# Efficient Techniques for Gate Leakage Estimation

Rahul M. Rao[1], Jeffrey L. Burns[2], Anirudh Devgan[2], Richard B. Brown[1]

[1]Department of EECS
University of Michigan, Ann Arbor
MI, USA 48109
{rmrao,brown}@eecs.umich.edu

[2]Austin Research Labs
IBM, Austin
TX, USA 78758
{jlburns,devgan}@us.ibm.com

## ABSTRACT

Gate leakage current is expected to be the dominant leakage component in future technology generations. In this paper, we propose methods for steady-state gate leakage estimation based on state characterization. An efficient technique for pattern-dependent gate leakage estimation is presented. Further, we propose the use of this technique for estimating the average gate leakage of a circuit using pattern-independent probabilistic analysis. Results on a large set of benchmark ISCAS circuits show an accuracy within 5% of SPICE results with 500X to 50000X speed improvement.

## Categories and Subject Descriptors

I.6.5 [Simulation and Modeling]: Model Development.

## General Terms

Algorithms, Measurement, Performance.

## Keywords

Leakage, gate leakage, estimation, pattern-dependent, pattern-independent.

## 1. INTRODUCTION

The desire to improve device performance has resulted in aggressive scaling of gate oxide thickness to below 2nm. At such oxide thicknesses, there occurs an increased probability of direct tunneling of charge carriers through the gate oxide. This has resulted in an alarming increase in gate leakage current. Gate leakage is predicted to increase at a rate of more than 500X per technology generation, while sub-threshold leakage increases by around 5X for each technology generation [1]. This could result in gate leakage becoming the dominant contributor to leakage current. Gate leakage power, which was almost non-existent in previous technology generations, is expected to contribute more than 15% to the total power consumption in a 2004 technology generation.

Extensive work has been done on the modeling, analysis and reduction of gate leakage current from a CMOS technology viewpoint [2,3]. Since these proposed solutions are currently infeasible, there is a need to address gate leakage from a circuits perspective. The effect of gate leakage on circuit performance and dynamic behavior of the floating body in SOI devices was examined in [4, 5]. In [6], the authors presented circuit-level techniques for gate leakage minimization. In each of these reports, extensive SPICE simulations were performed to obtain estimates of gate leakage. This can be extremely time-consuming, especially for large circuits. To be able to obtain full-chip estimates of gate leakage, it is necessary to accurately and quickly estimate individual gate leakage currents and sum the gate leakage of all of the devices.

In this paper, we propose efficient techniques for gate leakage estimation in steady state. These techniques are based on a switch-level analysis of the circuit and hence can efficiently analyze large circuits. An analysis of gate leakage and its dependencies is presented in Section 2. In Section 3 we present a technique for accurately determining gate leakage of a circuit for a given input pattern. We extend this technique by using a pattern-independent probabilistic analysis to estimate the average gate leakage current of the circuit in Section 4. These methods are applied to a set of benchmark ISCAS circuits and the results are compared to those of SPICE in Section 5. The findings and contributions are summarized in Section 6.

## 2. GATE LEAKAGE REVIEW

Gate leakage current for an NMOS transistor as a function of gate-to-source ($V_{GS}$) and drain-to-source ($V_{DS}$) bias in a state-of-the-art sub 0.1μm SOI process is shown in Fig. 1. Gate leakage is an exponential function of the electric field across the gate oxide, so gate leakage current shows an exponential dependence on the gate-to-source ($V_{GS}$) bias. At high gate

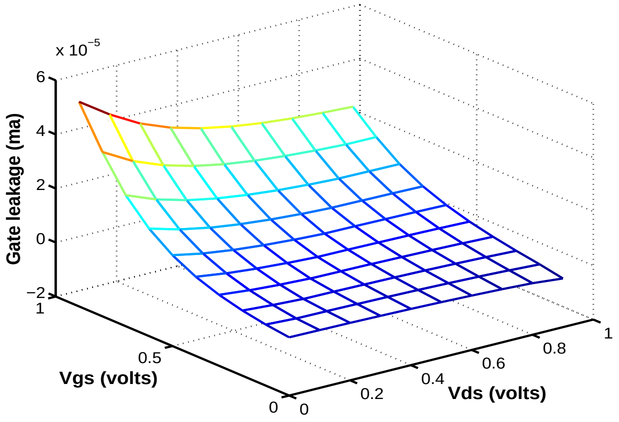

**Figure 1. Gate leakage current as a function of gate and drain bias for an NMOS device.**

*ISLPED'03*, August 25-27, 2003, Seoul, Korea.

bias, gate leakage current decreases with increasing drain-to-source bias. This can be attributed to the fact that a higher drain voltage results in a smaller electric field across the gate oxide at the drain end of the channel (lower $V_{GD}$). At low gate bias, gate leakage was found to increase with increasing drain bias (due to the increase in reverse gate leakage with increasing drain bias, i.e., $V_{GD}$).

