Transistor and Pin Reordering for Gate Oxide Leakage Reduction in Dual T_{ox} Circuits *

Anup Kumar Sultania†, Dennis Sylvester‡, and Sachin S. Sapatnekar†
† Department of ECE, University of Minnesota, Minneapolis, MN 55455.
‡ Department of EECS, University of Michigan, Ann Arbor, MI 48109.

Abstract

Gate oxide tunneling current (I_{gate}) is emerging as a key roadblock for device scaling in nanometer-scale CMOS circuits. A practical means to reduce I_{gate} is to leverage dual T_{ox} processes where non-critical transistors are assigned a thicker T_{ox}. In this paper, we generate a leakage/delay tradeoff curve for dual T_{ox} circuits, and propose a transistor and pin reordering technique that has a minimal layout impact to further reduce the total leakage current up to 18% and I_{gate} up to 26% without incurring any delay penalty.

1 Introduction

While aggressive downscaling of gate oxides is essential to improve the current drive of next-generation MOS transistors, quantum effects are seen to play a large role under these ultra-thin oxide devices. In 90nm and 65 nm technologies, the gate oxide leakage current (I_{gate}) due to such effects is comparable to subthreshold leakage. This new source of leakage is particularly important as low power has become a major design parameter, especially in digital CMOS circuits aimed at portable applications.

A principal source of I_{gate} arises from the tunneling of electrons through the gate oxide. The probability of electron tunneling is a strong function of the applied electric field and the barrier thickness itself, which is simply T_{ox}, with a small change in T_{ox} having a tremendous impact on I_{gate}. For example, in MOS devices with SiO_2 gate oxides, a difference in T_{ox} of only 2Å can result in an order of magnitude increase in I_{gate} [1], so that reducing T_{ox} from 18Å to 12Å increases I_{gate} by approximately 1000×[1]. The most effective way to control I_{gate} is through the use of new materials, namely, high-k dielectrics, but such materials are not expected to be manufacturable until approximately 2007 at the earliest.

The issue of power dissipation due to gate leakage arises in two contexts. In the *stand-by* mode, when a circuit is not undergoing any active operations, leakage may be controlled through various means, prominent among which are the use of multiple threshold CMOS (MTCMOS) sleep transistors [3], the assignment of circuit inputs to send the circuit into a low leakage state [4], and body biasing [5]. In the *active* mode, i.e., in normal operation, clearly, the use of neither sleep transistors nor state assignment is viable. Although recent studies show that at the 90nm mode, leakage can contribute over 40% of the total power [6], there are few effective techniques that have been studied in the community. Therefore, it is critical to develop methods that will effectively reduce I_{gate} in the active mode. To this end, in earlier work [7], we had described a dual T_{ox} ($T_{ox_{Lo}}$ and $T_{ox_{Hi}}$) assignment strategy to generate leakage/delay tradeoff curves. Starting with all transistors assigned to $T_{ox_{Hi}}$, [7] iteratively looked at critical path transistors and selected a transistor for $T_{ox_{Lo}}$ assignment based on a cost function.

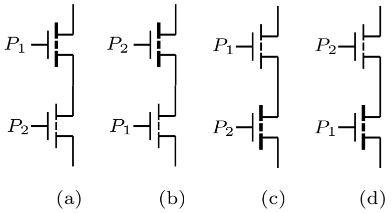


Figure 1: All possible configurations using pin and transistor reordering for two NMOS transistors in a series. The transistor gates with thick dotted lines correspond to a $T_{ox_{Hi}}$ assignment, while those with thin dotted line correspond to $T_{ox_{Lo}}$ assignment.

This paper adds another degree of flexibility to controlling leakage, as it addresses the optimization of dual T_{ox} circuits for reducing leakage power using *transistor and pin reordering.* A major advantage of this optimization is that it has a low layout impact, and is therefore a "cheap" optimization in terms of its impact on the design methodology. Our work will explore how I_{gate} varies when the order of pins and transistors in a stack is varied, and develop an algorithm that finds an optimal reordering. Under a specified delay constraint, the best configuration for each gate is chosen such that it results in the maximum leakage reduction without increasing circuit delay.

Although our motivation so far has primarily addressed gate leakage due to its growing dominance, it is important to note that the true optimization objective is the *total* leakage, which consists of the gate leakage and the subthreshold leakage. It is essential to optimize these in an integrated manner, and our work does exactly this.

In the literature, several research works [8–10] pertaining to transistor reordering techniques have been reported. These approaches aim at reducing the dynamic power dissipation due to the switching activity of transistors, rather than reducing the leakage power dissipation in the active mode. In [11], the authors apply two different pin reordering techniques: one attempts to minimize standby I_{gate}, while the other reduces runtime leakage. In both approaches the effect of this transformation on circuit delay is not considered. Furthermore, pin reordering without transistor re-

*This work was supported in part by the NSF under award CCR-0205227 and the SRC under contract 2003-TJ-1092.

