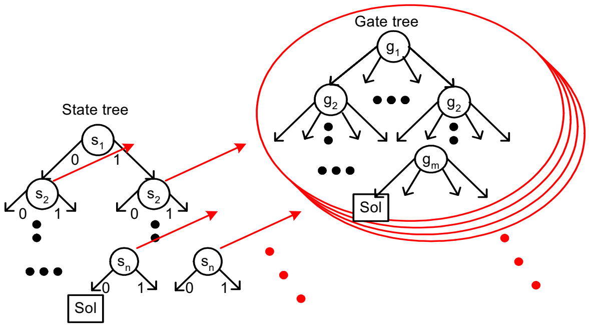

Figure 4. State tree with gate tree at each node

Table 3. Leakage current comparison between heuristics with 4-option library (current unit: μA)

	Average I_{leak} by random (10K) vectors	5% delay penalty					10% delay penalty					25% delay penalty				
		Heu1			Heu2		Heu1			Heu2		Heu1			Heu2	
		I_{leak}	X	Time	I_{leak}	X	I_{leak}	X	Time	I_{leak}	X	I_{leak}	X	Time	I_{leak}	X
c432	24.5	6.9	3.6	2	3.8	6.5	4.8	5.1	2	2.8	8.7	2.7	9.2	2	2.5	9.8
c499	65.8	24.8	2.7	6	23.4	2.8	19.7	3.3	6	19.3	3.4	7.5	8.8	4	7.1	9.3
c880	50.1	8.7	5.7	7	7.7	6.5	8.3	6.0	7	7.0	7.2	7.0	7.1	6	7.0	7.1
c1355	70.8	15.4	4.6	6	13.1	5.4	12.6	5.6	6	10.0	7.1	7.6	9.3	5	7.5	9.4
c1908	56.7	14.7	3.9	4	13.5	4.2	12.1	4.7	4	10.5	5.4	6.2	9.2	3	6.2	9.2
c2670	104.7	14.7	7.1	75	12.3	8.5	11.4	9.2	75	11.3	9.3	11.3	9.2	74	11.1	9.4
c3540	128.5	21.6	6.0	17	19.9	6.5	19.1	6.7	17	17.3	7.4	13.7	9.4	16	13.6	9.5
c5315	221.2	31.1	7.1	200	30.5	7.3	28.5	7.8	198	27.6	8.0	24.1	9.2	196	24.0	9.2
c6288	346.8	114.7	3.0	63	107.5	3.2	70.9	4.9	57	63.6	5.5	36.8	9.4	53	35.7	9.7
c7552	270.0	32.6	8.3	393	31.3	8.6	30.4	8.9	390	30.1	9.0	28.3	9.5	388	28.3	9.6
alu64	260.0	42.2	6.2	455	40.4	6.4	35.5	7.3	454	33.7	7.7	28.0	9.3	442	27.1	9.6
AVG			5.3			6.0		6.3			7.2		9.1			9.3

6 Results

The proposed methods for simultaneous state, V_t and T_{ox} assignments were implemented on a number of benchmark circuits [18] synthesized using an industrial cell library. A comparison of our first and second heuristics along with an average leakage computed using 10000 random input vectors is shown in Table 3. The total leakage current value is given in μA and runtime is given in CPU seconds. In heuristic 2, we set the runtime limit as 1800 seconds (30 minutes). 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_{ox} techniques were not employed. Furthermore, the delay penalties used in all results are defined by a percentage of the *maximum possible* delay penalty 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 nearly double the total circuit delay. Thus, when interpreting the results in this section, a 5% delay penalty means that the circuit after V_t and T_{ox} assignment has a delay that is also approximately 5% larger than the original fastest implementation.

As shown in Table 3, heuristic 2 generally provides somewhat better results but at much greater runtimes. The runtime overhead for heuristic 2 can be over 800X in the extreme (c432 at 25% delay penalty) but in the larger circuits is more often in the range of 5-10X. On average, heuristic 2 provides 7.3% lower leakage current than heuristic 1 across these benchmarks at the 5% delay penalty. The improvement of the two proposed heuristics compared to the average leakage without state, V_t or T_{ox} assignment is dramatic and approaches an order of magnitude if a 25% delay penalty is tolerable. More reasonably, with just a 5% delay penalty, the reduction in total standby leakage is 5.3-6X with a maximum improvement of 8.5X for heuristic 2 in circuit c2670.

In Table 4, we compare our results to traditional standby mode techniques, including state assignment alone and simultaneous state and V_t assignment. The total leakage current value is given in μA. Again, we report the reduction factor in relation to the average leakage current with 10000 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%. By adding V_t assignment, the algorithm of [12] shows an average reduction of 57.6% beyond state assignment alone with a 5% delay penalty. This number increases to 67.3% when a delay penalty of 25% is allowed. The approach outlined in this paper provides an additional 52.6% (65.5%) reduction in current *beyond* state and V_t assignment for the 5% (25%) delay penalty.