Gate leakage current was found to be almost insensitive to the body-node voltage. At high gate bias, the channel shields the body node from the gate. At low gate bias, the edge-tunneling currents from the source-drain extensions to the gate dominate [7]. Thus, gate leakage of a transistor for a given technology is determined by the gate-to-source ($V_{GS}$) and gate-to-drain ($V_{GD}$) bias seen by the device.

The bias conditions seen by a device in a circuit depend on its position in the circuit and the applied input vector. The six possible bias conditions for an NMOS transistor at steady state are shown in Fig. 2 with the associated gate leakage currents for those bias conditions. The remaining two states, namely <101> and <110> at the gate, drain and source of the transistor, are transient states. These states cannot occur in steady state as the drain/source nodes will eventually reach logic ‘1’ level resulting in the <111> bias state. In all of these states, the device terminals are assumed to be at full supply-rail voltages. An identical set of states exists for a PMOS transistor, i.e., a <111> at the gate, drain and source of a PMOS transistor would correspond to an S(0) state. Let the gate leakage for a unit-width transistor in each of these *k* states be denoted by *Ig(k)*.

It can be seen that a device exhibits gate leakage only in four of the six possible states. In each of these states, there exists a path from the gate and source or drain of the device to $V_{DD}$ or $V_{SS}$ lines. In other words, the gate of the leaky device acts like a resistor connected between the $V_{DD}$ and $V_{SS}$ lines. The gate leakage of any device in the circuit can hence be estimated from its bias state if there exists a conducting path from its device terminals to the supply-rails. The total gate leakage of the circuit is the sum of the gate leakage of the individual devices since they can be considered to be parallel resistances connected between $V_{DD}$ and $V_{SS}$.

## 3. PATTERN DEPENDENT GATE LEAKAGE ESTIMATION

Since the gate leakage of a device depends on its bias state, gate leakage of a transistor in a stack depends on the state of the internal nodes. The state of the internal nodes is determined by the applied input vector. This section describes a method to accurately estimate the total gate leakage of a circuit for a given input vector.

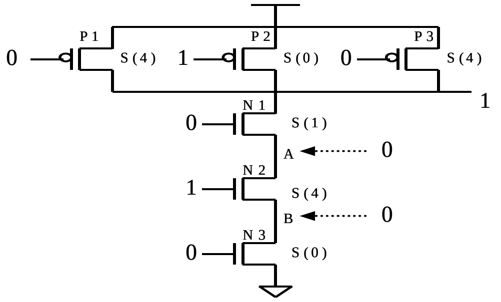

**Figure 3. Determining the state of each device for a given input pattern**

Given an input vector, it is possible to perform a switch-level simulation to determine the state of the internal nodes of the circuit. Here, it is assumed that the internal nodes attain full logic levels (i.e., are either at $V_{DD}$ or $V_{SS}$). Also, in a transistor stack, it is assumed that the entire voltage drops across the uppermost ‘OFF’ device. These are fairly good assumptions, since gate leakage is an exponential function of the terminal voltages and is negligible if no conducting path exists from the device terminals to the supply-rails. When a device in a NMOS transistor stack is ‘OFF’, there exists no conducting path from $V_{DD}$ to the drain terminal of devices below it and hence these nodes can be assumed to be nearly at ground potential. For instance, when input <010> is applied to a Nand3 cell as shown in Fig. 3, the output is at logic high state. The internal nodes A and B are assumed to be at ‘0’ state. The bias state of each transistor can then be identified. In this case, the PMOS transistors P1 and P3 are ‘ON’ with a high $V_{GS}$ and high $V_{GD}$ and hence are in S(4) state. On the other hand, the PMOS transistor P2 is ‘OFF’ with a zero $V_{GS}$ and $V_{GD}$ and hence is in state S(0). Similarly, the NMOS transistors N1, N2 and N3 are assumed to be in states S(1), S(4) and S(0), respectively. The total gate leakage can then be computed by scaling the width of each device by the unit-width leakage in that state and adding the individual gate leakages, i.e., $Ig = \Sigma_i W(i) Ig(s(i))$

A special check needs to be made for devices in S4 state. This is because a device may be in S4 state without having a path from either its drain or source terminals to the supply-rails. This arises due to the assumption that all internal nodes attain full logic levels. For instance, consider the transistor N2. Though it is in S(4) state, there exists no path from its drain or source to the $V_{SS}$ terminal and hence the device does not exhibit significant gate leakage. Therefore, all devices in such a state can be considered to be equivalent to a device in the S(0) state.