[1] The fundamental limit of gate oxide thickness scaling is projected to be about 8Å [2].

ordering limits the search space in dual T_{ox} circuits. To illustrate this, consider two NMOS transistors connected in series, as shown in Figure 1. Applying pin reordering leads to only two possible cases ((a) and (b)) whereas if transistor reordering is also allowed, the number of cases double as the search space now also includes the configurations in cases (c) and (d).

2 Leakage Models

In this section, we describe the models used to calculate I_{sub} and I_{gate} for each transistor, and the approach for computing the average I_{sub} and I_{gate} values for a given logic gate. The total leakage current for a logic gate is then computed as the sum of its average I_{sub} and I_{gate}. We also present a delay model that considers interconnect delay.

2.1 Subthreshold Leakage Model

It is well known that I_{sub} is exponentially dependent on threshold voltage (V_{th}). As described in [7], it is fairly complex to obtain an analytical model for V_{th}. For convenience, we use a simple look-up table (LUT) to determine I_{sub}. For dual T_{ox} circuits such an LUT could be extremely large: for a k-input NAND gate, for instance, we would store the leakage current for each of the 2^k possible T_{ox} assignments, and each T_{ox} assignment would require entries for the $2^k - 1$ leakage states corresponding to different input logic values[2], resulting in a total of $2^k \cdot (2^k - 1)$ entries. The LUT size can be reduced significantly using the following concepts:

Dominant input states: It has been shown [12] that I_{sub} can be accurately captured by using a set of dominant states, corresponding to the cases where only one transistor on each path to a supply node is on.

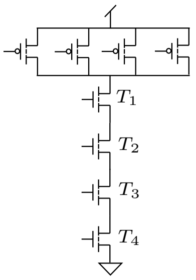
Weak T_{ox} dependencies: In a dominant state, for a given T_{ox} choice for the leaking transistor, the subthreshold leakage is only weakly dependent on the T_{ox} values of other transistors. Intuitively, this relates to the fact that the leaking transistor is the largest resistance on the path. We have validated this through SPICE simulations, and the results for a 4-input NAND gate are shown in Figure 2. When T_4 is the leaking transistor and is set to $T_{ox_{Lo}}$, it can be seen that I_{sub} has a range of only about 1% over all possible assignments for the other inputs. Similar results are seen for other logic gates over various T_{ox} assignments.

For a k-input NAND gate, there are k dominant states. The weak T_{ox} dependencies require that for each of these states, two I_{sub} numbers must be maintained: one at $T_{ox_{Hi}}$ and the other at $T_{ox_{Lo}}$. As a result, the LUT size reduces to $2k$. For a logic gate with k-parallel transistors (such as the pull-up in a k-input NAND, or a pull-down in a k-input NOR), two entries ($T_{ox_{Hi}}$ and $T_{ox_{Lo}}$) are sufficient as the value of I_{sub} per unit $\frac{w}{l}$ for each parallel branch is almost equal.

The average subthreshold leakage ($I_{sub_{avg}}$) for a logic gate under a given T_{ox} assignment may therefore be calculated as follows:

$$I_{sub,avg.} = \sum\nolimits_{i \,\in\, \text{dominant input states}} P_i \times I_{sub_i} \qquad (1)$$

where P_i is the probability of occurrence of state i, and I_{sub_i} is the subthreshold leakage current in that state.



T_{ox}				I_{sub}
T_1	T_2	T_3	T_4	(nA)
Lo	*Lo*	*Lo*	*Lo*	34.70
Lo	*Lo*	*Hi*	*Lo*	34.83
Lo	*Hi*	*Lo*	*Lo*	34.85
Lo	*Hi*	*Hi*	*Lo*	34.99
Hi	*Lo*	*Lo*	*Lo*	34.78
Hi	*Lo*	*Hi*	*Lo*	34.92
Hi	*Hi*	*Lo*	*Lo*	34.93
Hi	*Hi*	*Hi*	*Lo*	35.08

Figure 2: The variation of I_{sub} through the pull-down chain for the dominant state when only T_4 is off. Here, $T_{ox_{Lo}} = 12$Å(Lo), $T_{ox_{Hi}} = 17$Å(Hi), and T_4 is at $T_{ox_{Lo}}$.

2.2 Gate Oxide Tunneling Model

Gate oxide leakage can be primarily attributed to electron [hole] tunneling in NMOS [PMOS] devices. Physically, this tunneling occurs in the gate-to-channel region, as well as in the gate-to-drain/source overlap regions. The latter type of tunneling, referred to as edge direct tunneling (EDT), is ignored in our case for two reasons: first, the gate-to-drain/source overlap region is significantly smaller than the channel region, and second, the oxide thickness in this overlap region can be increased after gate patterning to further suppress EDT.

