Analysis and Minimization Techniques for Total Leakage Considering Gate Oxide Leakage

Dongwoo Lee, Wesley Kwong, David Blaauw, Dennis Sylvester
University of Michigan, Ann Arbor, MI

Abstract

In this paper we address the growing issue of gate oxide leakage current (I_{gate}) at the circuit level. Specifically, we develop a fast approach to analyze the total leakage power of a large circuit block, considering both I_{gate} and subthreshold leakage (I_{sub}). The interaction between I_{sub} and I_{gate} complicates analysis in arbitrary CMOS topologies and we propose simple and accurate heuristics based on table look-ups to quickly estimate the state-dependent total leakage current within 1% of SPICE. We then make several observations on the impact of I_{gate} in designs that are standby power limited, including the role of device ordering within a stack and the differing state dependencies for NOR vs. NAND topologies. Based on these observations, we propose the use of pin reordering as a means to reduce I_{gate} due to the dependence of gate leakage on stack node voltages.

Categories and Subject Descriptors

B.8.2 [Performance and Reliability]: Performance analysis

General Terms

Algorithms, performance, design, reliability

1 Introduction

Feature size reduction in MOSFETs has been the key enabler to the continuation of Moore's law. Just as significant as channel length (L_{eff}) reduction has been the shrinking of the gate oxide layer thickness (T_{ox}). Early indications of 90nm CMOS technologies set to come online in 2003 show T_{ox} values in the range of 12-16 Angstroms (1.2-1.6nm) [1][2]. While aggressive scaling of T_{ox} is required to provide large current drive at reduced voltage supplies and to suppress short-channel effects, such as drain-induced barrier lowering (DIBL), it results in the presence of significant gate tunneling leakage current (I_{gate}).

I_{gate} arises due to the finite (non-zero) probability of an electron directly tunneling through the insulating SiO_2 layer. The probability, and hence I_{gate} itself, is a strong exponential function of T_{ox} as well as a function of the voltage potential across the gate oxide. A difference in T_{ox} of just 2 Angstroms (Å) can lead to an order of magnitude change in I_{gate}, making it the most sensitive device performance parameter with respect to any physical dimension. Another key point is that I_{gate} for a PMOS device is typically one order of magnitude smaller than an NMOS device with identical T_{ox} and V_{dd} when using SiO_2 [3]. This is due to the much higher energy required for hole tunneling in SiO_2. However, in alternate dielectric materials the energy required for electron and hole tunneling can be completely different. In the case of nitrided gate oxides, in use today in some processes, PMOS I_{gate} can actually exceed NMOS I_{gate} for higher nitrogen concentrations [4].

For $T_{ox} > 20$Å, I_{gate} is typically very small in comparison to other forms of leakage current. In recent generations, I_{sub} has been seen to rise by a factor of 3 to 5X per generation under normal scaling theory. On the other hand, T_{ox} is 30% thinner in each new process technology and with an initial T_{ox} of 20Å, this results in a 1000X rise in I_{gate} in a subsequent process with T_{ox} of 14Å (it will be somewhat smaller due to a V_{dd} reduction). It is clear that I_{gate} either will, or in some cases already has, caught up to I_{sub} in magnitude. An example is NECs 100nm process with T_{ox} = 16Å [1]. High-V_{th} (mid-performance) devices exhibit an I_{sub} of 0.3nA/µm of gate width. NMOS I_{gate} for this process is 0.65nA/µm with 1V on the gate, exceeding I_{sub}. This NEC process uses a nitrided gate oxide (also called oxynitride) that raises the dielectric constant of the gate insulator from 3.9 to ~4.1-4.2 and can yield an order of magnitude reduction in I_{gate} for the same C_{ox} value. More aggressive high-k materials, such as hafnium oxide (HfO_2), provide dielectric constants in the range of 25-50 and will greatly diminish the significance of I_{gate}. However, there are numerous process integration problems with such high-k materials. As a result, the introduction of true high-k materials (beyond oxynitride) is not expected before the 65nm node in 2007 [2].

There has been extensive work in the analysis and minimization of I_{sub} as it poses a fundamental scaling limit to traditional CMOS design. However, I_{gate} has been growing at a much faster rate and to this point has almost solely received attention from device engineers and not circuit designers and EDA tool developers. In [5] and [6], the authors examined the impact of gate leakage on circuit functionality but did not address its contribution to leakage power. In [7], the authors contribute the first circuit design concepts to reducing the impact of gate leakage – these focus on leveraging the lower I_{gate} in PMOS devices by using p-type domino circuits rather than n-type.

