Gate Oxide Leakage and Delay Tradeoffs for Dual T_{ox} Circuits

Anup Kumar Sultania§, Dennis Sylvester†, and Sachin S. Sapatnekar‡
§ Calypto Design Systems, Inc., Santa Clara, CA 95054.
† Department of EECS, University of Michigan, Ann Arbor, MI 48109.
‡ Department of ECE, University of Minnesota, Minneapolis, MN 55455.

Abstract*— Gate oxide tunneling current (I_{gate}) is comparable to subthreshold leakage current in CMOS circuits when the equivalent physical oxide thickness (T_{ox}) is below 15Å. Increasing the value of T_{ox} reduces the leakage at the expense of increased delay, and hence a practical tradeoff between delay and leakage can be achieved by assigning one of two permissible T_{ox} values to each transistor. In this paper, we propose an algorithm for dual T_{ox} assignment to optimize the total leakage power under delay constraints, and generate a leakage/delay tradeoff curve. As compared to the case where all transistors are set to low T_{ox}, our approach achieves an average leakage reduction of 86% under 100nm models and 81% under 70nm models. We also propose a transistor and pin reordering technique that has minimal layout impact to further reduce the total leakage current up to 12% and I_{gate} up to 27% without incurring any delay penalty.

I. INTRODUCTION

Leakage current is a primary concern for low power, high performance digital CMOS circuits for portable applications, and industry trends show that leakage will be roughly 50% of the total power in future technologies. New leakage mechanisms, such as tunneling across thin gate oxides, leading to gate oxide leakage current (I_{gate}), come into play at the 90nm technology and remain a daunting challenge for a number of technology nodes.

The International Technological Roadmap for Semiconductors (ITRS) [1] predicts that physical oxide thickness (T_{ox}) values of 7–12Å will be required for high performance CMOS circuits by 2006, and quantum effects that cause tunneling will play a dominant role in such ultra-thin oxide devices. The probability of electron tunneling is a strong function of the barrier height (i.e., the voltage drop across gate oxide) and the barrier thickness, which is simply T_{ox}, and a small change in T_{ox} can have 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} [2], so that reducing T_{ox} from 18Å to 12Å increases I_{gate} by approximately 1000×.[1] Moreover, the other component of leakage, subthreshold leakage (I_{sub}), forms a reducing fraction of the total leakage as T_{ox} is reduced, so that the development of I_{gate} reduction techniques is vital. The most effective way to control I_{gate} is through the use of high-k dielectrics, but such materials are not expected to come online until the 2007-2010 timeframe.

This paper explores the use of dual T_{ox} values for performance optimization, considering a leakage-delay tradeoff. In order to simplify the search space, we divide this optimization in two stages. We first perform T_{ox} assignment based on a cost function, and then postprocess the result to perform *transistor and pin reordering*. Although this optimization can be exploited at a number of points in the design methodology, our solution considers T_{ox} assignment as a step that is performed after placement and transistor sizing, at which point it is used to achieve a final performance improvement. Unlike earlier stages of design, there is less design uncertainty at this point and minor changes in layout parasitics due to T_{ox} assignment can be dealt with as an incremental update. As a result, all of the delay gains from our procedure are guaranteed in the final design, with a low leakage power overhead. Furthermore, *transistor and pin reordering* is a postprocessing step that has a low layout impact, and is therefore an inexpensive optimization in terms of the changes that it may induce in the design.

Leakage power can be broadly divided into two categories, depending on the mode of operation of the circuit: *standby leakage*, which corresponds to the situation when the circuit is in a non-operating or sleep mode, and *active leakage*, which relates to leakage during normal operation. Numerous effective techniques for controlling standby leakage have been proposed in the past, including state assignment [4], the use of multiple threshold CMOS (MTCMOS) sleep transistors [5], body-biasing [6], and dual T_{ox} combined with state assignment [7]. Active leakage, on the other hand, has not been widely addressed in the literature to date, primarily because it has not been a major issue in present technologies. However, leakage power dissipation in the active mode has grown to over 40% in some high-end parts today [8]. Therefore, reducing active leakage is vital for advanced technologies in current-generation circuits and for next-generation technologies. The range of options that are available for reducing active leakage is considerably more limited than for standby leakage, and the use of dual T_{ox} assignments is a powerful method for this purpose.

Prior research related to our work[2] is summarized as follows. In [11], the impact of I_{gate} on delay is discussed, but its impact on leakage power is not addressed. The work in [12] presents an approach to reducing I_{sub}, but not I_{gate}, using separate optimizations to select the values of T_{ox}. Similarly, several research works [13]–[15] 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

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

[1]The fundamental limit of T_{ox} scaling is projected to be about 8Å [3].

[2]This paper is based on our two previous conference publications [9], [10].

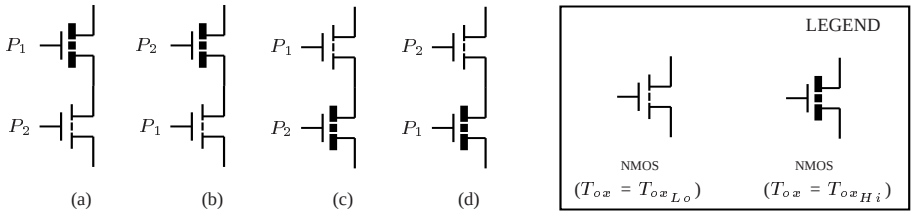


Fig. 1. All possible configurations using pin and transistor reordering for two NMOS transistors in a series - (a) initial configuration, (b) after pin reordering is applied to the initial configuration, (c) after transistor reordering is applied to initial configuration, and (d) after both transistor and pin reordering is applied to the initial configuration. The transistor gates with thick dotted lines correspond to a $T_{ox_{Hi}}$, while those with thin dotted line correspond to $T_{ox_{Lo}}$ assignment.

dissipation in the active mode. In [16], the authors examine the interaction between I_{gate} and I_{sub}, and their state dependencies. They 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 reordering 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)[3].

In our context, where we optimize the total leakage comprising both I_{gate} and I_{sub}, the rationale for optimizing T_{ox} is as follows. Choosing a lower value of T_{ox} can result in lower delays, but at the cost of increased leakage, and the value of T_{ox} can therefore be optimized to obtain a leakage/delay tradeoff. To maintain manufacturability and avoid enhanced short channel effects, it is important to scale the effective channel length L_{eff} along with T_{ox} [17]. Similarly, while applying transistor and pin reordering, the best configuration for each logic gate is chosen such that it results in maximum total leakage reduction without increasing circuit delay.

Due to processing constraints, rather than an unlimited range of T_{ox} values, it is more reasonable to choose between two permissible values. A suitable choice of T_{ox} should keep the I_{gate} to I_{sub} ratio to a reasonable value, as otherwise I_{gate} would completely dominate the total leakage current in the circuit. Furthermore, the two permissible values for T_{ox} should be fairly far apart in order to observe a noticeable tradeoff between total leakage and delay.

The organization of this paper is as follows. In Section II, we describe a method for selecting appropriate values of the low and high values of the oxide thickness, referred to as $T_{ox_{Lo}}$ and $T_{ox_{Hi}}$, respectively, and the corresponding values for the channel length. Next, in Sections III and IV, respectively, we introduce the leakage and delay models that are used in this work, and demonstrate that they show a good degree of accuracy compared to simulation results. Our iterative algorithm for finding the leakage/delay tradeoff is then presented in Section V. Next, we describe a transistor and pin reordering technique for I_{gate} minimization and reordering algorithm in Section VI and Section VII, respectively. Our experimental results are discussed in Section VIII and concluding remarks given in Section IX.

[3]This assumes the possibility of having different T_{ox} values in a series-connected stack, which may or may not be easily achievable from a technology standpoint

II. CHOOSING T_{ox} AND L_{eff}

While an increased value of T_{ox} can significantly reduce I_{gate}, several other physical effects must be taken into consideration. Increasing the value of T_{ox} while keeping the channel length constant may adversely impact the functionality of the transistor. Specifically, due to drain induced barrier lowering (DIBL), an increase in T_{ox} may result in a situation where the drain terminal takes additional control of the channel, so that the "on" or "off" state of the transistor is no longer completely governed by the gate terminal.

This effect is easily recognized during technology scaling, and scaling trends have shown that T_{ox} reduces nearly in proportion with L_{eff} [18]. We maintain this proportion for each of the chosen values of T_{ox} by setting

$$\frac{L_{eff}@T_{ox_{Lo}}}{T_{ox,e_{Lo}}} = \frac{L_{eff}@T_{ox_{Hi}}}{T_{ox,e_{Hi}}} \quad (1)$$

The term $T_{ox,e}$ in this equation refers to the *electrical* T_{ox}, which is related to the *physical* value of T_{ox} as follows[4]

$$T_{ox,e} = T_{ox} + T_{ox_{\text{offset}}} \quad (2)$$

The $T_{ox_{\text{offset}}}$ term is added to account for the gate depletion and channel quantization effects, and a typical value is 0.7nm [19]. In the remainder of this paper, it will be implicit that as we change T_{ox}, the value of L_{eff} is also scaled.

Before determining reasonable values for $T_{ox_{Lo}}$ and $T_{ox_{Hi}}$, we study the effect of varying T_{ox} on leakage for an inverter, whose NMOS and PMOS transistors are sized to be 0.8μm and 0.4μm, respectively, in a 100nm technology. The gate oxide leakage, I_{gate}, and the subthreshold leakage, I_{sub}, for both the NMOS and PMOS transistors in the inverter, are graphically depicted in Figure 2(a) for various values of $T_{ox_{Hi}}$, at $T_{ox_{Lo}} = 12$Å; the sum of these components is shown by the bottommost curve in Figure 2(b). The values of I_{sub} are obtained through SPICE simulations on predictive technology models [20], and an analytical model (described in Section III-B) is used to generate I_{gate}.[5] The average leakage of the inverter is calculated as the sum of the average I_{gate} and I_{sub} leakages (as described in greater detail in Section III), and is shown in Figure 2(b).

[4]Henceforth, our discussions will be with reference to T_{ox}, the *physical* value of the gate oxide thickness.

[5]We cannot use simulations here since the Berkeley predictive technology model [20] uses BSIM3, which does not model I_{gate}.