Our work focuses on gate-to-channel tunneling and we use the following analytic tunneling current density (J_{tunnel}) model based on the electron [hole] tunneling probability through a barrier height (E_B) [13].

$$\begin{aligned} J_{tunnel} \quad &= \quad \frac{4\pi m^* q}{h^3}(kT)^2(1+\frac{\gamma kT}{2\sqrt{E_B}}) \times \\ &\qquad \exp(\frac{E_{F0,Si/SiO_2}}{kT})\exp(-\gamma\sqrt{E_B}) \qquad (2) \end{aligned}$$

where $E_{F0,Si/SiO_2}$ is the Fermi level at the Si/SiO$_2$ interface and m^* is $0.19M_o$ for electron tunneling and $0.55M_o$ for hole tunneling, where M_o is the electron rest mass. The terms k, h and q correspond to physical constants (respectively, Boltzmann's constant, Planck's constant and the charge on an electron), $\gamma = 4\pi T_{ox}\sqrt{2M_{ox}}/h$ where M_{ox} is the effective electron [hole] mass in the oxide, T is the operating temperature, and E_B is the barrier height.

It was shown in [11] that, like I_{sub}, I_{gate} also exhibits state dependency. When the gate node of an NMOS [PMOS] transistor is at logic 0 (logic 1), the only tunneling component arises from EDT which is neglected in our work. Therefore, we will only consider the cases where the gate node is at logic 1 for NMOS and at logic 0 for PMOS. For example, while determining I_{gate} for transistor T_2 in the 4-input NAND gate in Figure 3, it can be shown that the maximum leakage for T_2 occurs at the input state[3] $(x, 1, 1, 1)$, and that the I_{gate} values for the states $(1, 1, 0, x)$, $(0, 1, 0, x)$ and $(x, 1, 1, 0)$ can be ignored. For further details, the reader is referred to [11].

In general, this may be restated as follows: the dominant state for I_{gate} for a particular transistor in a stack corresponds to the case when all of the transistors below (above) it in the NMOS (PMOS) stack are on. The average I_{gate} for a logic gate can then be calculated as:

$$I_{gate,avg.} = \sum\nolimits_{\text{transistor } t \,\in\, \text{logic gate}} P_t \times I_{gate_t} \qquad (3)$$

[2] The only input assignment with no leakage due to NMOS is the case when all transistors in the pull-down chain are on.

[3] "State" = logic values at the inputs to (T_1, T_2, T_3, T_4).

Here, P_t for NMOS [PMOS] transistors connected in parallel, as in a NOR [NAND] gate, is the probability that the input is at logic 1 [0]. For a stack of NMOS [PMOS] transistors in series in a NAND [NOR] gate, P_t for a transistor is the product of the probabilities that each of the transistors below [above] it has an input of logic 1 [0]. The value of I_{gate} is computed using Equation (2) for the specified L_{eff} and width of the transistor under consideration.

Observe that the use of dominant states for the computation of I_{gate} and I_{sub}, and keeping the $T_{ox_{Lo}}$ and $T_{ox_{Hi}}$ values reasonable spaced apart, rules out the complex interaction between these two components [11].

2.3 Delay Model

We use an LUT-based approach for delay computation. For each input of a logic gate, rise and fall delay values are determined through SPICE simulations over a range of output loads under a single-input switching model. A linear fit is carried out on these data to obtain the slope (delay/load) and intercept (delay at zero load) values. The LUT stores these two numbers for each input, along with gate input capacitance for each logic gate. Different combinations of T_{ox} in a stack of transistors will result in different input-to-output delays for the same input; for example, for a k-input NAND gate, 2^k entries would be required to compute the fall delay from each input to the output, for a total of $k \cdot 2^k$ entries in the LUT. This LUT size may be greatly reduced for a small loss in accuracy.

For the output fall transition, for each input-to-output delay, we create two LUTs, corresponding to a gate oxide thickness assignment of $T_{ox_{Lo}}$ and $T_{ox_{Hi}}$, respectively; similarly, two LUTs are constructed for the rise transition. In each LUT, we observe that the delay depends strongly on the *number* of transistors in the chain that are at $T_{ox_{Lo}}$ or $T_{ox_{Hi}}$, and very weakly on their position. We fit a simple formula as in [7], and in most cases, the error was under 2%, with a worst-case error of 3%.

3 Transistor and Pin Reordering

In Section 2 we described a probability based model to compute the total leakage of a logic gate. The $I_{sub_{avg}}$ and $I_{gate_{avg}}$ for a logic gate under a given T_{ox} assignment are determined by computing the leakage of the dominant input states for I_{sub} and I_{gate}, respectively.

We will now consider the problem of transistor and pin reordering to reduce the average leakage power, which is the sum of $I_{sub_{avg}}$ and $I_{gate_{avg}}$. While it is possible to reduce $I_{sub_{avg}}$ for a logic gate via transistor and pin reordering, this reduction is often dwarfed by the dominant effect of the changes in $I_{gate_{avg}}$, and therefore, we will limit our discussion to $I_{gate_{avg}}$ in this section.