Circuit level analysis of I_{gate} is complicated by two important factors: 1) state dependency and 2) the interaction of I_{sub} and I_{gate}. The state dependence of I_{sub} is fairly well understood, especially in the context of the stack effect. However, there are different considerations with gate tunneling current since conducting devices are most responsible for I_{gate} in contrast to I_{sub}. Furthermore, total leakage current is not always the sum of I_{sub} and I_{gate}. In some states the currents interact at internal nodes, altering the node voltages and complicating the analysis. In this paper, we make two primary contributions. First is the development of a fast approach for total leakage power analysis that considers both I_{gate} and I_{sub}. We consider the interaction between these two sources of current and make several observations about the nature of the standby current problem when I_{gate} is no longer negligible. We categorize the state dependence of a transistor stack into cases where 1) only I_{sub} or I_{gate} occurs, 2) I_{sub} and I_{gate} sum, and 3) I_{sub} and I_{gate} interact in a complex fashion. We partition these cases based on the on/off states of devices within a stack. We then build precharacterized tables for individual device I_{gate} and I_{sub} currents, apply our state dependence heuristic, and compute the total leakage current on a gate by gate basis. The second contribution is the proposal of pin reordering as a new method for reducing I_{gate}. While pin reordering is relatively ineffective for I_{sub}, we utilize the dependence of I_{gate} on the node voltages in the stack and show that I_{gate} can be significantly reduced

DAC 2003, June 2-6, 2003, Anaheim, California, USA.

by placing transistors that are off at the bottom of the stack. We then demonstrate how this method can be combined with state assignment targeted at reducing I_{sub} during standby mode, as well as for runtime reduction of I_{gate}.

2 Efficient Leakage Analysis Method

An empirical gate leakage model was incorporated in a 100nm BSIM3v3 (level 49) model generated using the Berkeley Predictive Technology Model (BPTM) [8]. The gate leakage was modeled using voltage dependent current sources from gate to source (I_{gs}) and gate to drain (I_{gd}), depending on, respectively, V_{gs} and V_{gd}. The model was based on an empirical model of total gate leakage fit to IBM data on thin SiO_2 dielectrics that was used in the 2001 ITRS. This model was then adjusted by fitting the data from an industrial 0.13 µm process over the full range of V_{ds} and V_{gs}, described in more detail in [9].

Two 100nm technology files were generated to study the impact of I_{gate} on circuit behavior - the first has a T_{ox} of 17Å and L_{eff} of 50 nm, while the second has a T_{ox} of 15 Å and L_{eff} = 60 nm. V_{th} in both technologies is approximately 200mV. In the 17Å process I_{gate} is roughly 1/9 of I_{sub} under worst-case biasing conditions while in the 15Å process I_{gate}/I_{sub} = 2/3. I_{sub} values are in the range of 20-40nA/µm of gate width at room temperature which is slightly below the ITRS projected value of 70nA/µm at 100nm. V_{DD} is 1V for both cases and all results are for room temperature.

Standby current estimation is complicated by the state dependence of both the I_{gate} and I_{sub} currents. The state dependence of subthreshold leakage current has been extensively studied and exhibits the so-called *stack effect.* Similarly, gate tunneling current has state dependence due to its strong dependence on the V_{gs} and V_{gd} of a device. For simplicity, we ignore the PMOS I_{gate} since it is approximately one order of magnitude lower than I_{gate} of NMOS devices [3]. However, our analysis method can be easily extended to include PMOS-based I_{gate}, as necessary for nitrided oxides.

To examine the state dependence of I_{gate}, we first consider a simple inverter shown in Figure 1. The maximum gate tunneling current occurs when the input is at V_{dd} and V_s = V_d = 0V for the NMOS device. In this case, V_{gs} = V_{gd} = V_{dd} and the I_{gate} is at its maximum. At the same time, the PMOS device exhibits subthreshold leakage current. If the input voltage is decreased, I_{gs} decreases rapidly and is reduced by more than 1 order of magnitude when V_{gs} = $V_{th,nmos}$, and becomes zero when V_{gs} = 0. However, at the same time V_{gd} will become negative as the output node pulls up, resulting in a reverse gate tunneling current from the drain to the gate node. In this case, tunneling is restricted to the gate-to-drain overlap region, due to the absence of a channel. Since this overlap region is substantially smaller than the channel region, reverse tunneling current is much smaller than the forward tunneling current, and hence can be ignored [10]. In addition, the oxide thickness of the overlap region can be increased by oxidizing the polysilicon after gate formation which would further suppress reverse tunneling in the overlap regions [11].

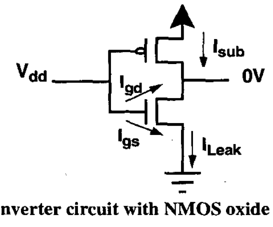

Figure 1. Inverter circuit with NMOS oxide leakage current.

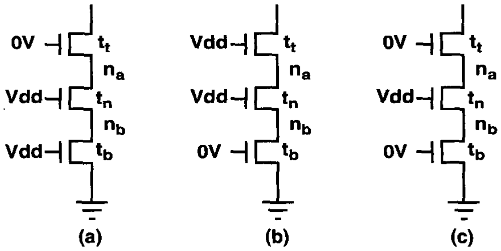

Figure 2. Three input NMOS-stack with three scenarios of combined I_{sub} and I_{gate}.

For a simple inverter with a high input state, the PMOS I_{sub} combines with the NMOS I_{gate} and each can be computed independently and then added to obtain the total leakage current I_{leak}, as shown in Figure 1. Note that the I_{gate} component of I_{leak} is being drawn from the power supply of the previous stage. For a low input state, the NMOS transistor is off and the total leakage current is equal to I_{sub} through the NMOS device.