This method can be used for the analysis of any circuit with full-rail CMOS logic families. It is equally applicable to full custom and standard-cell based designs and can also be used

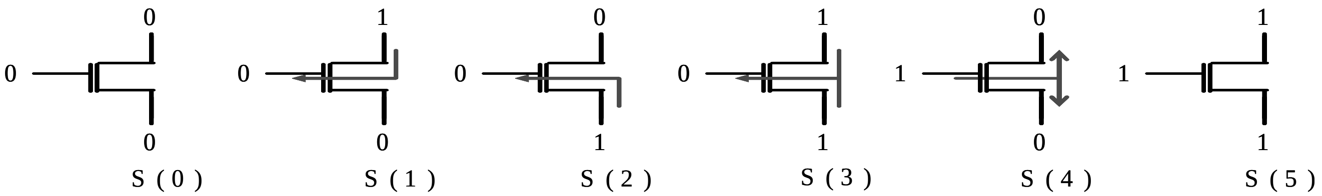

**Figure 2. All possible bias conditions for a NMOS transistor in a circuit. Identical set of state exist for a PMOS transistor with all terminal voltages flipped and arrow directions reversed.**

**Table 1. Pattern-dependent gate leakage estimate compared with SPICE for all patterns for a Nand3 cell.**

| Input <ABC> | Spice (mA) | Proposed Method (mA) | Error (%) |
|---|---|---|---|
| <000> | 6.52E-05 | 6.49E-05 | 0.46 |
| <001> | 1.30E-04 | 1.30E-04 | 0.15 |
| <010> | 5.94E-05 | 6.00E-05 | -1.01 |
| <011> | 1.94E-04 | 1.95E-04 | -0.36 |
| <100> | 4.56E-05 | 4.49E-05 | 1.54 |
| <101> | 1.08E-04 | 1.10E-04 | -1.67 |
| <110> | 2.44E-05 | 2.43E-05 | 0.41 |
| <111> | 2.58E-04 | 2.59E-04 | -0.23 |
| **Avg.** | **1.11E-04** | **1.11E-04** | **-0.27** |

for cell-library characterization. However, it should be noted that this method is only applicable for gate leakage estimation once the circuit has reached steady state (i.e., the internal nodes have settled to their dc values). To demonstrate its accuracy, the gate leakage estimate obtained by this method for all possible states of a Nand3 are tabulated in Table 1 and compared with SPICE simulation results. As seen, the leakage estimates obtained by this method are very accurate, with an average error of less than 1% and a maximum error of less than 2%.

## 4. PATTERN INDEPENDENT GATE LEAKAGE ESTIMATION

This section presents a technique for accurately and quickly estimating the average gate leakage of the circuit. This is achieved by using a probabilistic analysis in conjunction with the above method for pattern-dependent gate leakage estimation. The average gate leakage of the circuit is the probabilistic mean value of the gate leakage of the circuit.

$$Ig_{av} = E[\Sigma_i W(\dot{i}) Ig(s(i))] \text{ i.e.,}$$

$$Ig_{av} = \Sigma_i W(i) \cdot \Sigma_j P(j) Ig(s(j)) \qquad (1)$$

Here *P(j)* is the probability of occurrence of state *j*. Thus, computing the average gate leakage of the circuit requires determining the probability of occurrence of each state of each device in the circuit.