In order to motivate the idea of transistor reordering, consider an NMOS transistor stack in the pulldown of a 4-input NAND gate, as illustrated in Figure 3(a). In this example, transistors T_1 and T_4 have been assigned $T_{ox_{Hi}}$ and hence have low I_{gate}, whereas transistors T_2 and T_3 are assigned $T_{ox_{Lo}}$ leading to high I_{gate} values. In this example, we will assume I_{gate} for the transistors with $T_{ox_{Lo}}$ to be 10 nA, and for those with $T_{ox_{Hi}}$ to be 0.1 nA. We also assume that the probabilities of pins P_1, P_2, P_3 and P_4 being at logic "1" to be 0.1, 0.2, 0.3, and 0.4, respectively. These values are identical to the probability that the corresponding transistors to which the pins are connected are on.

The dominant state for I_{gate} for a particular transistor in the NMOS stack, say T_2, corresponds to the case where all of the transistors (T_3 and T_4) below it are on. Assuming that the inputs are all statistically independent, the probability of such a state will be the product of the probabilities of T_2, T_3 and T_4 being on. Similarly, the leakage for T_1, T_3 and T_4 can be found for their dominant states, and based on these calculations, the value of $I_{gate_{avg}}$ for the NMOS stack is computed to be 1.48nA, as shown in Figure 3(a).

Now consider the case of pin reordering. In order to reduce the probability of the dominant input state for transistor T_3, it is desirable that the pin with the highest probability be assigned to the transistor at the top of the stack, and that with the lowest probability be assigned to the bottom of the stack. This results in the configuration shown in Figure 3(b) and $I_{gate_{avg}}$ becomes 0.27nA, an 81% reduction from the original case.

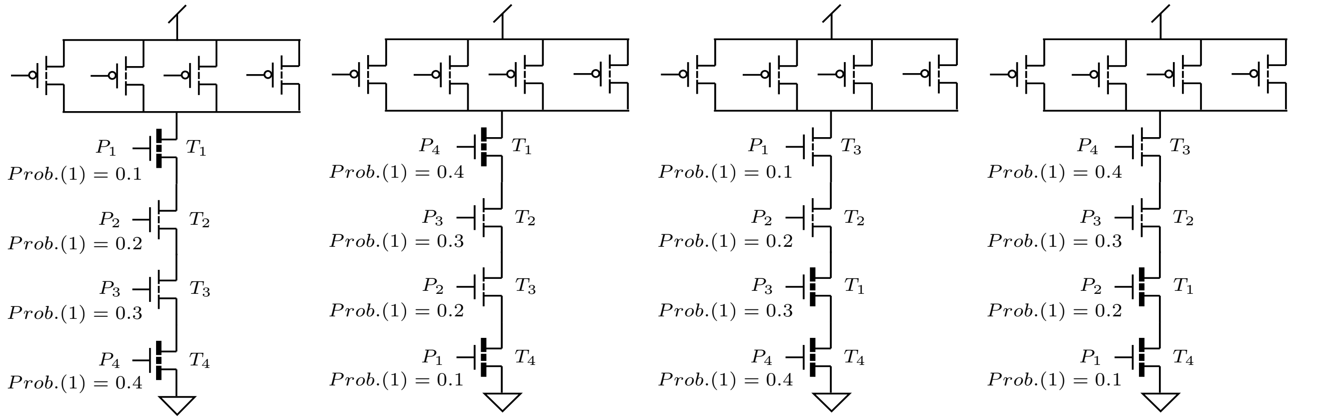
Similarly, instead of moving the pins, consider the case of transistor reordering, where the pins are fixed, but the transistors are moved. Specifically, the most leaky transistors (those assigned $T_{ox_{Lo}}$) can be moved to the top of the stack, as shown in Figure 3(c). In this case, the probability of the dominant state for the uppermost transistor, T_3, will be the probability of the entire stack being on. Observe that this probability for the topmost transistor is the lowest among all transistors in the stack (e.g., in the figure, T_3 corresponds to a probability of $0.1 \times 0.2 \times 0.3 \times 0.4$, while any lower transistor has a higher probability of a dominant state). Therefore, moving the most leaky transistors to the top of the stack yields a significant reduction in $I_{gate_{avg}}$, and we see from Figure 3(c) that this results in an $I_{gate_{avg}}$ of 0.316nA and a reduction of 78% from the original case.

Neither of the above reordering methods provide the maximum benefit when considered individually, and the best solution combines both the transistor and pin reordering, as shown in Figure 3(d). This results in an $I_{gate_{avg}}$ of 0.096nA and a total savings of 93% compared to the original case.

Any such changes also impact the gate delay, and hence, potentially, the circuit delay. In order to avoid any adverse impact on delay, we will develop a procedure in Section 4 that guarantees that only those transformations are accepted that result in zero or positive slack at the output of the logic gate during any step of the algorithm, and therefore guarantees that these transformations do not slow down the speed of the circuit. For this reason, it is entirely possible that the leakage-optimal arrangement for a gate, such as the one shown in Figure 3(d) may not be acceptable if it increases the circuit delay. We perform an exhaustive search on a gate-by-gate basis and accept the permissible configuration that satisfies the delay constraints. The total leakage of individual logic gate is considered during this exhaustive search in order to obtain reductions in the total expected leakage of the circuit rather than just I_{gate}.