We next consider a multi-input gate with an NMOS transistor stack. If all inputs have a high state, the analysis is again similar to that of the inverter. The total standby current is equal to the sum of I_{sub} through the PMOS transistors and I_{gate} through the NMOS transistors. However, for input states where at least one input is low and the gate output is high, I_{sub} through turned-off transistors and I_{gate} through turned-on transistors combine at internal stack nodes. I_{sub} and I_{gate} are therefore interdependent in these cases, and must be analyzed simultaneously.

We consider gate tunneling current in three distinct scenarios for a transistor in a transistor stack, as shown in Figure 2. We consider the gate tunneling current through the transistor labeled t_n, with a high gate input state. The complementary PMOS transistors are omitted for clarity. We now discuss each scenario in more detail:

1. Transistor t_n is positioned above zero or more conducting transistors and below one or more nonconducting transistors, Figure 2(a). In this case, the internal nodes n_a and n_b have a conducting path to the ground node and are at nominal 0V. The I_{gate} of transistor t_n therefore does not affect the voltage at nodes n_a and n_b and is added to the I_{sub} of the stack to obtain the total leakage.
2. Transistor t_n is positioned above one or more nonconducting transistors and below zero or more conducting transistors, Figure 2(b). In this case, nodes n_a and n_b are connected to the output of the logic gate through conducting NMOS transistors and will be held at V_{dd} - $V_{th,nmos}$ (with body effect). For transistor t_n, $V_{gs,n}$ and $V_{gd,n}$ are thus small; approximately one threshold voltage. Based on SPICE simulations, I_{gate} in this case is over one order of magnitude smaller than in scenario 1 and can be safely ignored.
3. There is at least one nonconducting transistor *both* above and below transistor t_n in the stack, Figure 2(c). In this case, the I_{sub} current exhibits the stack effect and the internal nodes n_a and n_b have a voltage in the range of 100-200mV. The top transistor t_t is therefore strongly turned off due to its negative $V_{gs,t}$. However, since $V_{gs,n}$ and $V_{gd,n}$ for transistor t_n are only slightly diminished from V_{dd}, t_n will exhibit significant I_{gate} current. This current combines with the I_{sub} through t_t and causes the node voltages at n_a, n_b to increase from their value with only subthreshold current.

A rise in the voltage at n_a and n_b reduces $I_{sub,t}$ through t_t, as $V_{gs,t}$ becomes further negative, and also reduces $I_{gate,n}$ through t_n. However, the dependence of $I_{sub,t}$ on $V_{gs,t}$ is exponential and is much stronger than the dependence of gate tunneling current on

Table 1. Simulation results for individual and combined I_{gate}/I_{sub}.

	17Å			15Å		
	I_{sub} only	I_{gate} only	combined	I_{sub} only	I_{gate} only	combined
V_{na}/V_{nb}	68mV	95mV	111mV	51mV	285mV	285mV
I_{sub}	399pA	-	65pA	693pA	-	32fA
I_{gate}	-	446pA	407pA	-	1.27nA	1.27nA
I_{leak}	399pA	446pA	472pA	693pA	1.27nA	1.27nA

$V_{gs,n}$ and $V_{gd,n}$.[1] Therefore, as the voltage of n_a is raised by I_{gate}, the I_{sub} is diminished by a nearly equal amount. The gate tunneling current therefore effectively displaces the subthreshold current, leaving the total leakage current relatively unchanged. When $I_{gate,n}$ becomes sufficiently large and exceeds the original $I_{sub,t}$, the $I_{sub,t}$ is effectively pinched off and becomes negligible. In this case, the total leakage current is equal to $I_{gate,n}$.

This effect is illustrated in Table 1, where we show the node voltage of n_a, n_b and the leakage currents for the circuit shown in Figure 2(c) for three SPICE simulations: when only I_{sub} is present, when only I_{gate} is present, and when both are present. For the 17Å process, the voltages at n_a and n_b increase by 43mV over the case with I_{sub} only when considering both I_{sub} and I_{gate}, resulting in a decrease of I_{sub} by a factor of 6. However, the voltages at n_a and n_b rise by only 16 mV when the analysis is expanded from only I_{gate} to I_{gate} and I_{sub}, resulting in a decrease of I_{gate} through t_n by just 9%. Table 1 also shows SPICE results for the 15Å process. In this case, I_{sub} is reduced by 4 orders of magnitude, and becomes negligible.

As a result, the total leakage with both I_{sub} and I_{gate} present is nearly equal to the maximum of I_{gate} and I_{sub}, when they are computed independently. In this scenario, we therefore find the total leakage current by computing I_{gate} and I_{sub} separately and set the total leakage current to their maximum.

Based on the three scenarios, we propose the following simple table-based leakage estimation method for arbitrary gate structures. First, we determine the subthreshold leakage current of the circuit, without consideration of gate tunneling current. A number of approximate analytical solutions have been proposed for this purpose [12] and may be used. In this paper, we use an empirical model in which the total subthreshold leakage current is expressed as follows:

$$I_{sub,k} = I_{sub,1} * S_k * s_t, \qquad \text{(EQ 1)}$$

where $I_{sub,1}$ is the leakage current for a single off-transistor of unit size, S_k is the stack factor for a stack with k off-transistors in series and s_t is the size of the transistor. Both $I_{sub,1}$ and S_k are precharacterized using SPICE for stacks with different size transistors and stored in a table.

