Topological Analysis for Leakage Prediction of Digital Circuits

Wenjie Jiang, Intel Corporation, Wenjie.Jiang@Intel.com
Vivek Tiwari, Intel Corporation, Vivek.Tiwari@intel.com
Erik de la Iglesia, Aplatform Internet Service, Erik@aplatform.com
Amit Sinha, Massachusetts Inst. Of Tech, Sinha@mit.edu

Abstract

Subthreshold leakage current is becoming an increasingly significant portion of the power dissipation in microprocessors due to technology and voltage scaling. Techniques to estimate leakage at the full chip level are indispensable for power budget allocation. In addition, simple and practical approaches and rules of thumb are needed to allow leakage to become part of the vocabulary of all designers. This paper focuses on the impact of circuit topology on leakage, which is often abstracted through what is referred to as the stacking factor. The stacking factor, which captures the leakage reduction in series connected devices, is a first order term in leakage estimation equations and has significant impact on estimation results. The authors present two analysis methods, a mathematical and an empirical, to identify the stacking factor for leakage prediction. Understanding the stacking factor, as well as obtaining an accurate estimate of its value, is critical in reducing prediction uncertainty. As leakage prediction becomes a bigger factor in roadmap decisions, reducing leakage prediction uncertainty will be key in accurate determination of product specifications.

1. Introduction

With the ever growing demand for low power and high performance processors, device dimensions and operating voltages are constantly being reduced. As processor voltages are scaled, transistor threshold voltages also have to be lowered to maintain performance. However, subthreshold leakage current in MOSFETs increases exponentially as the threshold voltage is reduced. This results in increased standby power in mobile and handheld systems. The leakage problem is not restricted to the battery-operated domain. Dynamic memory designers are already familiar with this problem. Predictive estimates in the industry indicate that the severity of the leakage problem will only increase and will play a crucial role in the feasibility of future processors. Leakage estimation and reduction therefore are becoming increasingly important.

For an individual transistor (device), leakage is a function of several device parameters (*cf.* Section 2.1) as well as terminal voltages and junction temperature. However, the leakage current for a circuit is not simply the sum of leakage current for all its devices. The actual circuit topology is a primary determinant of the overall leakage current. In particular, series connected devices, or stacked devices, have lower leakage than the sum of the leakage for each device taken in isolation. This is often referred to as the *stacking effect*. Leakage estimation can be done at different levels depending on computational resources and simulation feasibility. Precise circuit simulators (such as HSPICE[7]) can accurately account for stacking effects, but are only practical for small circuits. While being extremely accurate, such tools have convergence problems or might take too much time. Faster techniques based on stacking models have been explored in depth in [1] [2]. These techniques are based on the BSIM2 leakage model and exploit an iterative solution to the equation for parallel leakage paths. While these techniques provide accurate estimates of stacking effects, their use is limited to small circuits. We are primarily concerned with early estimates for large circuits and even at the full chip level.

The leakage estimation problem is aggravated by the fact that it not only depends on circuit topology but is also a strong function of the input vector. For a circuit with n primary inputs, there are 2^n input vectors. Due to the exponential complexity of the input vector space, exhaustive simulations are only feasible for small circuits. It is desirable to have techniques that provide simple methods to deal with input dependence.

This paper presents two methods for determining the effect of circuit topology (stacking effect) on leakage. First, a mathematical approach, based on analysis of a full-chip netlist is presented. Second, we explore an empirical heuristic technique, based on leakage measurements for small blocks or gates, to estimate the minimum, maximum and average leakage current in large circuits such as Functional Unit Blocks (FUBs) or even the entire processor. Neither approach requires exhaustive simulation for large circuits. This is important when quick leakage estimates are desired at an early phase in design. The techniques presented account for topological leakage dependence (stacking effects, size variations etc.) and also factor in the input dependence. The rest of the paper is organized as follows. Section 2 provides the theoretical background for the stacking effect. Section 3 presents the mathematical approach – *Effective Leakage Width* method, and Section 4 presents the empirical approach – *Equivalent Stacking Factor* method. Section 5 presents a brief overview of some secondary effects, accounting for which can enhance the accuracy of the above approaches.

2. Leakage analysis background

Leakage analysis is typically performed as follows: the subthreshold model is used to estimate *leakage per unit micron* and then that is extended to estimate leakage over the entire chip. Typically, the stacking factor (leakage reduction from stacking of devices) is a first order component of this extension and serves to modify the total effective width of devices under analysis. Analysis can be viewed as the modification of this total width by the stacking factor.