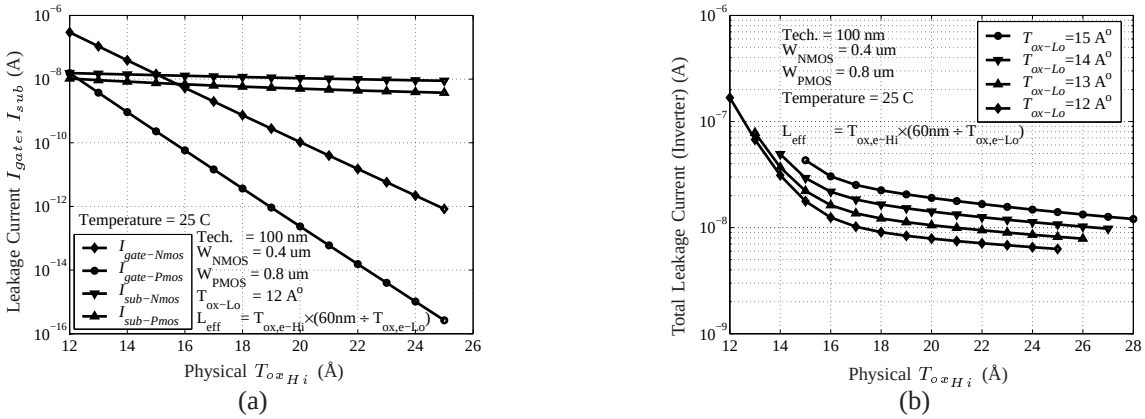
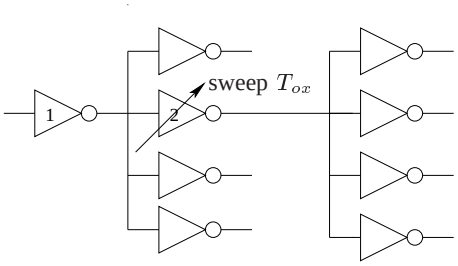


Fig. 2. (a) The four leakage components for an inverter (I_{gate} and I_{sub} for the NMOS and PMOS transistors, respectively) as a function of the gate oxide thickness. (b) The total leakage of an inverter for different values of $T_{ox_{Lo}}$ and $T_{ox_{Hi}}$. At each point, L_{eff} is scaled with respect to the minimum T_{ox} value on the curve; at this point, $L_{eff} = 60$nm.



(a)

T_{ox} (Å)	$T_{ox,e}$ (Å)	L_{eff} (nm)	D_{Inv1} (ps)	D_{Inv2} (ps)	C_{inv} (fF)	V_{th} (V)
12	19	60.0	33.84	33.56	1.98	0.119
14	21	66.3	33.77	36.70	1.99	0.120
16	23	72.6	33.71	39.98	1.99	0.122
18	25	78.9	33.67	43.40	1.99	0.124
20	27	85.2	33.64	46.97	2.00	0.126
22	29	91.6	33.62	50.69	2.00	0.127

(b)

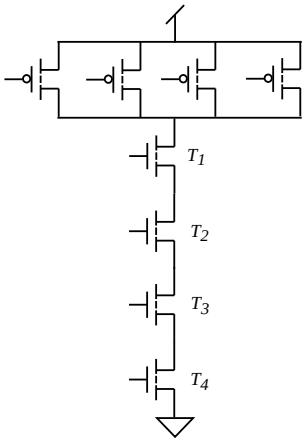
Fig. 3. (a) A test circuit for examining the effect of varying the T_{ox} value of an inverter on a larger circuit (b) A tabulation of results that show the effect of varying the T_{ox} value of Inverter 2 on its own delay, on delay of its fanin gate, Inverter1, on the input capacitance of its own input capacitance, C_{inv}, (calculated as the sum of the NMOS and PMOS gate capacitances), and on the threshold voltage V_{th} of its NMOS device. The transistor widths are chosen as $W_n = 0.4\mu m$ and $W_p = 0.8\mu m$ in a 100nm technology.

As T_{ox} is varied, I_{sub} shows a negligible change in comparison to I_{gate}. Furthermore, the average leakage decreases slowly for $T_{ox} > 17$Å, and increases sharply as T_{ox} goes below 17Å. On the other hand, the delay of the inverter (as will be seen by the experiment in Figure 3) increases linearly with T_{ox}, so that using a value of $T_{ox_{Hi}}$ of over 17Å results in a larger delay with no appreciable savings in total leakage. This leads us to choose $T_{ox_{Hi}} = 17$Å.

To choose $T_{ox_{Lo}}$, we consider several scenarios as shown by the plots in Figure 2(b). Each curve corresponds to a different choice of $T_{ox_{Lo}}$, and the value of L_{eff} is set to 60nm at this value. Each point on a curve now shows the total leakage for an inverter whose transistors are set to a candidate value of $T_{ox_{Hi}}$. For instance, for the curve where $T_{ox_{Lo}} = 15$Å, candidate values for $T_{ox_{Hi}}$ range from 28Å to 15Å, and the L_{eff} value for each case is scaled in accordance with Equation (1). Observe that for a given $T_{ox_{Hi}}$ on the curve, the total leakage decreases as $T_{ox_{Lo}}$ reduces. This is because, for the same L_{eff} values, a reduction in the corresponding $T_{ox_{Lo}}$ value reduces short-channel effects. For a fixed value of $T_{ox_{Hi}}$, this results in a reduction in the total leakage as $T_{ox_{Lo}}$ is decreased. It is easily seen that on each curve, the T_{ox} value at which the leakage begins to change steeply is about 17Å. In other words, for the entire range of candidate $T_{ox_{Lo}}$ values of 12Å through 15Å, our choice of $T_{ox_{Hi}}$=17Å is reasonable in terms of the leakage values. To incorporate delay considerations, we observe that in order to achieve a wider range of delay values, the difference between $T_{ox_{Lo}}$ and $T_{ox_{Hi}}$ should be as high as possible (we will soon substantiate this with an experiment). The choice of $T_{ox_{Lo}}$, however, is limited by several factors such as reliability and the maximum desired I_{gate}/I_{sub} ratio [1]. This ratio, at $T_{ox_{Lo}}$, should be such that I_{gate} does not completely dwarf I_{sub}. Furthermore, due to process variation in T_{ox}, the choice of $T_{ox_{Lo}}$ and $T_{ox_{Hi}}$ should be such that their probability distribution functions do not have a significant overlap. We choose $T_{ox_{Lo}} = 12$Å as it gives the best achievable leakage/delay tradeoff. A similar analysis is performed for the 70nm technology node, and provides values of $T_{ox_{Lo}} = 11$Å and $T_{ox_{Hi}} = 17$Å.

We now consider the impact of changing T_{ox} and L_{eff} on two parameters that they must clearly affect: the gate capacitance, C_{inv}, and the threshold voltage, V_{th}, of the MOS devices. We perform a set of SPICE simulations on a circuit set-up illustrated in Figure 3(a), and show the simulation results in the table in Figure 3(b). In this experiment, the T_{ox} value of Inverter 2 is varied, and all other inverters are maintained at a fixed T_{ox} value of 17Å. The proposed method of scaling the value of L_{eff} linearly with T_{ox} results in *nearly constant* values of C_{inv} and V_{th}, respectively. However, there is a noticeable impact on gate delay: increasing T_{ox} and L_{eff} decreases the channel transconductance, and hence increases delays. Changing T_{ox} from 12Å to 22Å alters the delays *linearly*, with a delay penalty of 51% over this range for Inverter 2.

The invariance of the capacitance of Inverter 2 over the entire range of T_{ox} has two notable consequences:



(a)

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

(b)

Fig. 4. (a) A four-input NAND gate. (b) The variation of I_{sub} in a 100nm technology through the pull-down chain, for the dominant state when only transistor T_4 (which uses $T_{ox_{Lo}}$) is off, under various combinations of T_{ox} for the other transistors. Here, $T_{ox_{Lo}} = 12$Å (*Lo*), $T_{ox_{Hi}} = 17$Å (*Hi*), and T_4 is at $T_{ox_{Lo}}$.

- A change in T_{ox} of a transistor leaves the load capacitance presented to the previous stage of logic unchanged. As a result, the delay of a fanin logic gate does not change significantly, and hence our optimization method needs only to consider the delay change of a given logic gate when its T_{ox} is altered.
- Since the capacitance is unchanged, the $CV_{dd}^2 f$ (dynamic) power remains unaffected by changes in T_{ox}. This is extremely important since our optimization is therefore guaranteed to reduce the total power, even though it focuses on minimizing leakage.

III. LEAKAGE MODELS

We will now 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 corresponding average I_{sub} and I_{gate}.

A. Subthreshold Leakage Model

As seen in the Figure 3(b), the value of V_{th} changes by a very small amount as T_{ox} is changed. In spite of this, it can have significant effects on I_{sub}, which is exponentially dependent on V_{th}. For convenience, we use a simple look-up table (LUT) to determine I_{sub}. Conceptually, 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[6], and each T_{ox} assignment would require entries for the $2^k - 1$ leakage states corresponding to different input logic values[7], resulting in a total of $2^k \cdot (2^k - 1)$ entries.

The LUT size can be reduced significantly using the following ideas:

Dominant input states: It has been shown [21] 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 4(b). 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 one at $T_{ox_{Lo}}$. As a result, the LUT size can be brought down to $2k$ entries.

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 (one each for $T_{ox_{Hi}}$ 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:

$$I_{sub,avg.} = \sum_{i \in \text{ dominant input states}} P_{state_i} \times I_{sub_i} \qquad (3)$$

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

B. 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 (I_{gc}) region, and in the gate-to-drain/source (I_{gd} and I_{gs}, respectively) overlap

[6]Series-connected devices can have different T_{ox} and the design rules that take this into account would increase the spacing between such devices as compared to the case where all devices have identical T_{ox} values.

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

regions. The latter type of tunneling, referred to as edge direct tunneling (EDT) is ignored in our case for three reasons: first, because the gate-to-drain/source overlap region is significantly smaller than the channel region [11], second, because the oxide thickness in this overlap region can be increased after gate patterning to further suppress EDT [22] and third, because EDT is smaller than tunneling in gate-to-channel region [23]. We also neglect the OFF state gate oxide leakage and consider only the ON state I_{gate} values [24].

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) [25].

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

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), and $\gamma = 4\pi T_{ox}\sqrt{2M_{ox}}/h$ 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 [16] that like I_{sub}, I_{gate} also exhibits a state dependency. When the gate node of the NMOS transistor is at logic 0, the only possible tunneling component is EDT, which is neglected in our work; therefore, we will only consider the cases where the gate node is at logic 1. For example, while determining I_{gate} for transistor T_2 in the 4-input NAND gate in Figure 4(a), it can be shown that the maximum leakage for T_2 occurs at the input state[8] $(x,1,1,1)$, 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. This is because, for the later three sets of states, voltage level at the source node of transistor T_2 increases due to the combined effect of I_{sub} and I_{gate}. This results in a smaller gate-to-source voltage for T_2. It is known that I_{gate} reduces by an order of magnitude for each 0.3v reduction in gate-to-source voltage [2]. A reduction in gate-to-source voltage by 0.3v is possible for transistors at $T_{ox_{Lo}}$. Thus the dominant state of I_{gate} for T_2 is $(x,1,1,1)$. Observe that for transistors at $T_{ox_{Hi}}$, I_{gate} is not of concern as I_{sub} dominates the total leakage current. For further details, the reader is referred to [16].