Next, we measure I_{gate} for a single transistor of unit-size in each of the three discussed scenarios when I_{sub} is eliminated. In scenario 3, the I_{gate} current is dependent on the number of off-transistors below transistor t_n. We therefore specify the gate tunneling current as $I_{gate,l}$, where l indicates the number of off-transistors below t_n, and characterize $I_{gate,l}$ for different value of l in a table. Note that the current $I_{gate,0}$ corresponds to the gate tunneling current in scenario 1.

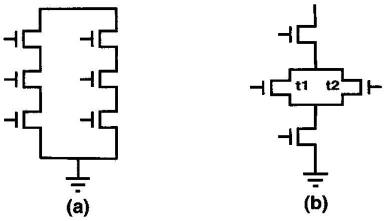

Figure 3. Leakage current computation for series/parallel structures.

The total leakage current, as well as its I_{gate} and I_{sub} components, are then computed as follows. First, the total number of off-transistors in the stack is determined and the I_{sub}, in the absence of I_{gate}, is found using EQ1. Next, the tunneling currents $I_{gate,l}$ of the on-transistors in scenarios 1 and 3 are determined based on precharacterized table values and are multiplied by their transistor size. The total leakage current I_{total}, and its tunneling and subthreshold components I_{gate} and I_{sub}, are then determined as follows:

$$I_{total} = \sum_{l=0} I_{gate,l} + Max\left(\sum_{l>0} I_{gate,l}, I_{sub,k}\right) \qquad \text{(EQ 2)}$$

$$I_{gate} = \sum_{l} I_{gate,l} \qquad \text{(EQ 3)}$$

$$I_{sub} = \begin{matrix} I_{sub,k} - \sum_{l>0} I_{gate,l} & if\left(I_{sub,k} > \sum_{l>0} I_{gate,l}\right) \\ 0 & \text{otherwise} \end{matrix} \qquad \text{(EQ 4)}$$

The first term in EQ2 corresponds to the I_{gate} current of transistors in scenario 1, which is independent of the other currents in the stack. The second term of EQ2 corresponds to the I_{gate} of transistors in scenario 3 which displaces the I_{sub} of the stack. Hence, the current for this term is the maximum of these two currents. EQ3 and EQ4 express the total I_{sub} and I_{gate} in the transistor stack.

For the analysis of series/parallel NMOS structures, such as AOI and OAI gates, we use the following rules to compute the total leakage current. Given multiple parallel transistor stacks, such as those shown for the AOI stacks in Figure 3(a), we compute the leakage current of each stack separately and then add them to obtain the total leakage of the gate. For parallel transistors within an NMOS stack, such as transistors t_1 and t_2 for the OAI gate in Figure 3(b), we first collapse the two parallel transistors using the following rules:

1. If the two parallel transistors t_1 and t_2 have the same gate input state, they are replaced with a single transistor with transistor size equal to the sum of their sizes.
2. If the two parallel transistors t_1 and t_2 have different input states, the off-transistor impacts neither I_{gate} nor I_{sub} and is neglected.

To demonstrate the accuracy of the proposed leakage estimation method, we show the analysis results for a 3-input NAND gate under all possible input states in Tables 2 and 3 for both 15Å and 17Å gate oxide thicknesses. The leakage current obtained from SPICE simulation using the proposed analysis method is also shown and has an average error of 1.2% over all input states. The maximum error occurs for state 110 with 17Å gate oxide thickness.

3 Reduction of I_{gate} through Pin Reordering

In this section, we propose a method for reducing I_{gate} through simultaneous pin reordering and state assignment. Traditionally, state assignment has been used to reduce standby mode I_{sub} by set-

[1] For example, [7] states that a 0.3V change in V_{gs}, V_{gd} leads to a decade change in I_{gate}. However, a reduction in V_{gs} of only ~0.1V yields a 10X drop in I_{sub}.

Table 2. Leakage estimation for 3 input NAND gate with 15Å oxides.

State	Estimated current [nA]			SPICE [nA]	% error
	I_{gate}	I_{sub}	I_{total}		
000	0.382	0.000	0.382	0.382	0.11%
001	0.709	6.339	7.048	7.047	0.02%
010	0.709	1.275	1.275	1.292	-1.25%
011	5.626	12.677	18.303	18.295	0.04%
100	0.676	0.000	0.676	0.675	0.18%
101	3.804	6.339	10.143	10.140	0.03%
110	3.804	0.000	3.804	3.641	4.48%
111	28.273	19.015	47.288	47.278	0.02%

Table 3. Leakage estimation for 3 input NAND gate with 17Å oxides.