4 Reordering Algorithm

In this section we describe our algorithm for finding the leakage-optimal configuration for the gates in a circuit under a specified delay constraint. The input to the algorithm is a netlist that has undergone dual T_{ox} optimization; in this case, this is provided by the algorithm in [7].


P_1 T_1 $Prob.(1) = 0.1$
P_2 T_2 $Prob.(1) = 0.2$
P_3 T_3 $Prob.(1) = 0.3$
P_4 T_4 $Prob.(1) = 0.4$
P_4 T_1 $Prob.(1) = 0.4$
P_3 T_2 $Prob.(1) = 0.3$
P_2 T_3 $Prob.(1) = 0.2$
P_1 T_4 $Prob.(1) = 0.1$
P_1 T_3 $Prob.(1) = 0.1$
P_2 T_2 $Prob.(1) = 0.2$
P_3 T_1 $Prob.(1) = 0.3$
P_4 T_4 $Prob.(1) = 0.4$
P_4 T_3 $Prob.(1) = 0.4$
P_3 T_2 $Prob.(1) = 0.3$
P_2 T_1 $Prob.(1) = 0.2$
P_1 T_4 $Prob.(1) = 0.1$

$$\begin{aligned} I_{gate,avg} &= 0.1\times0.2\times0.3\times0.4\times0.1 \\ &\quad + 0.2\times0.3\times0.4\times10 \\ &\quad + 0.3\times0.4\times10 \\ &\quad + 0.4\times0.1 \\ &= 1.480 \text{ nÅ} \end{aligned}$$

(a)

$$\begin{aligned} I_{gate,avg} &= 0.4\times0.3\times0.2\times0.1\times0.1 \\ &\quad + 0.3\times0.2\times0.1\times10 \\ &\quad + 0.2\times0.1\times10 \\ &\quad + 0.1\times0.1 \\ &= 0.270 \text{ nÅ} \end{aligned}$$

(b)

$$\begin{aligned} I_{gate,avg} &= 0.1\times0.2\times0.3\times0.4\times10 \\ &\quad + 0.2\times0.3\times0.4\times10 \\ &\quad + 0.3\times0.4\times0.1 \\ &\quad + 0.4\times0.1 \\ &= 0.316 \text{ nÅ} \end{aligned}$$

(c)

$$\begin{aligned} I_{gate,avg} &= 0.4\times0.3\times0.2\times0.1\times10 \\ &\quad + 0.3\times0.2\times0.1\times10 \\ &\quad + 0.2\times0.1\times0.1 \\ &\quad + 0.1\times0.1 \\ &= 0.096 \text{ nÅ} \end{aligned}$$

(d)

Figure 3: Various configurations for the pull-down of a 4-input NAND gate are shown here. The transistor gates with thick dotted lines correspond to a $T_{ox_{Hi}}$ assignment, while those with a thin dotted line correspond to an assignment of $T_{ox_{Lo}}$. The $I_{gate_{avg}}$ values for the NMOS transistor stack with (a) no transistor/pin reordering, (b) the best possible pin reordering only, (c) the best possible transistor reordering only, and (d) the best possible combination of transistor and pin reordering are shown here.

The circuit is represented by a graph where each gate corresponds to a node, and the interconnections between gates correspond to edges. In our implementation, the improved reordering configurations for a node, if any exist, will lead to a reduction in the total leakage ($I_{gate_{avg}} + I_{sub_{avg}}$) while either increasing or decreasing the node delay; in either case, the algorithm guarantees that the slack will remain positive. Bearing this in mind, we divide the search space of possible configurations into two categories:

Search_spc1 contains those nodes that have a reordering configuration resulting in an increase in node delay.

Search_spc2 contains those with a corresponding decrease in node delay.

The nodes in Search_spc2 are preferred since they reduce both leakage and delay. The cost function[4] assigned to each node is the total reduction in leakage. Therefore, a configuration for each node in this search space that has the maximum cost is chosen first, and these selections result in additional slack being created in the circuit.

This slack, and any existing slack in the circuit, can be "filled in" using node configurations in Search_spc1. The order in which these nodes are chosen is based on a TILOS-like [14] sensitivity based method. The node that provides maximum reduction in leakage with minimum increase in node delay is chosen. If $\triangle Lkg$ is the decrease in node leakage and $\triangle D$ is increase in node delay, we evaluate

$$Cost = \frac{\triangle Lkg}{\triangle D} \tag{4}$$

and select configurations for each gate in order of this cost until there is no leakage-reducing configuration that satisfies the delay constraints.

Algorithm 1 shows the heuristic employed in performing transistor and pin reordering. The primary input (PI) probabilities[5] are propagated to the intermediate nodes (line 4). In lines 5–9, the delay and leakage values for individual nodes are determined. A standard static timing analysis (STA) is then performed (line 10) in order to determine the slack of each node in the circuit. The search space, as explained above, is constructed in lines 11-14 using a subroutine described in Algorithm 2. The algorithm enters an iterative loop in lines 15–34. In each iteration, a node is selected based on the rule described above. In the event of a tie (for the case of Search_spc1), the node nearest to the primary output (PO) is chosen. Further ties are heuristically broken by selecting the node with lowest fanout; the rationale for the heuristic is that these have a smaller cone of influence and may affect fewer slack values. Observe that it is not necessary to break ties in the Search_spc2 case since the chosen configuration always results in a delay reduction. Once the appropriate node is chosen, the relevant data such as the arrival time and required time of effected nodes and the search spaces are updated. The iterations stop when there are no elements remaining in either search space.