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:

$$I_{gate,avg.} = \sum_{\text{transistor } i \in \text{ logic gate}} P_i \times I_{gate_i} \quad (5)$$

Here, P_i 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 as in a NAND (NOR) gate, P_i 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 (4) 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} automatically ignores the complex interaction between these two components, which was noted in [16].

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

TABLE I

DELAYS FROM THE INPUT OF SWITCHING TRANSISTOR T_2 IN A 4-INPUT NAND [FIGURE 4(A)] @$T_{ox_{Lo}}$ ($T_{ox_{Lo}} = 12$Å, $T_{ox_{Hi}} = 17$Å).

	T_{ox}				Delay		
	T_1	T_2	T_3	T_4	Spice	LUT	Error
D_0	*Lo*	*Lo*	*Lo*	*Lo*	13.89	—	—
D_1	*Lo*	*Lo*	*Lo*	*Hi*	14.84	14.51	-2.22 %
D_2	*Lo*	*Lo*	*Hi*	*Lo*	14.21	14.51	2.11 %
D_3	*Hi*	*Lo*	*Lo*	*Lo*	14.54	14.51	-0.21 %
D_4	*Lo*	*Lo*	*Hi*	*Hi*	15.11	15.13	0.13 %
D_5	*Hi*	*Lo*	*Lo*	*Hi*	15.47	15.13	-2.20 %
D_6	*Hi*	*Lo*	*Hi*	*Lo*	14.86	15.13	1.82 %
D_7	*Hi*	*Lo*	*Hi*	*Hi*	15.75	—	—

IV. DELAY MODEL

For advanced nanometer technologies, it is difficult to obtain accurate closed-form delay models, and we therefore use an LUT-based approach for delay modeling. For each input of the 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 performed on this 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 the gate input capacitance for each logic gate. The output load for a logic gate can be computed by summing the input gate capacitances of the fanout logic gates as well as any wireload model that may be used. The delay of the logic gate can now be obtained using output load, slope and intercept values.

The input transition time is not accounted for in the above model, although it is straightforward to extend the model to include this effect. 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 only a small loss in accuracy in the following way.

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 $T_{ox_{Hi}}$. 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. This is illustrated for a 4-input NAND gate in Table I for the delay from the input of T_4 to the output. We fit a simple formula as follows:

$$Delay = D_0 + n \times \frac{(D_7 - D_0)}{(k-1)} \quad (6)$$

where D_0 and D_7 are delay values (stored in the LUT) for the extreme cases of non-switching transistors being at all $T_{ox_{Lo}}$ and all $T_{ox_{Hi}}$, respectively, as shown in Table I, n is the number of transistors (other than the switching transistor) at $T_{ox_{Hi}}$ and (k-1) is 3 for a 4-input Nand gate. The errors under this method are shown in Table I. Therefore, all possible fall delay scenarios for a k-input NAND gate can be compacted into $4k$ LUT entries. This technique was applied to several gate types, and in most cases, the error was under 2%, with a worst-case error of 3%.

A similar compression for the case of output rise LUTs of a k-input NAND is possible. Since the PMOS transistors are in parallel, only the gate-to-drain overlap capacitance at the output node changes for different T_{ox} combinations for the transistors; this has an insignificant impact on the delay, and hence, $2k$ LUT entries (corresponding to $T_{ox_{Hi}}$ and $T_{ox_{Lo}}$ for each PMOS input) are sufficient.

A similar approach can be applied to build LUTs for a k-input NOR gate, and for other types of logic gates. Therefore, the total number of LUT entries varies linearly with the number of inputs to the logic gate. Furthermore, the input transition time can be accounted for in this model by creating one such LUT for each candidate transition time.

V. DUAL T_{ox} ASSIGNMENT

In this section we describe our heuristic to obtain acceptable tradeoffs between leakage and delay in a dual T_{ox} circuit. The input to the algorithm is a combinational netlist. The circuit is represented by a graph where each gate corresponds to a node and the interconnections between gates correspond to edges. We use a TILOS (TImed LOgic Synthesizer) like [26] sensitivity-based heuristic for assigning T_{ox} values to individual transistors in a circuit. A standard static timing analysis (STA) approach is used to find the critical path. The propagation delay D_p for each gate is computed using the LUTs described in Section IV. In principle, the STA must be repeated after each T_{ox} change; however, we observe that every such T_{ox} change is sufficiently local and only changes delays and arrival times in its transitive fanout region. Therefore, after the first iteration, we achieve efficiency by performing incremental STA that processes only the affected regions.

Once this critical path is found, the core of the optimizer iteratively changes one transistor on this path from $T_{ox_{Hi}}$ to $T_{ox_{Lo}}$ in each iteration. This transistor is identified by measuring the increase in the total average leakage, $\triangle Lkg$, with respect to the delay reduction, $\triangle D$, observed on the critical path when such a change is made. In other words, we evaluate

$$Cost = \frac{\triangle D}{\triangle Lkg} \tag{7}$$

The transistor with the minimum (most negative) cost provides the largest delay reduction for the smallest increase in leakage power, and is selected for assignment to $T_{ox_{Lo}}$. The corresponding L_{eff} is also concurrently changed as described earlier. If two transistors have the same cost, ties are heuristically broken, first by selecting the transistor with the higher fanout. The rationale for such a tiebreaking method is that this gate will have a larger cone of influence, and is likely to reduce the delay on a larger number of paths.

Algorithm 1 Pseudocode for Dual T_{ox} Assignment()

```
1:  Input: A combinational logic circuit
2:  Output: Leakage/delay tradeoff curve
3:  /*Circuit is represented as an acyclic graph G(V,E)*/
4:  /*The target delay is D_T*/
5:  Initialize all transistors to T_oxHi
6:  Propagate state probabilities from PI's to internal nodes
7:  for each node x ϵ G(V,E) do
8:     Find output load = ∑fanout nodes gate capacitance
9:     Get rise, fall delays (D_Pfall, D_Prise) from delay LUT
10:    Find I_sub, I_gate based on LUT's
11: end for
12: Perform STA to find rise and fall AT, RT for each node
    and circuit delay, D_max
13: while D_max > D_T do
14:    (△D/△Lkg)_worst = 0; N_chosen = NULL;
15:    for each node y on a critical path do
16:       if (critical path transistor(s) of y are at T_oxHi) then
17:          find (△D/△Lkg)_y for node y
18:          if (△D/△Lkg)_worst > (△D/△Lkg)_y then
19:             (△D/△Lkg)_worst = (△D/△Lkg)_y; N_chosen = y
20:          end if
21:       end if
22:    end for
23:    if (△D/△Lkg)_worst ≠ 0 then
24:       Assign T_oxLo to the worst transistor in N_chosen
25:       Update D_Pfall, D_Prise, I_sub, I_gate of N_chosen
26:       Perform Incremental STA and recalculate D_max
27:    else
28:       Report D_max; Exit()
29:    end if
30: end while
```

In evaluating $\triangle D$, it is sufficient to find the delay change of the logic gate that the transistor belongs to. Since changes in T_{ox} leave the transistor input capacitance unchanged (see Section II), the delay of the fanin gate is unchanged.

Algorithm 1 shows the heuristic for T_{ox} assignment. At the start of the algorithm all transistors are assigned to $T_{ox_{Hi}}$ (line 3). The primary input (PI) probabilities[9] 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 arrival time, required time and delay of each node in the circuit. Next, the algorithm enters an iterative loop (lines 13–30). In each iteration, it greedily identifies the transistor on the critical path that, when changed to $T_{ox_{Lo}}$, causes the largest delay reduction for the smallest increase in leakage. This iteration stops when no further improvement is possible, thus generating a complete leakage-delay tradeoff curve. Figure 5 shows a flow diagram

[9] In our implementation, we use a random function to generate the probabilities at the PIs.

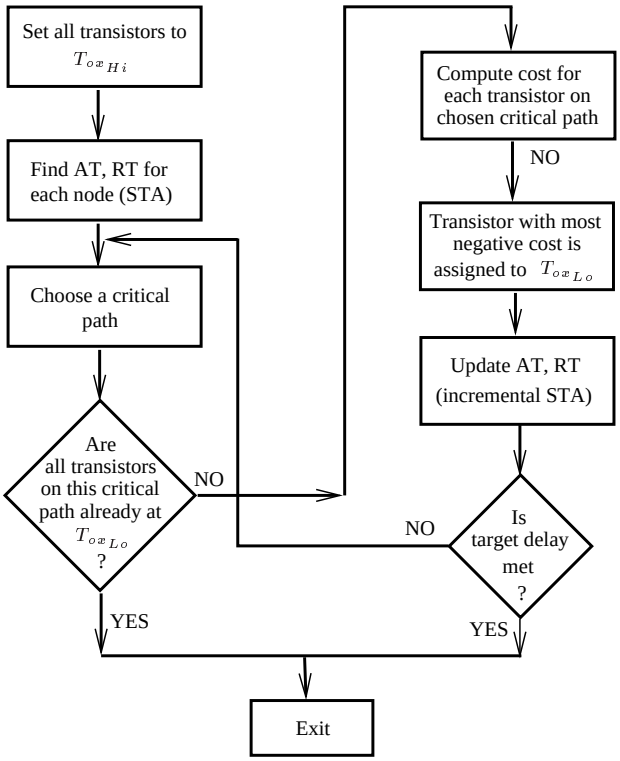


Fig. 5. Flow diagram for dual-T_{ox} assignment (Algorithm 1)

of Algorithm 1. This figure gives a general understanding of our dual-T_{ox} optimization.

The time complexity of this algorithm is $O(n^2)$, where n is the total number of logic gates in a circuit. Iteration (lines 13–30), in the worst case, will stop after assigning all of the transistors in a critical path to $T_{ox_{Lo}}$, hence it is bounded by $O(n)$. Each iteration performs an incremental STA, which, in the worst case, is linear in n. Therefore the total time complexity of Algorithm 1 is $O(n^2)$. However, it is also worth pointing out that this is a rather pessimistic analysis that does not reflect how the algorithm performs on typical examples. In most cases, the number of iterations is significantly smaller than n, and the cost of incremental STA is, in practice, almost a constant time computation.