2.1. Subthreshold leakage model

Most analytical works on leakage have used the BSIM2 subthreshold current model, which we have repeated here for convenience [3],

$$I_{sub} = Ae^{\frac{(V_{GS}-V_T-\gamma V_{SB}+\eta V_{DS})}{nV_{TH}}}\left(1-e^{-\frac{V_{DS}}{V_{TH}}}\right) \quad (1)$$

where V_{GS}, V_{DS} and V_{SB} are the gate-source, drain-source and source-bulk voltages respectively, V_T is the zero bias threshold voltage, V_{TH} is the thermal voltage (kT/q), γ is the linearized body-effect coefficient, η is the Drain Induced Barrier Lowering (DIBL) coefficient and A is given by:

$$A = \mu_0 C_{ox}\left(W/L_{eff}\right)V_{TH}^2 e^{1.8} \quad (2)$$

where μ_o is the carrier mobility, C_{ox} is gate capacitance/area, W is the width and L_{eff} is the effective gate length of the device.

The BSIM2 leakage model incorporates all the leakage behavior that we are presently concerned with. In summary, it accounts for the exponential increase in leakage with reduction in threshold voltage and gate-source voltage. It also accounts for the temperature dependence of leakage.

Calculating leakage current by applying Eq. (1) to every single transistor in the chip can be very time-consuming. To overcome this barrier, the following simple empirical model has been proposed [4]:

$$I_{leak} = I_{off}\frac{W_{tot}}{X_s}X_t \quad (3)$$

where I_{off} is the leakage current per micron of a single transistor measured from actual silicon at a given temperature, W_{tot} is the total transistor width (sum of all N and P devices). X_s is the empirical *stacking factor*. X_s is based on the observation that transistor stacks leak less than single devices. X_t is the temperature factor. X_t is used to scale I_{off} to the appropriate junction temperature of interest. The I_{off} value is typically specified at room temperature (therefore the need for a temperature factor to translate to the temperature of interest).

2.2. Stacking effects

A stack of 'OFF' transistors leaks less than a single device in the stack. This is primarily due to the self-reverse biasing of the transistor V_{GS} in the stack. Fig 1 illustrates the voltage allocation of four transistors in series. As one can see, V_{GS} is more negative when a transistor is closer to the top of the stack. In addition, the threshold voltages for the top three transistors are increased because of the reverse biased body-to-source voltage (body effect).

Both the self reverse biasing and the body effects reduce leakage exponentially as it is shown in Equation (1). Finally, the overall leakage is also modulated by DIBL effect for submicron MOSFETs. As V_{DS} increases, the channel energy barrier between the source and the drain is lowered. Therefore leakage current increases exponentially with V_{DS}.

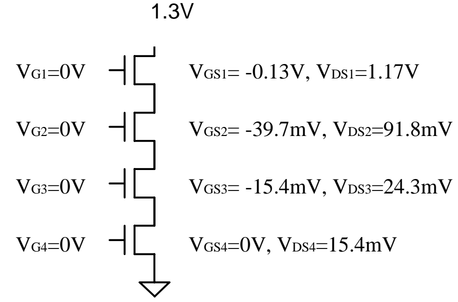

Figure 1: Voltage distribution of stacked transistors in OFF state

The combination of these three effects results in a progressively reduced V_{DS} distribution from the top to the bottom of the stack since all of the transistors in series must have the same leakage current (Figure 1). As a result of this significantly reduced V_{DS}, the effective leakage of stacked transistors is much lower than that of a single transistor. Analytical models for leakage reduction in stacks have been studied in [5].

Table 1: Stacking factors of 4-input Nand

	Min	Max	Avg
Stacking Factor Xs	1.75	70.02	9.95
Input Vector (a b c d)	(1 1 1 1)	(0 0 0 0)	

Table 1 quantifies the basic characteristics of the subthreshold leakage current for a fully static four-input NAND gate. The minimum leakage condition occurs for the '0000' input vector (i.e. all inputs a,b,c,d are at logic zero). In this case, all the PMOS devices are 'on' and the leakage path exists between the output node and ground through a stack of four NMOS devices. The maximum leakage current occurs for the '1111' input case when all the NMOS devices are 'on' and the leakage path, consisting of four parallel PMOS devices, exists between the supply and the output node. The stacking factor variation between the minimum and maximum leakage conditions reflects the magnitude of leakage dependence on the input vector. In the 4-input NAND case, we can conclude that the leakage variation between the minimum and maximum cases is a factor of about 40 (see Table 1). The values were measured using an accurate internal SPICE-like circuit simulator. The average leakage current was computed based on the assumption that all the 16 input vectors were equiprobable.