State	Estimated current [nA]			SPICE [nA]	% error
	I_{gate}	I_{sub}	I_{total}		
000	0.196	0.000	0.196	0.197	-0.29%
001	0.402	0.761	1.163	1.163	-0.07%
010	0.446	0.399	0.446	0.477	-5.51%
011	6.774	1.522	8.295	8.291	0.05%
100	0.382	0.000	0.382	0.383	-0.42%
101	3.720	0.761	4.481	4.482	-0.02%
110	3.720	0.000	3.720	3.471	7.17%
111	31.971	2.282	34.253	34.248	0.02%

ting the output of each flip-flop to a known state during standby-mode such that I_{sub} is minimized. The standby mode state is chosen so that the stack effect occurs in as many gates as possible [13]. Although the logic correlation between gates prevents all gates from being in a low I_{sub} state, reasonable reductions in subthreshold leakage currents have been obtained using this method for circuit blocks [12]. Furthermore, the area and delay penalty incurred by the additional transistors required for forcing the output of a flip-flop to a given sleep state is minor. However, the presence of significant I_{gate} affects the state-dependence of the total leakage and must be considered. In this section, we first discuss the impact of I_{gate} on standby-mode state assignment in general and then propose a new method that combines state assignment with pin reordering for more effective total leakage reduction.

In general, the worst-case and best-case leakage states of common CMOS gates behave differently when both I_{sub} and I_{gate} are considered compared to I_{sub} alone. Table 2 showed that when only I_{sub} is considered, the worst-case leakage state for NAND structures occurs when all inputs are high as the PMOS devices leak in parallel and sum. For NOR structures, the reverse is true: all inputs set to low causes all NMOS devices to leak concurrently in parallel. For these two cases, we now include I_{gate}. In NAND gates with all inputs tied high, the NMOS devices in the pull-down stack all exhibit *worst-case* I_{gate} which adds to the large I_{sub} of the PMOS devices to create a large total leakage current. In the NOR gate with all inputs set to low, the PMOS devices have V_{gd}=V_{gs}=V_{dd} but since PMOS devices show very small I_{gate}, the overall impact will be small. Meanwhile, the parallel pull-down devices exhibit only reverse edge direct tunneling which is negligible. As a result of the these trends, we find that the *range* of total leakage current across states is broadened for NAND gates and compressed for NORs.

This is illustrated in Table 4 where the average leakage and the ratio of max/min leakage is shown for NAND and NOR gates over all possible input states. Results for 15Å and 17Å technologies are shown both with and without considering I_{gate}. Columns 2 and 3 show that even with a relatively low I_{gate} value for the T_{ox} = 17Å technology, the average leakage over all states in the gates studied increases by 10-35% when considering both I_{gate} and I_{sub} together. In the more aggressive 15Å technology, the rise in average leakage is 65-160% for NANDs and up to 310% for 4-in NOR gates. The last two columns show that the presence of I_{gate} significantly reduces the range of leakage current for NOR gates over all input states, while at the same time increases this range for NAND gates. For the 15Å technology, the ratio of maximum to minimum leakage current over all possible states is reduced from 21.3X in a 3-input NOR to 1.48X. This results from the complementary nature of I_{gate} and I_{sub} over the input space, meaning that NOR states with large I_{gate} have small I_{sub} and vice versa. On the other hand, the max/min leakage ratio for NAND gates increases by 2X in the 15Å technology since the same states that exhibit maximum I_{sub} also exhibit maximum I_{gate}.

In general, standby mode leakage in the presence of significant I_{gate} can be addressed with similar methods as used for I_{sub} leakage current. However, state assignment will be significantly more effective for circuits constructed predominantly from NAND gates, as opposed to NOR gates. Since in most of our benchmark circuits NAND gates outnumbered NORs 2-to-1, we found that the overall spread of total leakage current is typically increased slightly when I_{gate} is considered.

A key difference between the state dependence of I_{sub} and I_{gate} is that the magnitude of I_{sub} primarily depends of the *number* of OFF vs. ON transistors in a stack, while I_{gate} also depends strongly on the position of the ON/OFF transistors. We consider the 3-input NAND gate with input combinations 110 and 101 (where the first input value corresponds to the topmost NMOS), as shown in Table 2 for the 15Å process. When I_{gate} is neglected, the leakage current in these two states is the same, equal to 3.8nA. When including I_{gate} in the analysis, the total leakage in the 101 state increases to 10.14nA whereas the leakage current in state 110 is unchanged. Furthermore, in state 011, I_{sub} is increased by approximately 30% to 5.6nA, while I_{gate} is doubled, yielding a total leakage of 18.3nA. This dependence is a consequence of the different leakage of ON-transistors in scenario 1, where I_{gate} is negligible and scenario 2, where the I_{gate} sums with I_{sub} as discussed in Section 2.

The dependence of I_{gate} on the position of the ON-transistors in the stack suggests a combined approach where state-assignment is utilized for reducing I_{sub} while pin-reordering is targeted at I_{gate} reduction. Since pin reordering and state assignment are inter-dependent, this requires solving a combined optimization problem where a state-assignment and pin-ordering is determined for the entire circuit that minimizes the total standby leakage current. A number of heuristic methods for state-assignment alone have been proposed in the literature [12] using branch-and-bound methods. We therefore extend such a branch-and-bound method to incorporate simultaneous pin reordering. An input state search tree is first formulated using the approach presented in [14] and is traversed using the branch-and-bound traversal algorithm. This algorithm is augmented