VI. TRANSISTOR/PIN REORDERING

In Section III, a probability-based model for computing the total leakage of a logic gate was described. 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 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, our observation so far has been that reordering has a stronger impact on $I_{gate_{avg}}$ as opposed to $I_{sub_{avg}}$, 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 pull-down of a 4-input NAND gate, as illustrated in Figure 6(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. For simplicity, we will assume here that I_{gate} for the transistors with $T_{ox_{Lo}}$ is 10 nA, and for those with $T_{ox_{Hi}}$ is 0.1 nA. We also assume that the probabilities of pins P_1, P_2, P_3 and P_4 being at logic "1" are 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, e.g., 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 (i.e., $(T_1, T_2, T_3, T_4) = (x, 1, 1, 1)$), 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 6(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 6(b) and $I_{gate_{avg}}$ becomes 0.27nA, an 81% reduction from the original case.

Similarly, instead of moving the pins now consider the case of transistor reordering, where the pins are fixed while 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 6(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 6(c) that this results in an $I_{gate_{avg}}$ of 0.316nA, 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 6(d). This results in an $I_{gate_{avg}}$ of 0.096nA and a total savings of 93% compared to the original case. It is worth noting that the magnitude of the savings depends on the probability values at the inputs: for example, if all input probabilities are 0.5, the savings are 49%.

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 VII 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 overall speed of the circuit. For this reason, it is entirely possible that the leakage-optimal arrangement for a gate, such as the one shown

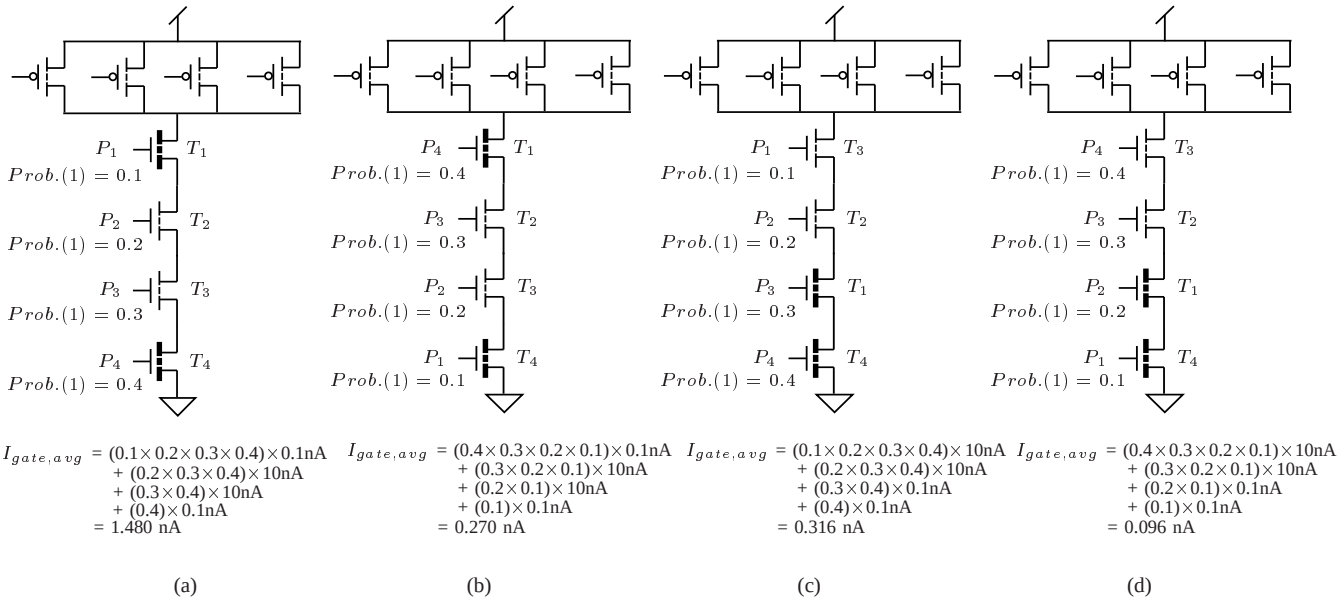


Fig. 6. Various configurations for the pull-down of a 4-input NAND gate are shown here. The transistor gates with thick dotted lines correspond to a $T_{ox_{Hi}}$ assignment, while those with a thin dotted line correspond to an assignment of $T_{ox_{Lo}}$. The $I_{gate_{avg}}$ values for the NMOS transistor 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. $I_{gate,avg} = Prob.(state(1,1,1,1)) \times I_{gate}(T_1) + Prob.(state(x,1,1,1)) \times I_{gate}(T_2) + Prob.(state(x,x,1,1)) \times I_{gate}(T_3) + Prob.(state(x,x,x,1)) \times I_{gate}(T_4)$, where 'state' corresponds to logic values at inputs to (T_1, T_2, T_3, T_4).

in Figure 6(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 gates is considered during this exhaustive search in order to obtain reductions in the total expected leakage of the circuit rather than just I_{gate}.

VII. REORDERING ALGORITHM

We now describe our algorithm for finding the leakage-optimal configuration for the logic gates in a circuit under a specified delay constraint. The input to the algorithm is a netlist that has undergone dual T_{ox} optimization, i.e., a specific design choice on the leakage/delay tradeoff curve obtained in Section V.

The optimization for leakage reduction through reordering is performed under the constraint that the circuit delay must remain the same. For a specific node, the improved reordering configurations will lead to a reduction in the total leakage ($I_{gate_{avg}} + I_{sub_{avg}}$) while either increasing or decreasing the node delay: any increase in the node delay must be within the slack at the node, so as not to increase the circuit delay.

To ensure that the slack remains positive, we divide the search space of possible configurations into two categories:

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

Search_spc2 contains those with a corresponding reduction in the node delay.

The nodes in Search_spc2 are preferred since they reduce both leakage and delay. The cost function[10] assigned to each node is the reduction in total leakage. Therefore, the configuration for each node in the second 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 consumed using node configurations from Search_spc1. The order in which these nodes are chosen is based once again on a TILOS-like [26] sensitivity-based method. The node that provides maximum ratio of leakage reduction to node delay increase is chosen. If $\triangle Lkg$ is the decrease in node leakage and $\triangle D$ is increase in node delay, we evaluate

$$Cost = \frac{\triangle Lkg}{\triangle D} \tag{8}$$

and select configurations for each gate in order of this cost until there is no leakage-reducing configuration that satisfies the delay constraints. It should be noted that we perform reordering on equal-sized stack of transistors. For the case where the transistors in a stack have unequal sizes, there could be a cost associated with reordering, and this could be taken into account by appropriately modifying the above cost function.

Algorithm 2 shows the heuristic employed in performing transistor and pin reordering. Lines 4–10 are the same as described in Section V for Algorithm 1. The search space, as explained above, is constructed in lines 11-14 using a subroutine described in Algorithm 3. 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 with lowest fanout is chosen. The rationale for this tie-breaking heuristic is that these have a smaller cone of influence and may affect fewer slack values. Observe that it is not necessary to break

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

Algorithm 2 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} = most negative cost node in Search_spc2
    else
        N_{chosen} = most negative cost node in Search_spc1
    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 3 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

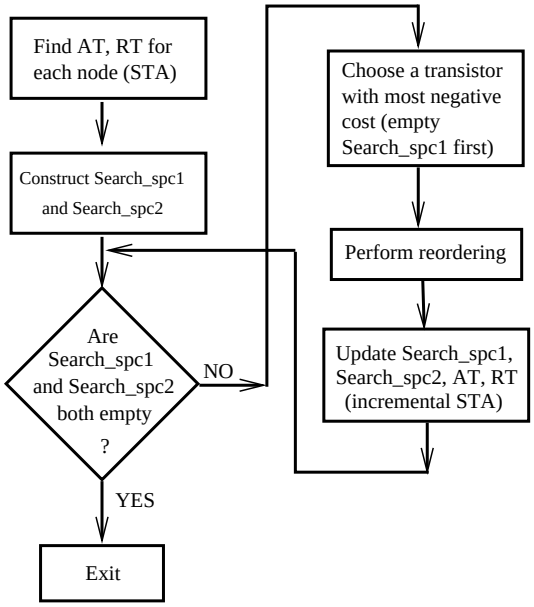


Fig. 7. Flow diagram for transistor and pin reordering (Algorithm 2)

ties in the Search_spc2 case since the chosen configuration always results in a delay reduction. Once the appropriate node is chosen, relevant data such as the arrival times and required times of affected nodes and the search spaces are updated. The iterations stop when there are no elements remaining in either search space. Figure 7 shows a flow diagram of Algorithm 2. This figure gives a general understanding of the transistor and pin reordering technique.

The time complexity of this algorithm is $O(n^2)$, where n is the total number of logic gates in a circuit. The complexity analysis is same as that of Algorithm 1, and the same caveats with respect to the validity of this analysis on typical circuits hold.

VIII. EXPERIMENTAL RESULTS

The proposed methods for optimizing total leakage were applied to the ISCAS85 benchmark circuits [27] at the 100nm and 70nm predictive technology nodes. The circuits were synthesized for minimum delay using SIS [28], using the "-n1 -AFG" options, based on a library consisting of inverters, as well as NAND and NOR gates with 2, 3, and 4 inputs. Capo [29] was then applied to obtain a placement, and finally the design was routed [30] to obtain interconnect wirelengths. The resulting wirelengths were used to determine the worst-case interconnect capacitance (using interconnect parameters from [31]) for delay computations. SPICE simulations were based on a predictive model [20] using inverter transistor widths $W_n = 8\lambda W_p = 16\lambda$ (widths for other gates were scaled accordingly). The values of V_{dd}, $T_{ox_{Lo}}$, and $T_{ox_{Hi}}$ used in the simulations are 1.2V, 12Å, and 17Å, respectively, at the 100nm node, and 1.0V, 11Å and 17Å, respectively, at the 70nm node.