Figure 3 shows an exhaustive plot of the stacking factors for different input vectors in the 4-input NAND. The vectors have been arranged such than equal number of 'on' devices are clubbed together. It is apparent from the figure that Xs (and therefore the leakage current) in a stacked configuration is almost independent of which combination of devices is 'on', as long as the number of 'on' devices in the stack is constant.

In contemporary microprocessors, the stack depth is rarely more than 4 for performance reasons. From a leakage perspective, the higher the stack, the more the range in stacking factors. A simple

transistor will have a minimum, maximum and average stacking factor of 1. The challenge lies in being able to predict the stacking factor for large circuits and entire chips, without having to simulate the full circuits or chip. That is the focus of this paper.

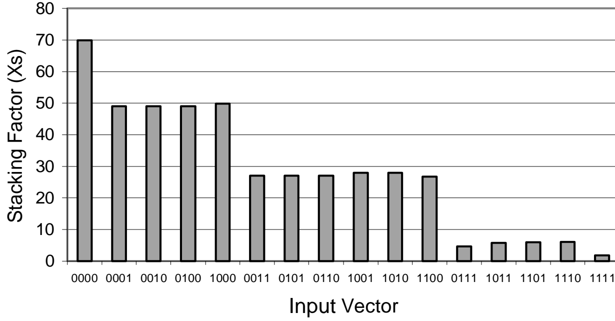

Figure 3: NAND4 stacking factors as a function of input vector

2.3. Xs analysis

It follows from Equation (3) that the quantities W_{tot} and X_s, when taken together, form an “effective leakage width.” This can be thought of as treating the device under study as a single inverter of width W_{tot}/X_s. Traditional prediction methodologies have focused on obtaining the W_{tot} term from schematics and then using some representative X_s factor. This latter factor has not been studied in great detail. Traditionally, X_s factors of 2 and 3 are used for cache and core predictions respectively.

2.4. Analysis methods

To enhance prediction accuracy, one of two optimizations can be performed. The total width could be obtained intelligently using some algorithm to analyze the circuit topology and report the “effective leakage width” (*cf.* Section 3) instead of total width. Such an algorithm would have to collapse serial and parallel structures into what would become a single inverter. Alternatively, an analysis algorithm could be implemented to examine cells or other structures on existing products and report realistic stacking factors for each circuit styles. Such an algorithm would utilize circuit simulations to average the leakage of cells for all possible input conditions (*cf.* Section 4: equivalent stacking factor method). This method is clearly more robust than the “effective total width” method as no optimized collapsation equations need to be derived. It should be noted that it is critical that the process file being used for cell characterization should accurately model leakage current. As leakage continues to emerge as one of the biggest design issues, we expect that the device models in process files will be enhanced to be very accurate with respect to the subthreshold region, and to accurately reflect the dependency of leakage on the important device parameters.

3. Effective leakage width method

The Effective Leakage Width (ELW) algorithm was developed based on extensive analysis of FUBs from a microprocessor core. The desired result was a method by which a FUB would be analyzed and the total device width and the effective leakage width reported. These widths could then be used in two ways. All FUBs for a future product could be analyzed in the same manner. Alternatively, the effective leakage width could be compared to the total device width to obtain stacking factors for different architectures (synthesized, datapath, array, etc.). These refinements of the X_s term would allow for more prediction accuracy for future products that might not have netlists ready for analysis at the time a leakage prediction is necessary.

3.1. Effective leakage function

To facilitate analysis, an effective leakage function (L-effective) which reports maximum and minimum drive strength of a gate was used. To obtain drive strength, the L-effective algorithm first recursively traverses the netlist tree and collapses parallel and serial networks using the standard equations shown in Figure 4. The L-effective algorithm then uses optimized collapsation equations to reduce the parallel and serial networks. Before studying how such equations might be derived, the reader is encouraged to examine the following example.