Table 5 provides a comparison of results using the various cell library options; 4 and 2 trade-off points with individual stack control, and also with uniform stacks. The main result in Table 5 is that there is very little leakage current penalty when moving from a full 4-option library to a smaller 2-option library. There are several cases where the smaller library outperforms the larger library due to the

Table 4. Leakage current comparison with 4-option library (current unit: μA)

	Number of		Average I_{leak} by random (10K) vectors	State Assignment Only		5% delay penalty				10% delay penalty				25% delay penalty			
						Vt & State		Heu1		Vt & State		Heu1		Vt & State		Heu1	
	Inputs	Gates		I_{leak}	X	I_{leak}	X	I_{leak}	X	I_{leak}	X	I_{leak}	X	I_{leak}	X	I_{leak}	X
c432	36	177	24.5	22.7	1.08	12.4	2.0	6.9	3.6	10.4	2.4	4.8	5.1	8.2	3.0	2.7	9.2
c499	41	519	65.8	63.9	1.03	37.0	1.8	24.8	2.7	33.3	2.0	19.7	3.3	23.8	2.8	7.5	8.8
c880	60	364	50.1	46.0	1.09	17.8	2.8	8.7	5.7	17.1	2.9	8.3	6.0	16.2	3.1	7.0	7.1
c1355	41	528	70.8	67.4	1.05	33.6	2.1	15.4	4.6	30.5	2.3	12.6	5.6	23.9	3.0	7.6	9.3
c1908	33	432	56.7	54.8	1.04	26.6	2.1	14.7	3.9	23.4	2.4	12.1	4.7	18.2	3.1	6.2	9.2
c2670	233	825	104.7	101.4	1.03	32.7	3.2	14.7	7.1	32.0	3.3	11.4	9.2	30.0	3.5	11.3	9.2
c3540	50	940	128.5	121.8	1.05	50.3	2.6	21.6	6.0	47.8	2.7	19.1	6.7	40.3	3.2	13.7	9.4
c5315	178	1627	221.2	215.1	1.03	77.6	2.9	31.1	7.1	74.6	3.0	28.5	7.8	70.6	3.1	24.1	9.2
c6288	32	2470	346.8	306.7	1.13	186.3	1.9	114.7	3.0	159.0	2.2	70.9	4.9	112.5	3.1	36.8	9.4
c7552	207	1994	270.0	262.6	1.03	86.5	3.1	32.6	8.3	86.0	3.1	30.4	8.9	84.2	3.2	28.3	9.5
alu64	131	1803	260.0	237.2	1.10	90.7	2.9	42.2	6.2	82.7	3.1	35.5	7.3	75.3	3.5	28.0	9.3
AVG					1.06		2.5		5.3		2.7		6.3		3.1		9.1

Table 5. Leakage current comparison between cell library options (current unit: μA)

	Average I_{leak} by random (10K) vectors	5% delay penalty							
		4-option		2-option		4-option uniform stack		2-option uniform stack	
		I_{leak}	X	I_{leak}	X	I_{leak}	X	I_{leak}	X
c432	24.5	6.9	3.6	7.5	3.3	6.7	3.7	7.8	3.1
c499	65.8	24.8	2.7	27.6	2.4	26.2	2.5	28.6	2.3
c880	50.1	8.7	5.7	9.0	5.6	9.4	5.3	10.3	4.8
c1355	70.8	15.4	4.6	17.0	4.2	22.4	3.2	23.8	3.0
c1908	56.7	14.7	3.9	15.2	3.7	15.2	3.7	15.8	3.6
c2670	104.7	14.7	7.1	12.2	8.6	16.2	6.5	14.8	7.1
c3540	128.5	21.6	6.0	23.9	5.4	25.2	5.1	24.7	5.2
c5315	221.2	31.1	7.1	30.7	7.2	32.1	6.9	33.0	6.7
c6288	346.8	114.7	3.0	120.6	2.9	134.0	2.6	149.6	2.3
c7552	270.0	32.6	8.3	31.2	8.7	32.0	8.4	30.6	8.8
alu64	260.0	42.2	6.2	42.3	6.2	42.8	6.1	46.9	5.5
AVG			5.28		5.27		4.91		4.77

heuristic nature of the algorithm used (heuristic 1 is used in this table). Since the library size required in the 2-option scenario is roughly half that of 4-option, we conclude that the use of 2-option represents a very good trade-off between library complexity and potential leakage reduction. Other library simplifications could be studied to further explore this trade-off. Also, the restriction that each stack of transistors must use the same V_t is shown in Table 5 to have only a minor impact on leakage. For instance, the stack 4-option case shows a 10.6% average power increase compared to the 4-option case; this still represents a nearly 5X reduction in standby leakage compared to the average case. Note that library complexity is not reduced in moving from individual to stack-based control; such a change would be dictated by manufacturing issues as well as the trade-off between standby power (lower for individual control) and cell area (expected to be slightly lower for stack-based control).