Tradeoff curves for two representative benchmarks are shown in Figure 8. Curve (I) represents the tradeoff curve with all transistor T_{ox}'s optimized. All curves marked as curve (I) show a knee region that corresponds to a set of good design points. The points to the right of the knee incur a large delay penalty for small reductions in total leakage, while those to the left exhibit large leakage overheads for minor delay benefits. A notable observation is that though I_{gate} of a single PMOS transistor is small, setting all PMOS transistors to $T_{ox_{Lo}}$ incurs a high cumulative expense. This is shown by the curves (II),

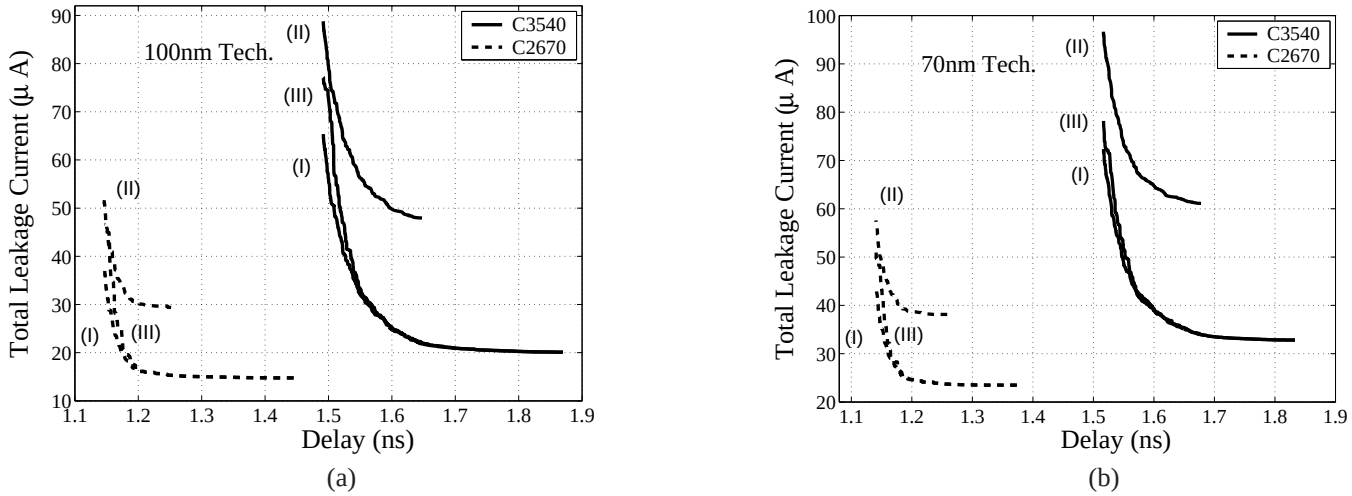


Fig. 8. Leakage/Delay tradeoff curves for C3540 and C2670 at the (a) 100nm and (b) 70nm technology nodes. In (I), all transistor T_{ox} values are optimized, in (II), all PMOS devices fixed at $T_{ox_{Lo}}$ and all NMOS T_{ox} values are optimized, and in (III), the optimization is performed at the stack level, by assigning a single T_{ox} value to an entire stack of transistors.

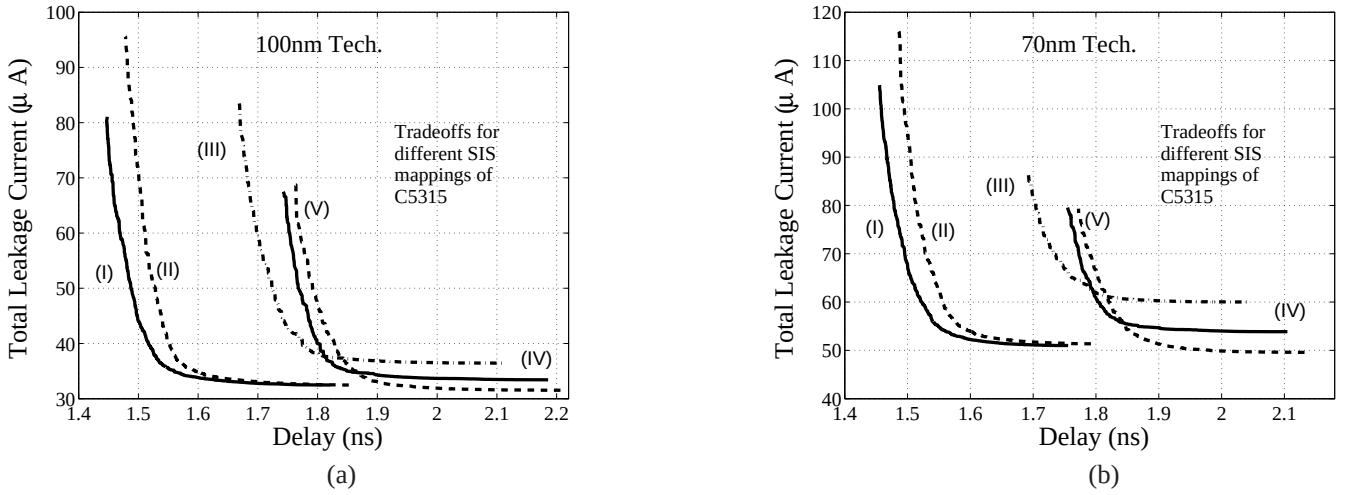


Fig. 9. Leakage/Delay tradeoff curves for C5315 for (a) 100nm, and (b) 70nm technology nodes, for five different circuit structures obtained from SIS [28].

which correspond to a case where all PMOS transistors are set to $T_{ox_{Lo}}$ and the T_{ox} values of only the NMOS devices are optimized. This curve is clearly inferior to the curves (I) that correspond to a full T_{ox} optimization for both NMOS and PMOS transistors.

In each of the possible design choices on tradeoff curves (I) and (II), series-connected devices, i.e., a stack of transistors, can have different T_{ox} values. Design rules that take this into account would increase the spacing between such devices compared to the case where all of the series-connected devices have identical T_{ox}. It is possible that this would lead to a significant increase in total chip area. In order to avoid such area increases, we explored a coarse-grained T_{ox} assignment strategy. If a stack of transistors is on the critical path, we assign all of the transistor to $T_{ox_{Lo}}$ instead of assigning only one transistor in a stack to $T_{ox_{Lo}}$. The tradeoff for this is shown by the curves (III) in Figure 8. Observe that for all points on the right knee region, curve (III) and curve (I) overlap. However, the points to the left of the knee have a small to moderate leakage overhead for the same delay. Hence, if the design choice is only limited to the knee or to points right of the knee, then a coarse-grained T_{ox} assignment would be preferable as it could achieve designs with smaller area than the original strategy of assigning T_{ox} to individual transistors. It should be pointed out that, the percentage of three- and four-input logic gates in all of our benchmark circuits range between 0–17%. Therefore, it is possible that owing to small percentage of large stacked transistors, curves (I) and (III) may have a large overlap. We expect that as the percentage of large stacked transistor increases, this overlap region will not only shrink, but may also lead to higher leakage overhead in curve (III), as compared to curve (I), for the same delay.

There are various techniques to reduce delay of a circuit, such as restructuring and resizing. In order to examine whether dual-T_{ox} approach is consistent with these techniques, leakage/delay tradeoff curves were generated for five different circuit structures for C5315 (using SIS [28] for mapping). Figure 9 shows tradeoff curves obtained at the 100nm and 70nm technology nodes. The results are consistent across different restructured circuits, i.e., for all of the five restructured circuits, our optimization yields a maximum possible delay reduction of about 20% for 100nm node, and about 17% for 70nm node. These results also suggest that dual-T_{ox} approach is orthogonal to other delay optimization approaches, and does

TABLE II

LEAKAGE/DELAY TRADEOFFS FROM DUAL T_{ox} OPTIMIZATION. FOR EACH CIRCUIT, ROW 1 = ALL TRANSISTORS AT $T_{ox_{Hi}}$, ROWS 2 = END RESULTS BASED ON OUR OPTIMIZATION, ROW 3 = ALL TRANSISTORS AT $T_{ox_{Lo}}$, ROW 4 = STARTING FROM "ALL $T_{ox_{Hi}}$" POINT, ALL TRANSISTOR OF CRITICAL PATH LOGIC GATES ARE BLINDLY ASSIGNED TO $T_{ox_{Lo}}$. ROW 2 MATCHES THE DELAY FOR THE "ALL $T_{ox_{Lo}}$" POINT WITH A LEAKAGE SAVINGS OF "%R," AND "%D" IN ROW 1 SHOWS THE DELAY PENALTY OF THE ALL $T_{ox_{Hi}}$ CASE RELATIVE TO THIS POINT. EACH ROW SHOWS I_{gate}, I_{sub} AND I_{total}, AND THE CPU TIME REQUIRED TO GENERATE THE ENTIRE LEAKAGE-DELAY TRADEOFF CURVE IS IN THE LAST COLUMN.

Circuit	100nm Technology					70nm Technology				
	Delay	Leakage Current (μA)			CPU Time	Delay	Leakage Current (μA)			CPU Time
	(ns)(%D)	I_{sub}	I_{gate}	I_{total}(%R)	(s)	(ns)(%D)	I_{sub}	I_{gate}	I_{total}(%R)	(s)
C432	1.38(25.6)	2.83	0.88	3.71	1.7	1.32(20.3)	5.76	0.20	5.96	1.6
	1.10	3.16	30.28	33.44 (75.8)		1.10	5.62	24.03	29.65 (73.4)	
	1.10	3.85	134.16	138.01		1.10	5.09	106.52	111.61	
	1.25	2.99	33.61	36.60		1.22	5.67	26.17	31.84	
C499	1.12(25.0)	7.05	1.49	8.54	12.1	1.09(21.2)	14.30	0.34	14.64	13.8
	0.89	8.09	45.76	53.85 (77.4)		0.90	13.78	54.92	68.70 (64.9)	
	0.89	9.61	229.07	238.68		0.90	12.66	182.84	195.50	
	1.10	7.16	11.92	19.09		1.08	14.24	8.28	22.51	
C880	1.06(25.5)	4.65	1.16	5.81	1.4	1.02(21.0)	9.26	0.26	9.52	1.5
	0.85	4.83	9.67	14.50 (92.2)		0.84	9.18	12.00	21.17 (86.0)	
	0.85	6.31	179.10	185.41		0.84	8.10	143.05	151.15	
	1.01	4.77	16.01	20.78		0.97	9.19	11.89	21.08	
C1355	1.13(24.9)	7.55	1.66	9.22	11.3	1.09(20.3)	15.43	0.38	15.81	13.3
	0.90	8.31	31.97	40.28 (84.8)		0.90	14.83	31.91	46.74 (78.3)	
	0.90	10.08	254.33	264.41		0.90	13.29	202.44	215.73	
	1.13	7.66	12.83	20.49		1.09	15.36	9.29	24.65	
C1908	1.44(25.1)	8.44	1.91	10.35	14.3	1.42(20.9)	16.94	0.43	17.38	14.1
	1.15	9.36	42.62	51.97 (83.1)		1.17	16.53	34.12	50.66 (79.9)	
	1.15	11.47	295.53	307.00		1.17	14.92	236.66	251.58	
	1.41	8.59	14.48	23.07		1.39	16.85	10.63	27.48	
C2670	1.45(26.0)	11.31	3.46	14.77	7.0	1.37(20.4)	22.70	0.78	23.49	6.9
	1.15	11.69	26.48	38.17 (93.0)		1.14	22.45	21.37	43.82 (90.0)	
	1.15	15.24	526.28	541.52		1.14	19.82	418.10	437.92	
	1.33	11.43	23.15	34.58		1.25	22.61	17.03	39.64	
C3540	1.87(25.3)	15.82	4.29	20.11	23.2	1.83(20.9)	31.85	0.97	32.82	21.9
	1.49	16.85	48.55	65.40 (90.4)		1.52	31.57	40.81	72.38 (87.0)	
	1.49	21.61	660.41	682.03		1.52	28.11	527.72	555.83	
	1.81	16.01	31.19	47.20		1.79	31.73	21.21	52.94	
C5315	1.82(25.7)	24.37	8.11	32.48	33.1	1.76(20.7)	49.17	1.84	51.01	35.6
	1.45	25.39	55.69	81.09 (93.6)		1.46	48.77	56.16	104.93 (89.8)	
	1.45	33.04	1234.38	1267.43		1.46	43.21	980.45	1023.67	
	1.79	24.58	35.79	60.37		1.73	49.06	25.37	74.42	
C6288	5.10(25.7)	37.10	8.80	45.90	261.0	5.02(20.7)	75.80	2.00	77.80	273.2
	4.06	41.67	320.22	361.89 (74.0)		4.16	73.48	276.75	350.23 (69.1)	
	4.06	50.29	1340.37	1390.66		4.16	66.65	1065.20	1131.85	
	4.82	37.71	72.80	110.51		4.73	75.41	54.34	129.75	
C7552	2.09(24.8)	36.00	9.71	45.71	30.9	1.93(19.4)	72.70	2.20	74.91	33.5
	1.67	36.55	25.33	61.88 (96.0)		1.62	72.63	18.98	91.61 (92.7)	
	1.67	49.11	1484.59	1533.69		1.62	64.36	1181.86	1246.22	
	2.03	36.09	17.70	53.80		1.86	72.65	9.28	81.93	