Table 4. Impact of I_{gate} on state dependence with I_{leak}

Gate type	Average I_{leak} [nA]		max I_{leak} / min I_{leak} across all states	
	w/o I_{gate} (15Å / 17Å)	w/ I_{gate} (15Å / 17Å)	w/o I_{gate} (15Å / 17Å)	w/ I_{gate} (15Å / 17Å)
NAND2	7.25 / 8.05	12.0 / 8.62	26.6 / 53.00	44.40 / 56.85
NAND3	5.5 / 5.97	11.1 / 6.61	74.0 / 162.8	123.8 / 174.4
NAND4	3.8 / 3.99	9.9 / 4.73	138 / 327.7	231.4 / 351.0
NOR2	7.3 / 7.84	13.6 / 8.60	7.57 / 19.50	1.40/6.10
NOR3	5.7 / 5.79	15.2 / 6.93	21.26 / 59.00	1.48 / 9.28
NOR4	4.1 / 3.93	16.8 / 5.45	21.26 / 120.5	1.94 / 12.37

such that each time a leaf node is reached, and the input state of the circuit is completely defined, we apply pin reordering by placing all off transistors at the bottom of the stack for each gate. This substantially decreases I_{gate} while also slightly decreasing I_{sub}. We then update the total leakage for that leaf solution with the new I_{gate} and I_{sub} leakage and continue the traversal of the state tree. Despite the pruning that is performed during the traversal, the search space is very large and an exhaustive traversal of the tree is not possible. We therefore place a limit on the run time of the algorithm and report the best solution found by the search within this allotted time.

In addition to the branch-and-bound approach, we also implemented a simple random search approach. For each randomly generated input state, the state of each transistor in a stack is determined and optimal pin reordering is performed. The input state/pin reordering combination with minimum total leakage is then recorded. In Section 4, we show a comparison between the two approaches. Since pin reordering can affect the circuit performance, it must be restricted to stack inputs that are not timing critical. However, the delay impact of pin reordering is relatively small and was ignored in our implementation.

Finally, we apply pin reordering for the purpose of runtime leakage reduction. Since I_{sub} depends on the number of OFF transistors in series, it is difficult to reduce I_{sub} during runtime since the state of the circuit cannot be changed. However, I_{gate} also depends on the position of the OFF-transistors in the stack. The probability of being in a high state (referred to as the *state probability*) is significantly lower for certain nodes in the circuit than others. We can use this information to place nodes with a low state-probability at the bottom of the transistor stack. Based on given state probabilities of the primary inputs (PIs), we compute the state probability of each node in the circuit using the method described in [15]. We then order the transistors in a stack from top to bottom in decreasing order of their state probabilities. In this manner, the likelihood of scenarios 2 and 3 (from Section 2) occurring during run time is increased while the occurrence of scenario 1 is reduced and hence the total I_{gate} for the circuit is diminished. Although this method is not as effective at reducing I_{gate} as combined state-assignment and pin-reordering, we assert that runtime approaches to leakage reduction (i.e., approaches that do not rely on the use of standby modes) will become increasingly important in the future due to shrinking I_{on}/I_{leak} ratios in nanometer MOSFETs.

4 Results

The proposed method for gate tunneling and subthreshold leakage current estimation was implemented and tested for 21 benchmark circuits. These circuits include 10 ISCAS85 circuits [16], 10 MCNC benchmark circuits [17], and one 64-bit ALU benchmark circuit. All circuits were synthesized with a 0.18 µm Artisan library using Synopsys Design Compiler and were scaled to a 100nm technology (results in this section use the 15Å process only). For SPICE simulation, Berkeley predictive SPICE models for 100nm technology were used along with the gate tunneling current model discussed. The total leakage current for each circuit was determined for 100 random input states using the proposed leakage estimation method and also using SPICE simulation. The results are shown in Table 5. For each circuit, the average leakage current with and without gate tunneling current is shown. The proposed method had an average error of 0.09% over all circuits and simulated circuit states, with a maximum error of 0.67% across any circuit/input state combination. The final column in Table 5 shows the run time for the proposed leakage estimation method (note units differ). The run time speedup compared to SPICE ranged from 5,000 to 52,000X.

Table 5. Leakage estimation results for benchmark circuits.

Circuit	# gates	Estimated I_{leak}, [µA] (avg)		SPICE I_{leak} [µA] (avg)	% error (avg/max)	Run time	
		w/o I_{gate}	w/ I_{gate}			Proposed (ms)	SPICE (s)
C432	121	1.71	2.82	2.82	0.13/0.32	0.18	9.36
C499	517	6.44	9.99	9.99	0.01/0.02	2.4	38.4
C880	325	4.49	7.08	7.08	0.06/0.14	1.5	27.8
C1355	478	6.36	10.22	10.22	0.02/0.06	2.5	41.4
C1908	425	5.55	8.61	8.61	0.01/0.04	2.7	35.8
C2670	750	9.48	14.46	14.46	0.02/0.06	3.9	60.6
C3540	890	11.76	18.98	18.98	0.04/0.08	6.3	100.2
C5315	1524	20.49	32.28	32.28	0.01/0.02	11.1	180.8
C6288	2388	32.82	54.54	54.53	0.02/0.04	34.4	971.3
C7552	1916	25.86	39.67	39.69	0.04/0.07	14.5	207.8
alu64	1791	25.29	40.58	40.63	0.14/0.35	42.6	245.0
i1	39	0.45	0.69	0.69	0.11/0.42	0.4	2.0
i2	95	0.91	1.89	1.88	0.36/0.67	0.8	9.7
i3	92	1.25	1.89	1.88	0.17/0.48	0.5	6.1
i4	160	2.33	3.81	3.81	0.03/0.08	1.2	11.0
i5	198	2.61	4.04	4.04	0.01/0.02	1.3	10.1
i6	359	5.02	8.11	8.13	0.22/0.44	1.5	26.6
i7	450	6.15	10.02	10.04	0.24/0.45	2.2	36.2
i8	725	10.40	16.73	16.74	0.07/0.22	3.7	67.0
i9	459	6.48	10.54	10.56	0.19/0.39	2.1	38.3
i10	1794	24.33	38.42	38.43	0.04/0.05	15.0	747.9
Avg.					0.09/0.21		