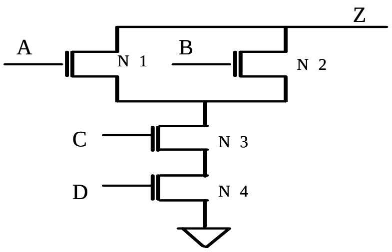

**Figure 4. NMOS tree of a typical Oai211 cell**

**Table 2. Probability of occurrence of state S(1) for all devices in a typical Oai211 cell shown in Fig. 4**

| Device | Probability of S(1) |
|---|---|
| **N1** | (1-pA)*(1-pB) |
| **N2** | (1-pA)*(1-pB) |
| **N3** | (pA + pB - pA*pB)*(1-pC) |
| **N4** | (pA + pB - pA*pB)*pC*(1-pD) |

**Table 3. Probability of occurrence of state S(4) for all devices in a typical Oai211 cell shown in Fig. 4**

| Device | Probability of S(4) |
|---|---|
| **N1** | pA*pC*pD |
| **N2** | pB*pC*pD |
| **N3** | pC*pD |
| **N4** | pD |

Consider the NMOS tree of a typical Oai211 cell as shown in Fig. 4. Let the inputs to the cell have probability pA, pB, pC and pD of being in the logic high state, respectively. A device is in state S(1) if it is the uppermost 'OFF' device in a transistor stack. For the device to be in S(1) state, there should exist an 'ON' path from the supply-rail to its drain terminal (i.e., there exists a conducting path above it), while the device itself should be 'OFF'. In other words, the devices above it in the stack should have their inputs at logic 1 state, while the device itself has a logic 0 input. For instance, for the device N4 to be in state S(1), either input A or input B should be high, input C should be high, while input D is low. The probability of occurrence of this combination can easily be computed using the input probabilities as shown in Table 2. Similarly, a device is in a valid S(4) state if the device is 'ON' and there exists a path from its source terminal to the $V_{SS}$ rail (i.e., there exists a conducting path below it). In other words, the devices below it in the transistor stack and the device itself should be 'ON'. For device N1 to be in state S(4), inputs A, C and D should be high. This can also be computed as shown in Table 3. Here, the inputs are assumed to be uncorrelated. Thus, the input occurrence probabilities can be used to determine the probability of occurrence of each state in the cell. This can then be used to estimate the average gate leakage of the cell as given by Eqn. 1. The accuracy of this method depends on the accuracy of the input occurrence probabilities for each cell and their correlation. Table 4 lists the average gate leakage obtained using this method compared with SPICE for a variety of commonly used library cells. The error in average gate leakage estimation is less than 5% for all these cells, with an average error of 2% over all the cells.

**Table 4. Pattern-independent gate leakage estimates compared with SPICE for various commonly used cells.**

| Cells | Spice (mA) | Proposed Method (mA) | Error (%) |
|---|---|---|---|
| **Inv** | 2.64E-04 | 2.65E-04 | -0.19 |
| **Nand2** | 6.60E-04 | 6.68E-04 | -1.21 |
| **Nand3** | 1.11E-04 | 1.11E-04 | 0.32 |
| **Nand4** | 1.56E-04 | 1.54E-04 | 0.97 |
| **Nor2** | 6.04E-05 | 6.06E-05 | -0.31 |
| **Nor3** | 9.78E-05 | 9.74E-05 | 0.44 |
| **Oai21a** | 1.07E-04 | 1.05E-04 | 1.81 |
| **Oai21b** | 1.23E-04 | 1.20E-04 | 2.09 |
| **Aoi21a** | 1.09E-04 | 1.04E-04 | 4.42 |
| **Aoi21b** | 1.18E-04 | 1.12E-04 | 4.81 |
| **Oai211a** | 1.91E-04 | 1.86E-04 | 2.83 |
| **Oai211b** | 1.54E-04 | 1.51E-04 | 1.90 |

## 5. RESULTS

The proposed methods were implemented and tested for a set of benchmark ISCAS circuits implemented in a sub 0.1µm advanced SOI process. The total gate leakage was computed for 1000 random input vectors using the proposed pattern-dependent leakage estimation scheme presented in Section 3 and compared with the results obtained from SPICE. The average error across all the input patterns was less than 0.5% with a maximum error of 3.5% for a vector for the c880 circuit. The average speed-up compared to SPICE was about 500X. As an example, Fig. 5 shows the percentage error for all 32 possible vectors for the 5-input c17 benchmark circuit compared to SPICE.

The pattern-independent gate leakage estimation technique was extended to determine the average gate leakage of the entire circuit by computing the probability of occurrence for each node in the circuit using the approach presented in [8] while assuming an occurrence probability of 0.5 for each input to the circuit. The probability of the output node of each cell was computed by determining the probability of the existence of a conducting path in its PMOS tree. Given the probabilities of the inputs to a cell, the average leakage of the cell was computed using the technique presented in the Section 4. Table 5 lists the average gate leakage obtained for a set of ISCAS benchmark circuits using this approach. The average leakage from SPICE is obtained by applying 10000 randomly generated vectors to the circuit. The average error is less than 2% (with a maximum error or less than 4%) while the run-time is over 50,000X faster as than SPICE.