5 Experimental Results

The proposed method for total leakage reduction was tested on the ISCAS85 benchmark circuits [15] at the 100nm and 70nm predictive technology nodes. The circuits were synthesized using SIS [16] based on a library consisting of inverters, as well as Nand and Nor gates with 2, 3, and 4 inputs. Capo [17] was then applied to obtain a placement, and finally the design was routed to obtain the interconnect wirelengths. The resulting wire lengths were used to determine the worst case interconnect capacitance (using interconnect parameters from [18]) for delay computations.

[4] This is something of a misnomer since the "cost" is actually a benefit in this case.

[5] We use a random function to generate PI probabilities.

Algorithm 1 Transistor-Pin-Reordering()

```
Input: A dual-$T_{ox}$ circuit
Output: A transistor/pin reordered dual-$T_{ox}$ circuit
/*Circuit is represented as an acyclic graph $G(V, E)$*/
Propagate state probabilities from PIs to internal nodes
for each node $x \epsilon G(V, E)$ do
    Find output load = $\sum_{\text{fanout nodes}}$ gate capacitance + interconnect capacitance
    Get rise, fall delays ($D_{P_{fall}}$, $D_{P_{rise}}$) from delay LUT
    Find $I_{sub}$, $I_{gate}$ based on leakage models
end for
Perform STA to find rise and fall $AT$, $RT$ for each node
Create empty sets, Search_spc1 and Search_spc2
for each node $x \epsilon G(V, E)$ do
    Update-Search-Space($x$)
end for
while (Search_spc1 and Search_spc2 are not empty) do
    if (Search_spc2 is not empty) then
        $N_{chosen}$ = Node with max. cost in Search_spc2
    else
        $N_{chosen}$ = Node with max. cost in Search_spc1
        /*Tie-breakers: #fanouts, proximity to PO*/
    end if
    Assign the best configuration to $N_{chosen}$
    Update $D_{P_{fall}}$, $D_{P_{rise}}$, $I_{sub}$, $I_{gate}$ of $N_{chosen}$
    Perform incremental STA to update rise and fall $AT$, $RT$ of effected nodes.
    for each node $y$ encountered during incremental STA do
        if ($y \epsilon$ Search_spc1) then
            Search_spc1 = Search_spc1 - {$y$}
        else if ($y \epsilon$ Search_spc2) then
            Search_spc2 = Search_spc2 - {$y$}
        end if
        Update-Search-Space($y$)
        /*nodes might be added, removed or their cost might change while updating the search space.*/
    end for
end while
```

Algorithm 2 Update-Search-Space(x)

```
if (Found best configuration with no negative slack) then
    if ($\triangle D > 0$) then
        Search_spc1 = Search_spc1 $\cup \{x\}$
        cost($x$) = $(\frac{\triangle Lkg}{\triangle D})_x$
    else
        Search_spc2 = Search_spc2 $\cup \{x\}$
        cost($x$) = $\triangle Lkg_x$
    end if
end if
```

SPICE simulations were based on a predictive model [19] using inverter transistor widths $W_n = 8\lambda/W_p = 16\lambda$ (widths for other gates were accordingly scaled). The values of V_{dd}, $T_{ox_{Lo}}$, and $T_{ox_{Hi}}$ that were used in our simulations are 1.2V, 12Å, and 17Å , respectively, at the 100nm node, and 1.0V, 11Å and 15Å , respectively, at the 70nm node.

The method in [7] was used to obtain a leakage/delay tradeoff curve for the placed and routed layout, and reordering was performed at each delay point on this curve. Figure 4 shows experimental results at the 100nm and 70nm technology nodes for a representative benchmark circuit. Each set of results show the tradeoff curves for before and after reordering, and the corresponding percentage reduction in I_{gate}, I_{sub} and the total leakage current. Observe that the delay remains the same after reordering, as constrained by our optimization. Furthermore, the savings observed in I_{gate} are seen to reduce as the delay reduces. The intuition behind this is as follows. As the delay decreases, the number of nodes that lie on the critical path increases. This constrains the permissible reordering on the nodes as our optimizer does not permit any transformation that would result in an overall delay increase.