Consider an inverter, and a two-input nand gate composed of identical P and N devices as shown in Figure 4. The total width of the gates would increase in a 1:2 progression. However, the effective leakage width of each gate is nearly identical. To understand why, one must consider the effect of a statistical distribution of input patterns. The inverter can be thought to leak through the N device half the time (A = 0) and through the P device the other half (A = 1). The 2-input nand gate will leak through a single N device half the time, a two-stack of N devices a quarter of the time, and two P devices a quarter of the time. Let W_x be the width of an X-type device whose I_{off} is denoted as (L_x/μ). The coefficient S_{xn} is used to account for the reduction in leakage due to stacking for a stack of N X-type devices. Based on simple simulations of stacks, we can assume that $S_{x2} < 1/8$, $S_{x3} < 1/20$, and $S_{x4} < 1/45$ for these examples. The equations in Figure 4 can be readily derived assuming random inputs. In all diagrams, a patterns denoted * indicates cases where the leakage is underestimated as a result of the V_T drop of an ON device.

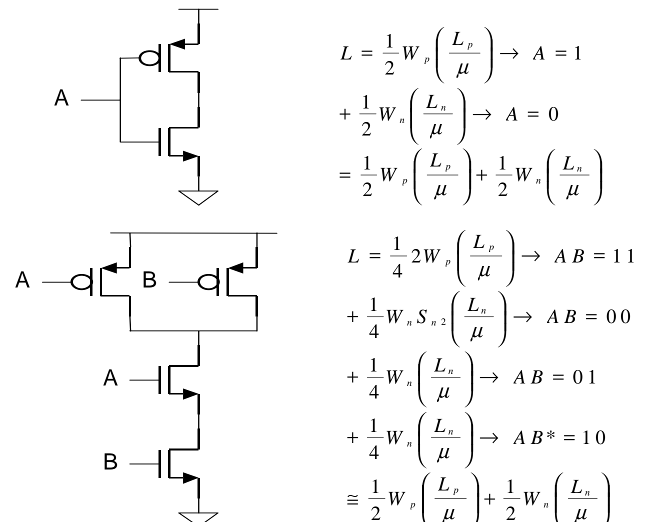

Figure 4: Inverter and NAND2 equivalent leakage derivation

In each of the above cases, the actual leakage comes out to be about the same. Therefore, the appropriate X_s to be used for the two-input nand gate would be slightly greater than 2. Analysis for the 3 and 4-input nand gates is similar and would require X_s factors greater than 3 and 4 respectively.

The analysis for nor structures is symmetric. A consequence of this analysis is that the equivalent leakage inverter of simple static gates is an inverter constructed using devices the same size as the devices in the static gate. This analysis can be extended to complex static gates and yields similar results. Complex gates have effective leakage widths slightly larger or slightly smaller than the equivalent inverter. By noting that there are always more simple static gates and incorporating the underestimation from disregarding the V_T drop on some patterns, it is possible to model all static gates as inverters of the same size as the component devices of those static gates. Analysis for the 3-input nand gate and a complex static gate is shown in Figures 5 and 6.

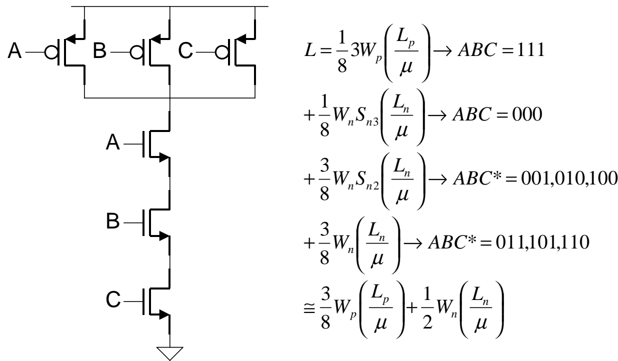

Figure 5: NAND3 equivalent leakage derivation

Analysis for domino structures is more complicated due to the mutually exclusive conditions of some signals. The keeper device can be thought of as a constant source of leakage whereas the discharge stack will leak differently depending on whether it is D1 or D2, as well as structural factors. For the purpose of this analysis, several test structures were analyzed.