Finally, Figure 5 plots the leakage current results for the proposed method and traditional methods as a function of the delay constraint for circuit c7552. Here, a 100% delay penalty implies a complete replacement of low-V_t and thin-oxide devices with high-V_t and thick-oxide. This is the lowest leakage solution but is also very slow. The key point in Figure 5 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 penalty 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)

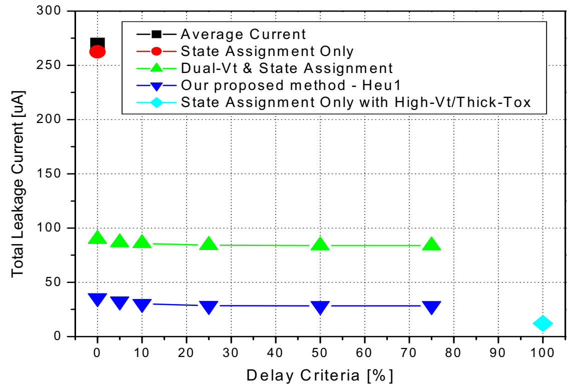

Figure 5. Leakage current comparison for c7552

7 Conclusions

In this paper, we propose a new approach for total leakage current minimization including I_{gate} as well as I_{sub}, under delay constraints. Our approach uses simultaneous state, V_t and T_{ox} assignments. An efficient method for computing the simultaneous state, V_t and T_{ox} assignments leading to the minimum total leakage current was presented. The proposed methods were implemented and tested on a set of synthesized benchmark circuits. The proposed approach was shown to reduce the total leakage current by more than 5X on average compared to the state assignment only approach (with just a 5% delay penalty) and by over 2X compared to a previous state and V_t assignment approach.

Acknowledgements

The work has been supported by NSF, SRC, GSRC/DARPA, IBM, and Intel.

References

[1] J. Halter and F. Najm, "A gate-level leakage power reduction method for ultra-low-power CMOS circuits," Proc. CICC,1997.

[2] S. Mutoh, *et al.*, "1-V power supply high-speed digital circuit technology with multithreshold-voltage CMOS," IEEE JSSC, vol.30, pp.847-854, Aug. 1995.

[3] V. De, *et al.*,"Techniques for leakage power reduction," in *Design of high-performance microprocessor circuits*, IEEE press, 2001

[4] M.C. Johnson, *et al.*, "Models and algorithms for bounds on leakage in CMOS circuits," IEEE Trans. CAD, June 1999.

[5] D. Lee, *et al.*, "Analysis and minimization techniques for total leakage considering gate oxide leakage," Proc. DAC, 2003.

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

[7] F. Hamzaoglu *et al.*, "Circuit-level techniques to control gate leakage for sub-100nm CMOS," Proc. ISLPED, pp.60-63, 2002.

[8] T. Inukai, *et al.*, "Boosted gate MOS (BGMOS): device/circuit cooperation scheme to achieve leakage-free Giga-scale integration," Proc. CICC, pp. 409-412, 2000.

[9] S. Sirichotiyakul, *et al.*, "Duet: an accurate leakage estimation and optimization tool for dual Vt circuits," IEEE Trans. VLSI, pp. 79-90, April 2002.

[10] M.Ketkar, *et al.*, "Standby power optimization via transistor sizing and dual threshold voltage assignment," Proc.ICCAD, 2002.

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

[12] D. Lee, *et al.*, "Static leakage reduction through simultaneous threshold voltage and state assignment," Proc. DAC, 2003.

[13] International Technology Roadmap for Semiconductors, 2002.

[14] N. Yang, *et al.*, "A comparative study of gate direct tunneling and drain leakage currents in N-MOSFETs with sub-2nm gate oxides," IEEE Trans. Electron Devices, Aug.2000.

[15] B. Yu, *et al.*, "Limits of gate oxide scaling in nano-transistors," Proc. Symp. VLSI Tech., pp. 90-91, 2000.

[16] Y.-C. Yeo, *et al.*, "Direct tunneling gate leakage current in transistors with ultra thin silicon nitride gate dielectric," IEEE Electron Device Letters, pp. 540-542, Nov. 2000.

[17] Ruchir Puri, IBM T.J. Watson Research, personal communication.

[18] F. Brglez and H. Fujiwara, "A Neutral netlist of 10 combinational benchmark circuits", Proc. ISCAS, 1985, pp.695-698