## 6. CONCLUSIONS

In this paper we propose techniques for quick and accurate estimation of gate leakage current. We present an efficient pattern-dependent gate leakage estimation technique applicable to any full-rail CMOS logic family for determining the gate leakage of a circuit for any specified input vector. Further, we propose a pattern-independent gate leakage estimation technique that can be used to determine the average gate leakage of a circuit with an average error of less than 2% . These techniques obtain 500X to 50,000X speedup compared to SPICE.

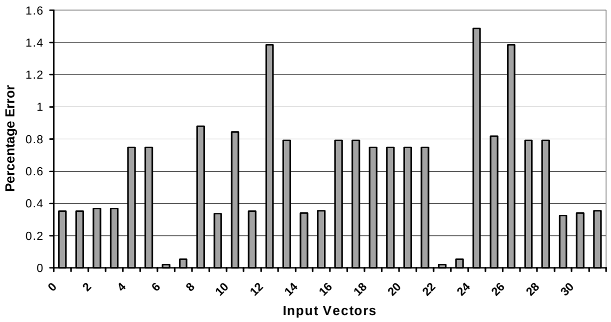

**Figure 5. Percentage error in gate leakage estimate (obtained using method in Section 3) for all vectors for c17 circuit.**

## 7. REFERENCES

[1] International Technology Roadmap for SemiConductors, *http://public.itrs.net/Files/2001ITRS/Home.html.*

[2] R. Langevelde, A. Scholten, R. Duffy, F. Cubaynes, M. Knitel and D. Klaassen, "Gate current: Modeling, $\Delta L$ Extraction and impact on RF performance,"

[3] Y.C. Yeo, et.al, "Direct Tunneling Gate Leakage Current in Transistors with Ultrathin Silicon Nitride Gate Dielectric," *IEEE Electron Device Letters*, vol. 21, no. 11, pp. 540-542, Nov. 2000.

[4] C. T. Chuang and R. Puri, "Effects of Gate-to-Body Tunneling Current on Pass-Transistor Based PD/SOI CMOS Circuits," *IEEE Intnl. SOI Conf.*, pp. 121-122, Oct. 2002.

[5] C. Choi, K. Nam, Z. Yu and R. Dutton, "Impact of Gate Direct Tunneling Current on Circuit Performance: A Simulation Study," *IEEE Trans. on Electron Devices,* vol. 48, no. 12, pp. 2823-2829, Dec. 2001.

[6] F. Hamzaoglu and M. Stan, "Circuit-Level Techniques to Control Gate Leakage for sub-100nm CMOS," *Proc. ISLPED*, pp. 60-63, Aug. 2002.

[7] K. Yang, et.al, "Edge Hole Direct Tunneling Leakage in Ultrathin Gate Oxide p-Channel MOSFETs," *IEEE Trans. on Electron Devices,* vol. 48, no. 12, pp. 2790-2795, Dec. 2001.

[8] E. Acar, et.al, "Leakage and Leakage Computation Analysis for Combinational Circuits," *Proc. ISLPED*, Aug. 2003.

**Table 5. Average gate leakage estimate (using method in Section 4) compared with SPICE for benchmark circuits.**

| Ckt | Spice (mA) | Proposed Method (mA) | Error (%) |
|---|---|---|---|
| **c17** | 4.39E-04 | 4.42E-04 | 0.68 |
| **c432** | 1.83E-02 | 1.82E-02 | -0.63 |
| **c499** | 4.32E-02 | 4.32E-02 | -0.04 |
| **c880** | 2.99E-02 | 3.01E-02 | 0.47 |
| **c1355** | 4.16E-02 | 4.16E-02 | 0.16 |
| **c1908** | 5.87E-02 | 5.98E-02 | 1.85 |
| **c2670** | 8.80E-02 | 8.93E-02 | 1.49 |
| **c5315** | 1.89E-01 | 1.91E-01 | 0.92 |
| **c6288** | 1.56E-01 | 1.61E-01 | 3.23 |
| **c7552** | 2.52E-01 | 2.55E-01 | 1.24 |