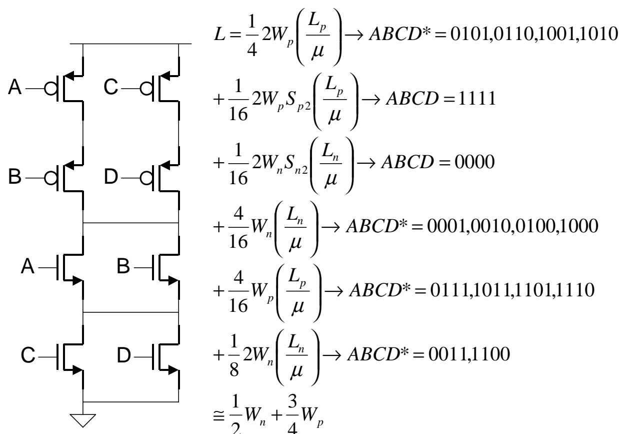

Figure 6: Complex static gate equivalent leakage derivation

3.2. Collapsation equations

Having obtained equivalent leakage targets for representative structures, the parallel and serial element collapsation equations were now altered to match the targets as closely as possible. In optimizing the equations, a linear error term was used such that the error to fits of single and dual stacks was weighted less than the error for three and four stacks. This ensures that the equations obtained provide the best fit for larger structures, which have the largest difference between total width and effective leakage width.

The choice of collapsation equation classes was arbitrary. The recursive nature of the L-effective routine makes a linear equation most suitable as each recursion forms a higher-order polynomial. It should also be noted that in all the above analysis, the size of all devices of each type within a gate was assumed to be identical. Addition error terms were added to handle the analysis of two series or parallel devices that differed in size. The error term here was a decaying exponential based on the sizing difference as the occurrence of two devices that differ greatly in size within a stack is rare. After many attempts to fit different equation classes, it was discovered that the standard L-effective equations were within 6% of the best alternative equations found.

3.3. Results of effective width analysis

The L-effective routines were run on 30 datapath and control fubs from the microprocessor core giving effective leakage widths between 0.35 and 0.95 times the total width. This corresponds to a stacking factor of between 5.72 and 2.1. Averaging over all the fubs, the resulting stacking factor was approximately 3. The L-effective routine provides separate information for P and N devices. While the ratio of total drawn N width to total drawn P width was 5:4, only 42% of the drawn N width was unstacked vs. 71% of the drawn P width. This leads to the consideration that P and N width could be analyzed for leakage separately.

4. Equivalent stacking factor method

An empirical alternative to calculating the effective leakage width is to estimate realistic stacking factors for real circuits using simulations of their component cells. Such analysis removes the mathematical problems of modeling stacking as a recursive equation.

4.1. Estimation technique

For small circuits, we can exhaustively simulate for bounds on the leakage current. In this section, we present a method to predict the stacking factor for the entire chip based on the stacking factors for the basic cell library.

Step 1 : Determine the primitive cells libraries used in the design. (In our case we used the basic, domino, complex and sequential cell libraries for the 0.18µm process. The basic library comprised of inverters, buffers, NAND, NOR etc. Similarly the domino, complex and sequential libraries comprised of corresponding circuits frequently used in contemporary microprocessors.)

Step 2 : Simulate all the cells exhaustively for minimum, maximum and average leakage currents and stacking factors. (This was done with the aid of automated tools that generated flat netlists, simulation files and automatic test pattern generators) .

Step 3 : Determine the freq. of cell usage in the entire design.

Step 4 : Compute a weighted minimum, maximum and average stacking factor using the following formulae.

$$\frac{W_{tot}}{X_{s,eff_i}} = \sum_{i \in cells} \frac{W_i \cdot f_i}{X_{s_i}} \quad (3)$$

where, $W_{tot} = \sum W_i \cdot f_i$, W_i being the total width of cell *I,* and f_i being the frequency of its occurrence in the design. The computation is done individually for the minimum, maximum and average staking factors.

The computation of effective stacking factors is based on the idea that each cell represents an individual leakage path from supply to ground. Of course, the assumption is that the overall minimum leakage is equal to the sum of the individual minimum leakage currents. This indeed is an over-simplified assumption. Input dependency and signal constraints would dictate that all cells would not have the minimum leakage generating input vector at their respective primary inputs. A similar argument holds for the maximum and average cases. The idea however is to generate heuristic bounds on leakage without having to simulate for signal constraints or statistical information. From our experiments we observe that the minimum stacking factor (corresponding to the maximum leakage) and the average stacking factor for large circuits is reasonably close to the actual value. However, errors of 50% were not uncommon in the maximum stacking factor (corresponding to the minimum leakage current) value.