Table 6. Pin reordering results for leakage reduction.

Circuit	Max. reduction (%)									
	Sleep mode								Runtime mode	
	State assignment only				State assign. & pin reordering					
	I_{leak}		I_{gate}		I_{leak}		I_{gate}		I_{leak}	I_{gate}
	Rand.	B-n-B	Rand.	B-n-B	Rand.	B-n-B	Rand.	B-n-B		
C432	13.47	15.90	19.87	25.60	25.88	28.32	46.88	59.52	4.02	8.79
C499	5.64	6.96	8.04	9.16	11.81	11.98	25.79	28.25	2.09	4.58
C880	15.54	13.24	20.89	23.78	25.13	25.05	42.52	45.43	3.30	7.51
C1355	5.95	8.16	8.05	10.18	17.79	19.94	32.24	33.50	3.43	7.19
C1908	4.19	5.84	6.14	9.23	12.66	13.67	25.93	27.76	3.38	7.91
C2670	6.55	13.21	12.59	22.43	15.07	17.90	29.90	33.80	2.17	5.14
C3540	5.70	7.33	6.66	6.58	19.93	19.00	36.02	36.40	5.85	13.17
C5315	6.48	9.13	9.91	10.28	16.65	17.39	31.68	34.34	1.98	4.39
C6288	11.31	14.80	8.01	12.02	28.64	29.35	46.63	45.86	4.79	10.21
C7552	3.47	6.94	6.21	8.86	12.88	16.74	25.85	28.10	1.71	4.04
alu64	13.32	16.34	20.05	32.86	23.15	28.64	38.80	49.23	2.97	6.49
i1	19.09	21.49	33.23	39.03	19.09	21.50	33.23	40.19	0.05	0.11
i2	11.13	17.60	22.81	55.42	13.92	22.20	27.62	69.68	1.25	2.09
i3	12.91	25.05	17.20	22.19	12.91	25.05	17.74	22.19	0.01	0.03
i4	10.72	28.84	17.64	34.63	22.96	25.31	38.39	32.09	1.55	3.15
i5	4.02	10.65	5.16	12.22	20.47	38.40	35.58	55.93	3.28	7.45
i6	43.16	56.32	51.12	65.12	52.55	62.92	73.37	82.12	4.22	8.78
i7	42.52	58.46	60.90	70.52	44.93	61.37	66.77	75.81	3.47	7.09
i8	24.79	16.11	27.14	25.32	37.16	26.82	54.55	48.84	6.25	13.56
i9	36.31	25.63	39.46	44.35	48.97	33.91	64.67	61.87	3.86	8.42
i10	4.61	9.00	6.93	12.36	14.05	16.27	26.30	28.94	6.03	13.58
Avg.	14.33	18.43	19.43	26.28	23.65	26.75	39.07	44.76	3.13	6.84

Table 6 shows the results of leakage minimization through state assignment and pin reordering for circuits in sleep mode, using the two optimization approaches discussed in Section 3: random search with 10000 input vectors and the branch-and-bound algorithm. In columns 2-5 the leakage reduction results are shown when only state assignment is used while columns 6-9 show the results when combined state assignment and pin reordering are applied. As seen from Table 6, state assignment is less effective for large circuits (implying many levels of logic) due to functional correlations among the gates. Most of the literature focuses on comparing the minimum leakage state with the maximum possible leakage but comparing to the average state is more relevant and we use that convention here. Since gate leakage is strongly dependent on the stack ordering, we also compare our results with the leakage current considering an average pin ordering. Based on the state probability of the nodes, we find the leakage under best and worst pin ordering for a circuit, and then take the average of these two leakage values. As shown in Table 6, the branch-and-bound approach performs better than random search method. In the branch-and-bound approach, the average leakage reduction using only state assignment over all circuits is 18%, while the reduction in the gate leakage component of the total (I_{gate}) is 26%. When performing simultaneous pin reordering and state assignment (columns 6-9), the reduction in total leakage is 27% on average over all circuits with an average reduction in the I_{gate} component of 45%. The impact of pin reordering on I_{gate} is pronounced, reducing I_{gate} by up to 82%.