not duplicate the benefits obtained from those methods: as shown in Figure 9, curve (I) is superior to curve (V), and curve (I) could be obtained only if the dual-T_{ox} approach is applied along with restructuring. In other words, the dual-T_{ox} technique should be used in combination with other approaches for better delay optimization.

Table II shows leakage/delay tradeoffs for the entire ISCAS85 benchmark suite (except for the 6-gate C17 circuit), including values of I_{sub}, I_{gate}, and I_{total} for various target delays. The all-$T_{ox_{Hi}}$ case typically has a delay penalty of about 25% for the 100nm node and about 20% for the 70nm node compared to the case where all of the critical path transistors are at $T_{ox_{Lo}}$. Similarly, as more and more transistors are assigned to $T_{ox_{Lo}}$, I_{sub} and I_{gate} typically increase, the latter being at a much more rapid rate. The delay corresponding to setting all transistors to $T_{ox_{Lo}}$ is the minimum achievable delay, and can be matched by our optimization with an average reduction, over all circuits, of 86% and 81% in I_{total}, for the 100nm and 70nm nodes respectively. Row 4 for each circuit in Table II shows results for the case where, starting from all transistors assigned to $T_{ox_{Hi}}$, a simple approach is used where all transistors of critical path logic gates are assigned to $T_{ox_{Lo}}$. Further iterations are not performed. Clearly this approach yields only a marginal reduction in delay for significantly high total leakage penalty when compared to the case where all transistors are assigned to $T_{ox_{Lo}}$. This is because of the presence of many near critical paths in the circuits, whose transistors are still at $T_{ox_{Hi}}$.

An insight to these leakage savings can be obtained from slack histograms. Figure 10 shows slack histograms for C3540 at 100nm and 70nm technology node, for the cases where all transistors are set to $T_{ox_{Lo}}$, and for the result of our optimization. Since circuits are mapped for minimum delay, the histograms show a large number of nodes with near-zero slack. However, observe that the histogram for dual-T_{ox}-optimized circuits has a steeper step function-like histogram at slack $\approx$ 0 ns, as compared to the case where all transistors in the circuit are at $T_{ox_{Lo}}$. This highlights the superiority of our optimization, which does not over-optimize path delays, and consequently result in a larger total leakage. The minimum reduction in I_{total} at the tightest delay constraint is 74% for C6288 (100nm) and 64.9% for C499 (70nm).

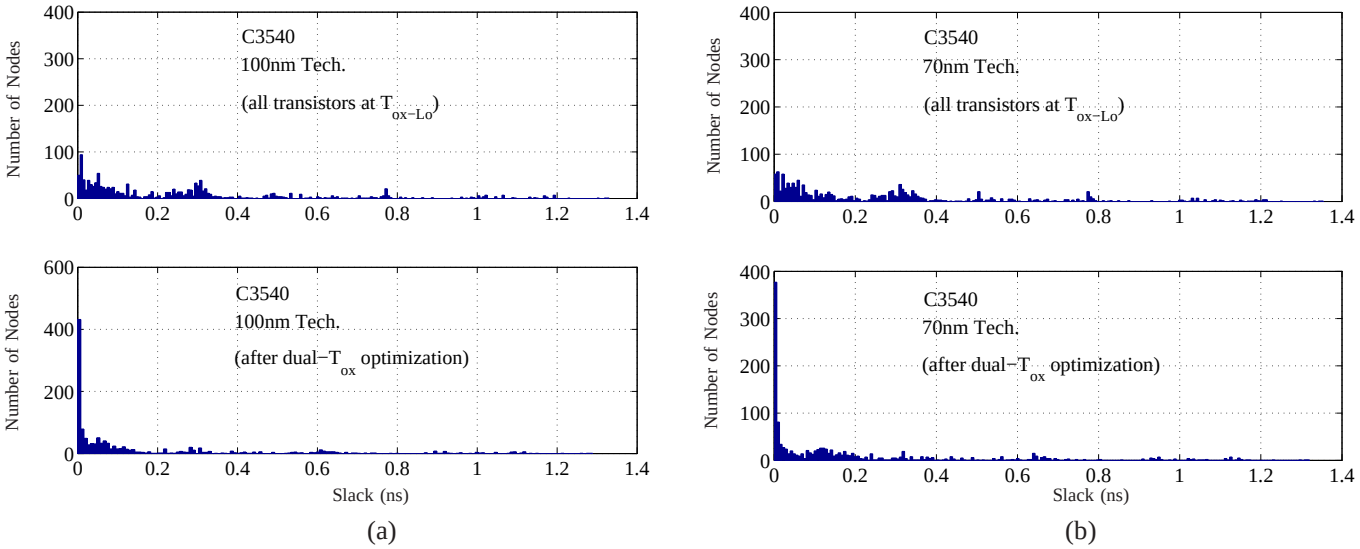


Fig. 10. Slack histograms for C3540 for the (a) 100nm (b) 70nm technology node, for the case where all transistors are set to $T_{ox_{Lo}}$, and after our dual-T_{ox} optimization.

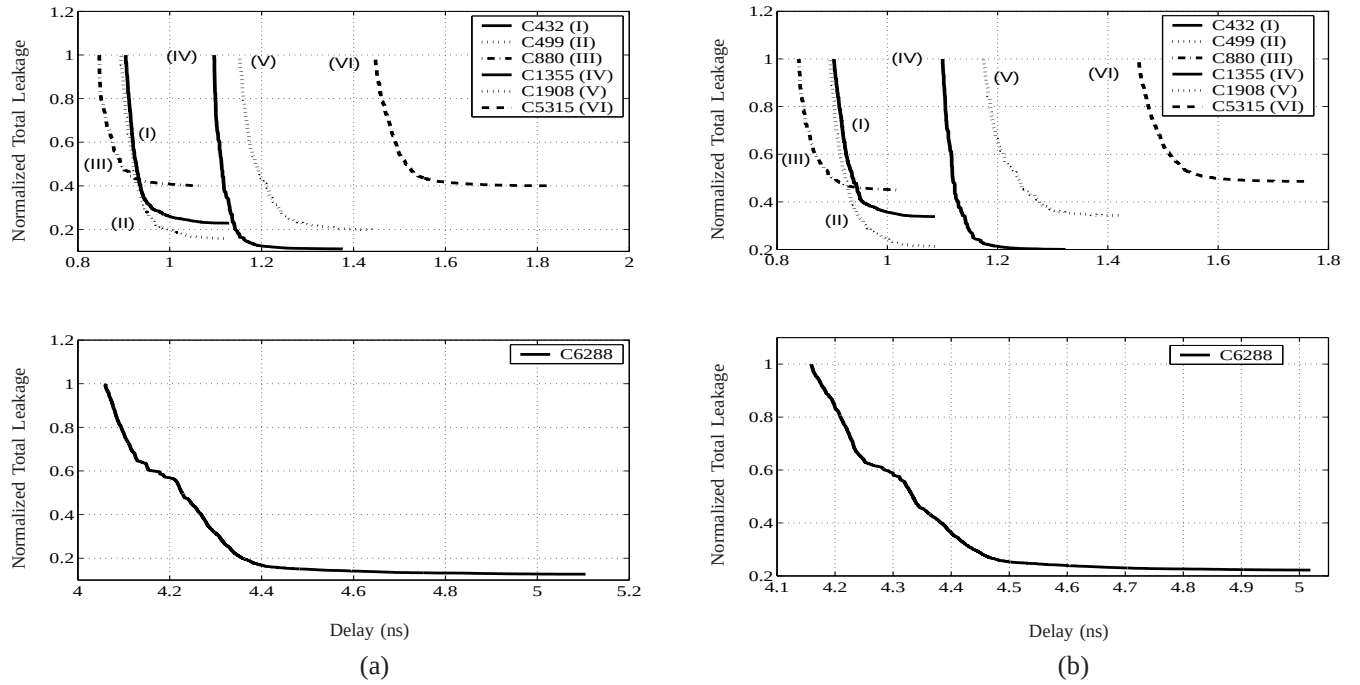


Fig. 11. Normalized Leakage/Delay tradeoff curves for different benchmark circuits for (a) 100nm (b) 70nm technology. The total leakage value at each point on the tradeoff curve has been normalized with respect to the total leakage observed at the end of optimization for each circuit.

Furthermore, in each case of tradeoff curves, the knee point on the curve performs far better than the minimum-delay point. Our optimization technique yields a tradeoff curve that results in a smooth tradeoff starting from all transistors set to $T_{ox_{Hi}}$, leading to increase in the total leakage current and delay reduction that is in the range of about 20% for 100nm and 17% for 70nm node. In order to better represent our results we show tradeoff curves for various benchmark circuits in Figure 11. The total leakage value at each point on the tradeoff curve for all circuits has been normalized with respect to their corresponding total leakage value observed at the end of the optimization.