Since the regions to the left of the knee of the curve do not constitute reasonable engineering solutions as they involve large increases in leakage for small delay reductions, the suitable design choices lie to the right of the knee of the tradeoff curve, and we limit our discussion to this region. Table 1 shows the percentage leakage reduction values at three design points on the leakage/delay tradeoff curve for each circuit. We choose one data point from the knee region (C1) and choose the remaining two points (C2 and C3) at arbitrary points to its right. The reductions in I_{gate} for C2 and C3 are significant, with a maximum savings of 24% and 26% for the 100nm and 70nm technology nodes, respectively. The savings in I_{gate} for C1 is relatively lower, with maximum reductions of 12% and 16% for the 100nm and 70nm nodes, respectively, and the reasons for this are described above. The reduction in I_{sub} is between 3-5% and is practically constant for all of the benchmarks. The CPU times for all circuits are shown in the table, and each number corresponds to the maximum of the CPU times over all points on the leakage/delay tradeoff curve. It is clear that the procedure is extremely fast, and only requires a few seconds.

The table also shows the reductions in the total leakage, which are seen to be up to 18.0% (for point C3 of C2670). Although these are not startlingly dramatic numbers, they still correspond to very solid reductions in the total leakage. An important point to note is that this is an in-place optimization with low layout impact, so that the reductions can actually be guaranteed, and are not likely to suffer from significant estimation errors.

References

[1] F. Hamzaoglu and M. R. Stan, "Circuit-Level Techniques to Control Gate Leakage for Sub-100 nm CMOS," in *Proc. of ACM/IEEE ISLPED*, pp. 60–63, Aug. 2002.

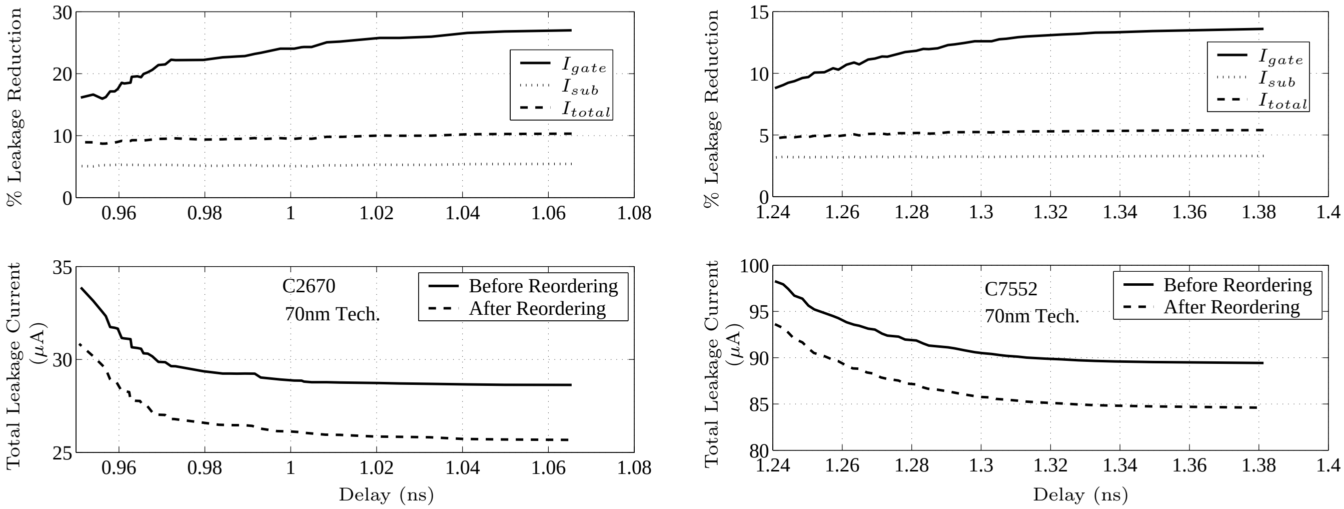
[2] M. Hirose *et al.*, "Fundamental Limit of Gate Oxide Thickness Scaling in Advanced MOSFETs," *Semiconductor Science and Technology*, vol. 15(5), pp. 485–490, May 2000.

[3] J. Kao *et al.*, "Transistor Sizing Issues and Tool for Multi-Threshold CMOS Technology," in *Proc. of ACM/IEEE DAC*, pp. 409–414, Jun. 1997.

[4] D. Lee and D. Blaauw, "Static Leakage Reduction through Simultaneous Threshold Voltage and State Assignment," in *Proc. of ACM/IEEE DAC*, pp. 191–194, Jun. 2003.

[5] Y. Oowaki *et al.*, "A sub-0.1 μm Circuit Design with Substrate-Over-Biasing," in *IEEE ISSCC Dig. of Tech. Papers*, pp. 88–89, Feb. 1998.

[6] S. Narendra *et al.*, "Leakage Issues in IC design: Trends, Estimation, and Avoidance." Tutorial at the IEEE/ACM ICCAD, Nov. 2003.



(b) (b)

Figure 4: Leakage/Delay tradeoff curve and percentage leakage reduction for C7552 for the (a) 100nm and (b) 70nm technology nodes.