The runtime leakage reduction using pin reordering is also shown in Table 6. These experiments were conducted as described in Section 3 - a single pin reordering is performed based on state probabilities at all circuit nodes and 10000 input vectors with each input having a state probability of 0.5 are applied to both the best- and worst-reordered topologies. In Table 6, we show the reduction rate between the leakage of best reordered topology and that of an average ordered circuit. The total leakage savings over all 10000 states is 3.13% on average over all circuits for an input state probability of 0.5. Note that I_{gate} is reduced by a larger factor than total leakage (I_{leak}), as expected; by 6.84% on average and > 10% in several cases. Also, the leakage reduction is dependent on the PI state probabilities. For instance, when all PIs have state probabilities of 0.25 rather than 0.5, the average runtime I_{leak} reduction becomes 4.53% over all circuits with C6288 showing an 11.51% reduction and I_{gate} improvements range up to 25%. While the runtime improvements using pin reordering are not large, they do benefit power consumption at all times rather than during standby mode only. Note that i1 and i3 benchmark circuits have almost no improvement from pin reordering. While all other circuits consist of at least 50% NAND gates, only ~5% of the gates in these two small circuits are NAND gates. Since pin reordering is only effective for NAND gates for our implementation, the leakage improvement is negligible for circuits i1 and i3.

Figure 4 shows the impact of state assignment and pin reordering on circuit c6288 with the state probabilities of primary inputs set to 0.5. The figure shows the achievable reductions in I_{gate}, I_{sub}, and I_{leak} for the three different scenarios of Table 6. State assignment works equally well for I_{sub} and I_{gate} whereas the addition of pin reordering can be seen to provide substantial benefits for both I_{gate} and I_{leak} with little improvement for I_{sub}. Technologies with higher components of I_{leak} due to I_{gate} will exhibit greater improvements in both sleep mode and runtime leakage when applying pin reordering.

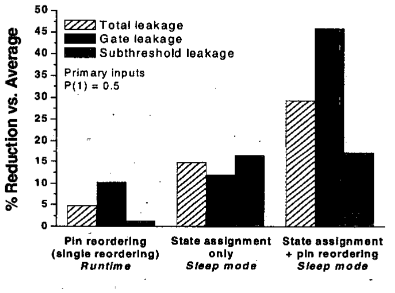

Figure 4. Various leakage reduction techniques compared to the average leakage over 10000 random input states for C6288.

5 Conclusions

We developed a fast approach to computing total leakage current in large circuit blocks considering both subthreshold and gate tunneling currents. The proposed approach accurately accounts for the complex interaction between I_{gate} and I_{sub} in stacked MOS configurations and is based on precharacterized tables of individual leakage currents for three distinct scenarios. Based on the proposed analysis method, we propose the use of pin reordering to effectively limit gate leakage as I_{gate} depends strongly on the location of OFF devices within a non-conducting stack. Results show 22-82% reductions in I_{gate} during standby modes using pin reordering. When applied to runtime leakage, pin reordering reduces I_{gate} by up to 25% depending on circuit topology and input data statistics.

Acknowledgements

This research was performed with the support of MARCO research grant 98-DF-660 and SRC contract 2003-TJ-1074.

References

[1] A. Ono, *et al.*, "A 100nm node CMOS technology for practical SOC application requirement," Proc. IEDM, pp. 511-514, 2001.

[2] 2001 International Technology Roadmap for Semiconductors.

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

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

[5] C.-H. Choi, *et al.*, "Impact of gate direct tunneling on circuit performance: a simulation study," IEEE Trans. Electron Devices, pp. 2823-2829, Dec. 2001.

[6] S. Schwantes and W. Krautschneider, "Relevance of gate current for the functionality of deep submicron CMOS circuits," European Solid-State Device Research Conf., 2001.

[7] F. Hamzaoglu and M.R. Stan, "Circuit-level techniques to control gate leakage for sub-100nm CMOS," Proc. ISLPED, 2002.

[8] http://www-device.eecs.berkeley.edu/~ptm

[9] D. Lee, *et. al.*, "Simultaneous subthreshold and gate-oxide tunneling leakage current analysis in nanometer CMOS design," Proc. ISQED, pp. 287-292, 2003.

[10] N. Yang, W. K. Henson, and J. J. Wortman, "A comparative study of gate direct tunneling and drain leakage currents in N-MOSFETs with sub-2nm gate oxides," IEEE Trans. Electron Devices, pp. 1636-1644, Aug. 2000.

[11] Y. Taur, "CMOS design near the limit of scaling," IBM J. R&D, pp. 213-222, March/May 2002.

[12] M.C. Johnson, *et. al.*, "Models and algorithms for bounds on leakage in CMOS circuits," IEEE Trans. CAD, pp. 714-725, June 1999.

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

[14] D. Lee and D. Blaauw, "Static leakage reduction through simultaneous threshold voltage and state assignment," Proc. DAC, in press, 2003.

[15] S. Ercolani, M. Favalli, M. Damiani, P. Olivo, B. Ricco, "Estimate of signal probability in combinational logic networks," Proc. European Test Conference, 1989, pp. 132 - 138.

[16] F. Brglez and H. Fujiwara, "A Neutral Netlist of 10 Combinatorial Benchmark Circuits", Proc. ISCAS, 1985, pp.695-698.

[17] http://www.cbl.ncsu.edu