We now discuss the results obtained after reordering was performed at each delay point on the tradeoff curve. Figure 12 shows experimental results at the 100nm and 70nm technology nodes for two representative benchmark circuits. Each set of results shows the tradeoff curves 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 achieved in I_{gate} are seen to reduce as the target delay reduces (i.e., tighter delay constraints). This can be intuitively explained as follows: as the delay decreases, the number of nodes that lie on critical paths 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.

The value of I_{gate} worsens as one goes to finer transistor

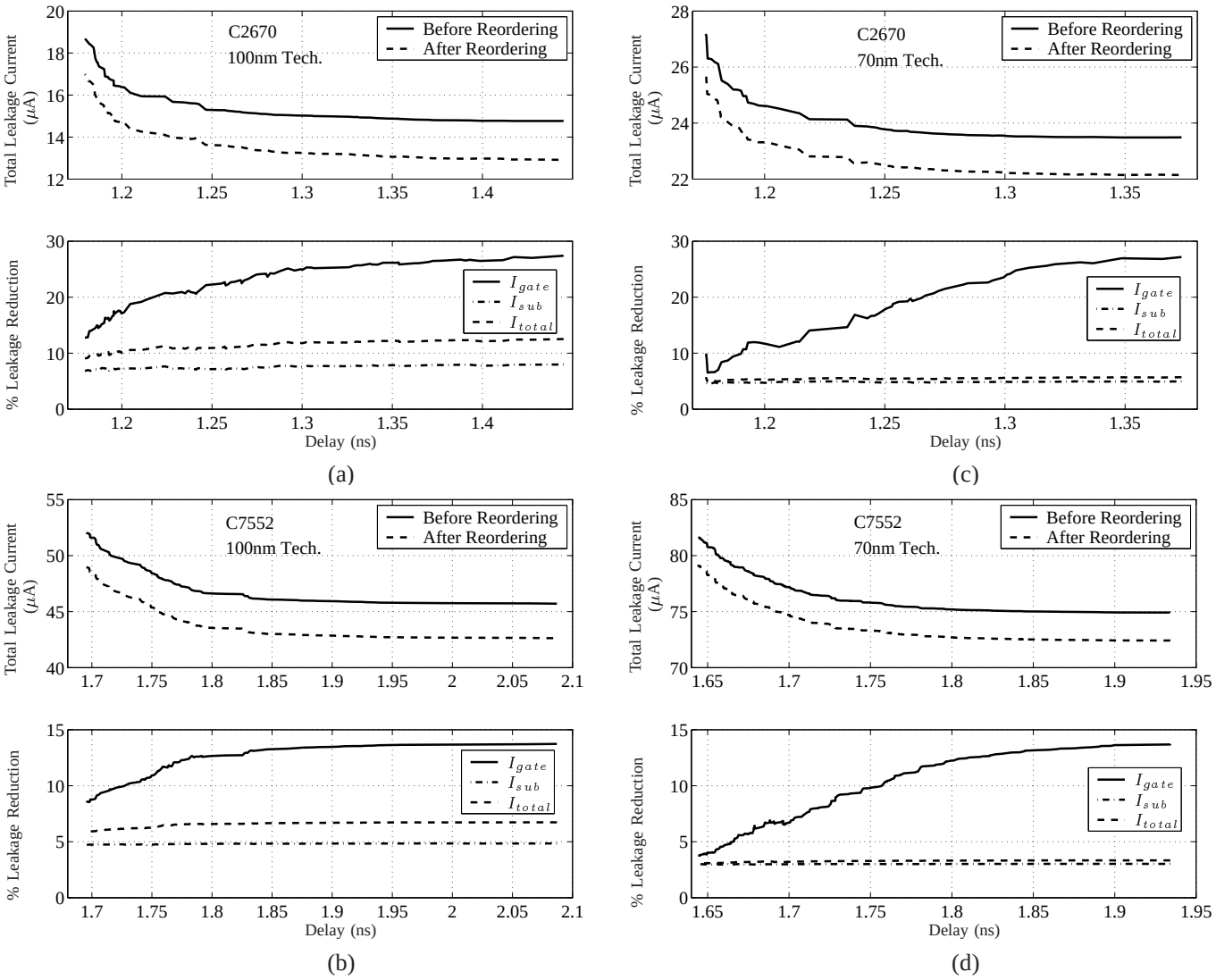


Fig. 12. Leakage/Delay tradeoff curve and percentage leakage reduction for (a) C2670, (b) C7552 for 100nm technology node and (c) C2670, (d) C7552 for 70nm technology node.

geometries due to oxide thickness scaling. Hence one would expect a stronger dominance of I_{sub} in 100nm node and a higher contribution of I_{gate} to I_{total} in 70nm node. In other words, in Figure 12, the curve corresponding to I_{total} should be nearer to I_{sub} for 100nm and closer to I_{gate} for 70nm node. Clearly, this is not true in our case. Furthermore, the leakage/delay tradeoff results, discussed above, show better leakage/delay tradeoffs for the 100nm than for the 70nm technology node. The sole reason for this is the choice of $T_{ox_{Lo}}$ values for the 70nm technology. Although it is desirable to use lower $T_{ox_{Lo}}$ values for a better tradeoff, the choice of a very low $T_{ox_{Lo}}$ would lead to complete dominance of I_{gate} over I_{sub}, which does not correspond to a reasonable process design point. Therefore, as a general rule of thumb, we chose $T_{ox_{Lo}}$ such that the ratio I_{gate}/I_{sub} is reasonable [1], which resulted in the choice of $T_{ox_{Lo}}$ of 12Å for the 100nm, and 11Å for the 70nm technology node. Moreover, we observe that although $T_{ox_{Lo}}$ for 70nm is less than 100nm technology node, the total I_{gate} value at 70nm is less than at 100nm (see Row 2 for each circuit in Table II). This is not counter-intuitive: as T_{ox} reduces, the tunneling current density, J_{tunnel}, increases, but this is counterbalanced by the fact that the effective area ($L_{eff} \times W_{eff}$) decreases. Since I_{gate} is also linearly dependent on the effective area, the net result is a smaller I_{gate} value for the 70nm node, as compared to the 100nm node, for the same circuit. Of course, at finer geometries, the number of transistors that can be packed into the same area is larger, and therefore, one could expect that for circuits of similar area, a 70nm technology would see a larger net I_{gate}.

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 III shows the percentage leakage reduction obtained using transistor and pin reordering at three design points on the leakage/delay tradeoff curve for each circuit. We choose one data point from the knee region (C1) and select 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 about 26% for both the 100nm and 70nm technology nodes. The savings in I_{gate} for C1 is relatively

TABLE III

RESULTS OF TRANSISTOR AND PIN REORDERING, APPLIED TO A SET OF DESIGN POINTS ON THE LEAKAGE/DELAY TRADEOFF CURVE.

Circuit	Percentage Leakage Reduction									CPU Time (sec)
	100nm Technology									
	C1			C2			C3			
	I_{gate}	I_{sub}	I_{total}	I_{gate}	I_{sub}	I_{total}	I_{gate}	I_{sub}	I_{total}	
C432	3.5	3.2	3.3	14.7	5.4	7.8	18.0	5.8	8.7	0.59
C499	4.0	5.1	4.6	9.0	5.1	5.9	11.9	5.3	6.5	0.89
C880	10.5	6.0	7.4	17.3	6.4	8.8	19.8	6.7	9.4	0.39
C1355	3.9	3.3	3.5	7.5	3.6	4.4	9.5	3.8	4.8	0.82
C1908	4.9	3.5	4.0	8.3	3.6	4.7	10.8	3.7	5.1	1.20
C2670	17.1	7.1	10.0	25.0	7.6	11.8	26.5	7.8	12.1	1.24
C3540	8.7	5.3	6.5	13.7	5.6	7.4	15.2	5.6	7.6	3.12
C5315	11.3	6.2	8.1	19.3	6.4	9.7	20.3	6.5	9.9	3.55
C6288	2.9	2.9	2.9	4.1	2.9	3.3	6.7	3.1	3.8	19.45
C7552	10.1	4.8	6.2	13.3	4.8	6.7	13.7	4.8	6.7	3.73
	70nm Technology									
C432	3.6	3.1	3.1	12.4	3.3	3.7	18.1	3.5	4.0	0.66
C499	2.1	3.1	2.7	3.6	3.2	3.3	9.4	3.3	3.5	1.01
C880	7.1	4.6	4.8	12.0	4.6	5.0	17.4	4.9	5.3	0.42
C1355	1.4	2.2	2.1	2.7	2.1	2.2	8.1	2.3	2.4	0.89
C1908	1.5	2.7	2.4	5.9	2.9	3.0	10.8	2.9	3.1	1.32
C2670	11.8	4.7	5.3	20.4	4.8	5.4	27.0	5.0	5.7	1.30
C3540	2.7	4.3	4.0	10.7	4.4	4.7	15.2	4.5	4.8	2.64
C5315	6.7	4.1	4.4	16.7	4.1	4.6	20.6	4.1	4.7	3.40
C6288	2.0	1.9	1.9	4.0	1.9	2.0	5.5	1.9	2.0	20.48
C7552	6.7	3.0	3.2	12.2	3.0	3.3	13.6	3.0	3.3	3.50

lower, with maximum reductions of 17% and 11% for the 100nm and 70nm nodes, respectively, and the reasons for this are described above. The reduction in I_{sub} is under 7% and is practically constant for all 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, only requiring a few seconds. Observe that transistor reordering is not performed for the case of coarse-grained T_{ox} assignment (see Figure 8 curve (III)) as all of the transistors in a stack are assigned to either $T_{ox_{Lo}}$ or $T_{ox_{Hi}}$. Hence, the reordering search space is significantly reduced and so we do not perform reordering on this coarse-grained tradeoff curve.

The table also shows the reductions in total leakage, which are seen to be up to 12.0% (for point C3 of C2670). Although these are not startlingly dramatic numbers, they still correspond to solid reductions in the total leakage with no delay penalties. 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.

IX. CONCLUSION

We have presented a technique for reducing the total active mode leakage current, including gate oxide leakage, by determining appropriate values of T_{ox}, and iteratively assigning them to individual transistors in the circuit. Our approach provides the complete tradeoff curve between leakage and delay, and achieves delay reductions of 20% and 17% for predictive 100nm and 70nm technologies, respectively.

Furthermore, complex gates with series-connected devices show some flexibility in varying the relative ordering of the pins and transistors. We have presented a simple transistor and pin reordering technique that exploits this design space for reducing the total active leakage in dual T_{ox} circuits. A major advantage of this optimization is its low impact on layout. It has been shown that this optimization results in an overall leakage reduction of up to 12.0%, and a reduction in gate leakage of up to 26.0% with no delay penalties while the optimization requires under 25 seconds on all benchmarks.