4.2. Results of equivalent stacking factor

Figure 7 plots the stacking factors for the standard cell libraries used in the microprocessor design. Also, shown is the number of times a given cell was used in the design. (The cell names have been removed for proprietary reasons). The basic cells show a wide variation in stacking factors. This can be attributed to the fact that the basic cells have widely varying stack depths – from a stack of four in the 4-input NAND / NOR to a stackless inverter. With complex gates, which are generally a combination of two or three basic gates, the fluctuation is reduced since such extreme stacking cases tend to get amortized. The stacking factors seem to show lesser fluctuation as the logic depth and circuit complexity increase. Indeed, for the sequential circuits, which have multiple basic cells and latches, the stacking factors seem to converge.

The sizes of cells plotted in Figure 7 are the averaged sized cells in the library. Of course, each cell will have multiple sized instances available for use depending on drive strength requirements. One approach would be to exhaustively simulate all sizes and find global stacking factors using the method previously described. Alternatively, our experiments indicate that the stacking factors averaged over different sizes of the same cell tend to be fairly close to the stacking factor of the average sized cell. This observation can be used to simulate exhaustively for averaged sized cells and compute a weighted global stacking factor based on frequency and stacking data for averaged sized cells only.

Based on the above algorithm we obtained the global minimum, maximum and average stacking factors for a contemporary microprocessor as 1.73, 4.48 and 2.64, respectively. It is practically impossible to simulate the whole microprocessor exhaustively to determine the input-dependent leakage bounds and compare the stacking factors computed thereof with our predicted values. Therefore, we simulated large portions (FUB level) of the processor, computed the stacking factors from exhaustive leakage measurements and compared it with our predicted values.

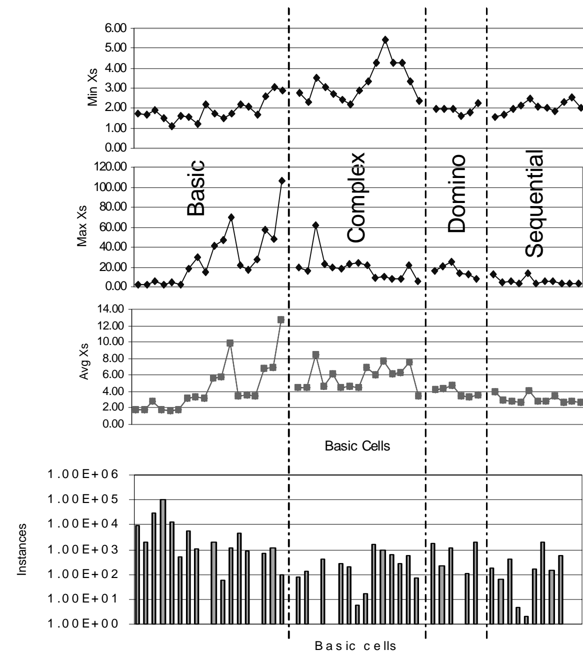

Figure 7: Stacking factors for standard cell libraries

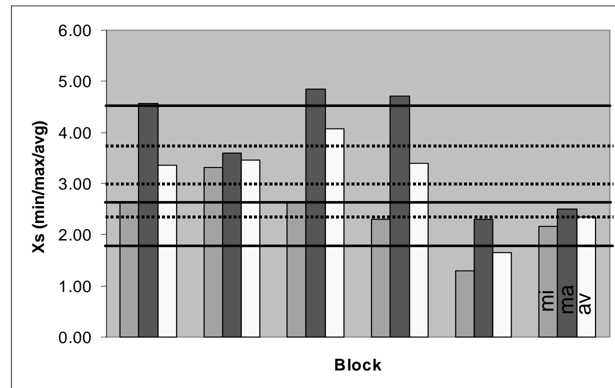

Figure 8: Predicted and simulated stacking factors

Figure 8 illustrates the prediction accuracy of our methodology. Each "Block" corresponds to a large FUB that was exhaustively simulated and its minimum, maximum and average stacking factors (indicated as mi, ma, av respectively) were determined.

The bold line corresponds to the global stacking factors predicted for the entire microprocessor. The dotted line corresponds to the stacking factors averaged for our sample circuit blocks. We conclude that our method results in a pessimistic bound for leakage for each of the minimum, maximum and average cases. The measured stacking factors for some circuits (particularly the last two) fall significantly below the global predictions. This can be attributed to 2 factors. (i) The FUBs were dominated by inverters and other library cells where the stacking effect was minimal (ii) The assumption that leakage per micron is fixed - is a simplification. This is elaborated in the next section.