Circuit	Percentage Leakage Reduction									CPU Time (sec)
	100nm Tech.									
	C1			C2			C3			
	I_{gate}	I_{sub}	I_{total}	I_{gate}	I_{sub}	I_{total}	I_{gate}	I_{sub}	I_{total}	
C432	4.5	4.5	4.5	14.0	5.0	7.5	18.0	6.0	9.0	0.41
C499	5.4	4.5	5.0	8.0	5.0	5.7	12.0	5.3	6.5	0.64
C880	8.0	6.0	6.7	15.0	6.0	8.0	19.2	6.5	9.0	0.28
C1355	3.2	2.0	2.5	5.5	3.3	4.0	8.0	3.5	4.5	0.56
C1908	3.6	3.2	3.4	8.0	3.7	4.6	10.0	3.7	5.0	0.89
C2670	11.0	7.0	8.5	21.0	7.5	11.5	24.0	12.0	18.0	0.89
C3540	7.0	5.4	6.0	13.5	5.5	7.4	15.0	5.7	7.7	2.17
C5315	12.0	6.0	8.0	18.0	6.3	9.5	20.0	6.5	10.0	2.72
C6288	3.0	3.0	3.0	5.0	3.0	3.5	6.0	3.2	3.7	16.19
C7552	8.0	4.5	5.5	12.0	4.7	6.2	13.2	4.8	6.7	2.56
	70nm Tech.									
C432	7.0	3.0	5.0	16.0	4.0	6.2	19.0	4.0	7.5	0.30
C499	7.0	3.3	4.5	10.0	3.5	5.0	12.5	3.6	5.2	0.59
C880	10.0	5.0	7.0	15.0	5.0	7.2	19.2	5.0	8.0	0.26
C1355	3.3	2.3	2.6	6.0	2.3	3.1	8.0	2.3	3.3	0.53
C1908	6.0	3.0	4.0	9.0	3.0	4.2	11.0	3.2	4.6	0.80
C2670	16.0	5.0	9.0	22.0	5.0	9.5	26.0	5.4	10.0	0.90
C3540	8.5	4.7	6.0	14.0	4.8	6.9	15.2	5.0	7.0	1.63
C5315	13.0	4.5	7.5	18.0	4.5	8.0	20.0	4.5	8.3	2.35
C6288	3.3	2.0	2.5	5.0	2.0	2.7	6.2	2.0	2.9	12.49
C7552	10.0	3.2	5.0	12.0	3.2	5.2	13.4	3.4	5.4	2.47

Table 1: Results of transistor and pin reordering, applied to a set of design points on the leakage/delay tradeoff curve provided by [7].

[7] A. Sultania, D. Sylvester, and S. S. Sapatnekar, "Tradeoffs between gate oxide leakage and delay for dual T_{ox} circuits," in *Proc. of ACM/IEEE DAC*, pp. 761–766, June 2004.

[8] R. Hossain *et al.*, "Reducing Power Dissipation in CMOS Circuits by Signal Probability Based Transistor Reordering," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and System*, vol. 15(3), pp. 361–368, Mar. 1996.

[9] E. Musoll and J. Cortadella, "Optimizing CMOS Circuits for Low Power using Transistor Reordering," in *Proc. of ED&TC*, pp. 219–223, Mar. 1996.

[10] S. C. Prasad and K. Roy, "Circuit Optimization for Minimization of Power Consumption under Delay Constraint," in *Proc. of International VLSI Design Conference*, pp. 305–309, Jan. 1995.

[11] D. Lee *et al.*, "Analysis and Minimization Techniques for Total Leakage Considering Gate Oxide Leakage," in *Proc. of ACM/IEEE DAC*, pp. 175–180, Jun. 2003.

[12] S. Sirichotiyakul *et al.*, "Duet: An Accurate Leakage Estimation and Optimization Tool for Dual-V_t Circuits," *IEEE Trans. on VLSI Systems*, vol. 10(2), pp. 79–90, Apr. 2002.

[13] K. A. Bowman *et al.*, "A Circuit-Level Perspective of the Optimum Gate Oxide Thickness," *IEEE Trans. on Electron Devices*, vol. 48(8), pp. 1800–1810, Aug. 2001.

[14] J. Fishburn and A. Dunlop, "TILOS: A Posynomial Programming Approach to Transistor Sizing," in *Proc. of ACM/IEEE ICCAD*, pp. 326–328, Nov. 1985.

[15] F. Brglez and H. Fujiwara, "A Neutral Netlist of 10 Combinatorial Benchmark Circuits," in *Proc. of ISCAS*, pp. 695–698, Jun. 1985.

[16] E. M. Sentovich *et al.*, "SIS: A System for Sequential Circuit Synthesis," Tech. Rep. UCB/ERL M92/41, Electronics Research Laboratory, Dept. of EECS, University of California, Berkeley, May 1992.

[17] Capo: A Large-Scale Fixed-Die Placer from UCLA. Available at: `http://vlsicad.ucsd.edu/GSRC/bookshelf/Slots/Placement/`.

[18] J. Cong, "Challenges and Opportunities for Design Innovations in Nanometer Technologies," in *SRC Design Sciences Concept Paper*, Dec. 1997.

[19] Device Group at UC Berkeley, "Berkeley Predictive Technology Model," 2002. Available at `http://www-device.eecs.berkeley.edu/~ptm/`.