In this work, we have shown a technique for computing I_{total} by estimating $I_{sub,avg}$ and $I_{gate,avg}$ individually. This approach is based on the concept of dominant states with the assumption that EDT in the ON state of the device is negligible. While we are aware of commercial technologies where this assumption is valid, this may not be true of all devices in the future. In such a case, the $I_{gate,avg}$ in the on state can still be estimated using a similar calculation that sums up its gate-to-channel and EDT currents, invoking the dominant states. Effectively, this implies that the constant used to express the gate leakage per unit width is changed.

The results in this work are based on a heuristic approach, and there is room for the use of more sophisticated algorithmic methods to be applied to this problem in future work.

REFERENCES

[1] Semiconductor Industry Association, "International Technology Roadmap for Semiconductors," 2003. Available at `http://public.itrs.net`.

[2] F. Hamzaoglu and M. R. Stan, "Circuit-Level Techniques to Control Gate Leakage for Sub-100 nm CMOS," in *Proceedings of International Symposium on Low Power Electronics and Design*, pp. 60–63, Aug. 2002.

[3] M. Hirose, M. Koh, W. Mizubayashi, H. Murakami, K. Shibahara, and S. Miyazaki, "Fundamental Limit of Gate Oxide Thickness Scaling in Advanced MOSFETs," *Semiconductor Science and Technology*, vol. 15(5), pp. 485–490, May 2000.

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

[5] J. Kao, A. Chandrakasan, and D. Antoniadis, "Transistor Sizing Issues and Tool for Multi-Threshold CMOS Technology," in *Proceedings of ACM/IEEE Design Automation Conference*, pp. 409–414, Jun. 1997.

[6] Y. Oowaki, M. Noguchi, S. Takagi, D. Takashima, M. Ono, Y. Matsunaga, *et al.*, "A sub-0.1 μm Circuit Design with Substrate-Over-Biasing," in *IEEE International Solid-State Circuits Conference Digest of Techinal Papers*, pp. 88–89, Feb. 1998.

[7] D. Lee, H. Deogun, D. Blaauw, and D. Sylvester, "Simultaneous State, V_t and T_{ox} Assignment for Total Standby Power Minimization," in *Proceedings of ACM/IEEE Design, Automation and Test in Europe*, pp. 494–499, Feb. 2004.

[8] S. Narendra, D. Blaauw, A. Devgan, and F. Najm, "Leakage Issues in IC design: Trends, Estimation, and Avoidance." Tutorial at ACM/IEEE International Conference on Computer Aided Design, Nov. 2003.

[9] A. Sultania, D. Sylvester, and S. S. Sapatnekar, "Tradeoffs between Gate Oxide Leakage and Delay for Dual T_{ox} Circuits," in *Proceedings of ACM/IEEE Design Automation Conference*, pp. 761–766, June 2004.

[10] A. Sultania, D. Sylvester, and S. S. Sapatnekar, "Transistor and Pin Reordering for Gate Oxide Leakage Reduction in Dual T_{ox} Circuits," in *Proceedings of IEEE International Conference on Computer Design*, pp. 228–233, Oct. 2004.

[11] C.-H. Choi, Z. Yu, and R. W. Dutton, "Impact of Gate Direct Tunneling on Circuit Performace: A Simulation Study," *IEEE Transactions on Electron Devices*, pp. 2823–2829, Dec. 2001.

[12] N. Sirisantana, L. Wei, and K. Roy, "High-Performace Low-Power CMOS Circuits Using Multiple Channel Length and Multiple Oxide Thickness," in *Proceedings of IEEE International Conference on Computer Design*, pp. 227–232, Sept. 2000.

[13] R. Hossain, M. Zheng, and A. Albicki, "Reducing Power Dissipation in CMOS Circuits by Signal Probability Based Transistor Reordering," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 15(3), pp. 361–368, Mar. 1996.

[14] E. Musoll and J. Cortadella, "Optimizing CMOS Circuits for Low Power using Transistor Reordering," in *Proceedings of European Design and Test Conference*, pp. 219–223, Mar. 1996.

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

[16] D. Lee, W. Cong, D. Blaauw, and D. Sylvester, "Analysis and Minimization Techniques for Total Leakage Considering Gate Oxide Leakage," in *Proceedings of ACM/IEEE Design Automation Conference*, pp. 175–180, Jun. 2003.

[17] K. Bernstein, "Private Communication." IBM T. J. Watson Research Center, Yorktown Heights, NY, 2003.

[18] Y. Taur, "CMOS Design Near the Limits of Scaling," *IBM Journal of Research and Development*, vol. 46(2/3), pp. 213–222, Mar./May 2002.

[19] K. Chen, C. Hu, P. Fang, M. R. Lin, and D. L. Wollensen, "Predicting CMOS Speed with Gate Oxide and Voltage Scaling and Interconnect Loading Effects," *IEEE Transactions on Electron Devices*, vol. 44(11), pp. 1951–1957, Nov. 1997.

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

[21] S. Sirichotiyakul, T. Edwards, O. Chanhee, R. Panda, and D. Blaauw, "Duet: An Accurate Leakage Estimation and Optimization Tool for Dual-V_t Circuits," *IEEE Transactions on Very Large Scale Integration Systems*, vol. 10(2), pp. 79–90, Apr. 2002.

[22] A. Chandrakasan, W. J. Bowhill, and F. Fox, *Design of High-Performance Microprocessor Circuits*. Piscataway, NJ: IEEE Press, 2001.

[23] M. Dražd̄iulis and P. Larsson-Edefors, "A Gate Leakage Reduction Strategy for Future CMOS Circuits," in *Proceedings of European Solid-State Circuits Conference*, pp. 317–320, Sept. 2003.

[24] W. Henson, N. Yang, S. Kubicek, E. M. Vogel, J. J. Wortman, K. D. Meyer, and A. Naem, "Analysis of Leakage Currents and Impact on Off-State Power Consumption for CMOS Technology in the 100-nm Regime," *IEEE Transactions on Electron Devices*, vol. 47(7), pp. 1393–1400, July 2000.

[25] K. A. Bowman, L. Wang, X. Tang, and J. D. Meindl, "A Circuit-Level Perspective of the Optimum Gate Oxide Thickness," *IEEE Transactions on Electron Devices*, vol. 48(8), pp. 1800–1810, Aug. 2001.

[26] J. Fishburn and A. Dunlop, "TILOS: A Posynomial Programming Approach to Transistor Sizing," in *Proceedings of ACM/IEEE International Conference on Computer Aided Design*, pp. 326–328, Nov. 1985.

[27] F. Brglez and H. Fujiwara, "A Neutral Netlist of 10 Combinatorial Benchmark Circuits," in *Proceedings of IEEE International Symposium on Circuits and Systems*, pp. 695–698, Jun. 1985.

[28] E. M. Sentovich, K. J. Singh, L. Lavagno, C. Moon, R. Murgai, A. Saldanha, *et al.*, "SIS: A System for Sequential Circuit Synthesis," Tech. Rep. UCB/ERL M92/41, Electronics Research Laboratory, Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, May 1992.

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

[30] J. Hu and S. Sapatnekar, "A Timing-Constrained Simultaneous Global Routing Algorithm," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 21, pp. 1025–1036, Sept. 2002.

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

Anup Kumar Sultania received the B.E. degree in electrical engineering from Birla Institute of Technology and Science, Pilani in 2002, the M.S. degree in electrical engineering from University of Minnesota, Twin-Cities in 2004. He is currently working in Calypto Design System, Inc., Santa Clara, CA. He has previously worked as an intern for six months at ST Microelectronics, India. His present research interests are power analysis and optimization.

Dennis Sylvester (S '95, M '00, SM '04) received the B.S. degree in electrical engineering summa cum laude from the University of Michigan, Ann Arbor, in 1995. He received the M.S. and Ph.D. degrees in electrical engineering from University of California, Berkeley, in 1997 and 1999, respectively. His dissertation research was recognized with the 2000 David J. Sakrison Memorial Prize as the most outstanding research in the UC-Berkeley EECS department.

He is now an Associate Professor of Electrical Engineering at the University of Michigan, Ann Arbor. He previously held research staff positions in the Advanced Technology Group of Synopsys, Mountain View, CA, and at Hewlett-Packard Laboratories in Palo Alto, CA. He has published numerous articles along with one book and several book chapters in his field of research, which includes low-power circuit design and design automation techniques, design-for-manufacturability, and on-chip interconnect modeling. He also serves as a consultant and technical advisory board member for several electronic design automation firms in these areas.

Dr. Sylvester received an NSF CAREER award, the 2000 Beatrice Winner Award at ISSCC, a 2004 IBM Faculty Award, and several best paper awards and nominations. He is the recipient of the ACM SIGDA Outstanding New Faculty Award, the 1938E Award from the College of Engineering Award for teaching and mentoring, and the Henry Russel Award, which is the highest award given to faculty at the University of Michigan. He has served on the technical program committee of numerous design automation and circuit design conferences and was general chair of the 2003 ACM/IEEE System-Level Interconnect Prediction (SLIP) Workshop and 2005 ACM/IEEE Workshop on Timing Issues in the Synthesis and Specification of Digital Systems (TAU). He is currently an Associate Editor for IEEE Transactions on VLSI Systems. He also helped define the circuit and physical design roadmap as a member of the International Technology Roadmap for Semiconductors (ITRS) U.S. Design Technology Working Group from 2001 to 2003. He is a member of ACM, American Society of Engineering Education, and Eta Kappa Nu.

Sachin Suresh Sapatnekar received the B.Tech. degree from the Indian Institute of Technology, Bombay in 1987, the M.S. degree from Syracuse University in 1989, and the Ph.D. degree from the University of Illinois at Urbana-Champaign in 1992. From 1992 to 1997, he was an assistant professor in the Department of Electrical and Computer Engineering at Iowa State University. He is currently the Robert and Marjorie Henle Professor in the Department of Electrical and Computer Engineering at the University of Minnesota.

He has authored several books and papers in the areas of timing and layout. He has held positions on the editorial board of the IEEE Transactions on VLSI Systems, and the IEEE Transactions on Circuits and Systems II, IEEE Design and Test, and the IEEE Transactions on CAD. He has served on the Technical Program Committee for various conferences, and as Technical Program and General Chair for Tau and ISPD, and Techical Program co-chair for DAC. He has been a Distinguished Visitor for the IEEE Computer Society and a Distinguished Lecturer for the IEEE Circuits and Systems Society. He is a recipient of the NSF Career Award, three best paper awards at DAC and one at ICCD, and the SRC Technical Excellence award. He is a fellow of the IEEE.