5. Additional considerations

A fundamental assumption up to this point has been that all devices in the circuit or chip under consideration have uniform parameters. This is not always the case. This section provides a brief overview of additional effects that lead to the assumption of constant I_{off} (leakage per micron of device width) being a simplification. Accounting for these effects in the framework of the approaches discussed above is feasible but a detailed discussion of this is outside the scope of this paper.

The first issue is that the stacking factor is heavily dependent upon process skew. Experimental evidence shows that the stacking reduction for three series devices varies largely over the device skew of the stack components. This is expected since the higher leakage current accentuates the physical effects that determine the stacking factor (*cf.* Section 2.1). This complicates matters two fold. First, the analytical method (*cf.* Section 3) is based on certain assumptions regarding stacking reductions for two, three, and four-stacks of devices. It is initially unclear if the optimized equations developed for collapsing a device tree should include this variation or simply assume a best-case reduction. Because stacking effect increases with device leakage, it is reasonable to assume that using the highest stacking effect numbers, although analytically incorrect for slow and typical skews, would yield the best results for the highest leakage parts. It is after all these parts for which leakage prediction matters most.

The other two effects are to do with devices that differ from the typical or nominal device for the process. The first of these is the *"long channel effect"*. Devices with channel lengths longer than the nominal minimal length for the process will have different leakage current per micron (I_{off}) than the nominal device. If a uniform I_{off} is used for the entire chip, then an additional compensating factor needs to be used to account for the longer devices. Depending on whether the longer device uses halo implants or not, the relationship between leakage and channel length will be different. In either case, however, this relationship will be reflected in simulations if the device models are accurate with respect to this effect.

The final effect is the *"narrow channel effect"*. This refers to the fact that for channels narrower than a certain threshold (i.e. comparable to the size of the depletion layer at the silicon surface [6]), I_{off} (leakage per micron) is not constant with respect to the channel width. For narrow widths, the threshold voltage increases since some of the gate-induced space-charge is lost in fringing field. Thus, narrower widths will have a smaller I_{off}. In the simulation based approach, one way to account for this effect is to bin the gates based on their widths, and then use a lower I_{off} for gates that have narrow widths.

6. Summary

The work presented in this paper shows that the stacking factor can be analytically obtained and analyzed for use in leakage prediction. In the Effective Leakage Width method, the L-effective function served as a first-order estimate of the effective leakage width of a circuit. On an average, that is within 6% of an optimized equation approach. Use of this method also allows the possibility of separating P and N device width for analysis to improve accuracy. The Effective Staking Factor approach shows that stacking factors can be empirically obtained for component circuit blocks or gates and can then be heuristically extended to model larger circuits and even for full-chip estimates. Finally, additional relevant effects are also described. Accounting for these can lead to greater accuracy of the leakage estimates. It is hoped that this paper provides the reader with the background, as well as working knowledge, for a topic that is becoming an increasingly important design issue.

7. Acknowledgments

We would like to acknowledge Adam Brand for providing leakage measurement data from test wafers.

8. References

[1] R. X. Gu and M. I. Elmasry, "Power Dissipation Analysis and Optimization of Deep Submicron CMOS Digital Circuits", IEEE Journal of Solid-State Circuits, vol. 31, no. 5, May 1996, pp. 707-713

[2] M. C. Johnson, D. Somasekhar and K. Roy, "Models and Algorithms for Bounds on Leakage in CMOS Circuits", IEEE Trans. On CAD of Integrated Circuits, vol. 18, no. 6, June 1999, pp. 714-725

[3] B. J. Sheu, et al., "BSIM: Berkeley Short-Channel IGFET model for MOS Transistors", IEEE Journal of Solid-State Circuits, vol. 22, Aug. 1987, pp. 558-566

[4] Scott Thompson, Design, Test and Technology Conference, 1998, Intel Corp.

[5] Z. Chen, et al., "Estimation of Standby Leakage Power in CMOS Circuits Considering Accurate Modeling of Transistor Stacks", ISLPED98, pp. 239-244.

[6] R. S. Muller and T. L. Kamins, *Device Electronics for Integrated Circuits*, John Wiley & Sons, 1986.

[7] HSPICE manual, Meta Software Inc.