

Accurate Estimation of Vector Dependent Leakage Power in the Presence of Process Variations

Romana Fernandes, Ranga Vemuri

University of Cincinnati, Cincinnati, OH, USA
fernanra@email.uc.edu, ranga@ececs.uc.edu

Abstract— With the increasing importance of run-time leakage power dissipation (around 55% of total power), it has become necessary to accurately estimate it not only as a function of input vectors but also as a function of process parameters. Leakage power corresponding to the maximum vector presents itself as a higher bound for run-time leakage and is a measure of reliability. In this work, we address the problem of accurately estimating the probabilistic distribution of the maximum run-time leakage power in the presence of variations in process parameters such as threshold voltage, critical dimensions and doping concentration. Both sub-threshold and gate leakage current are considered. A heuristic approach is proposed to determine the vector that causes the maximum leakage power under the influence of random process variations. This vector is then used to estimate the lognormal distribution of the total leakage current of the circuit by summing up the lognormal leakage current distributions of the individual standard cells at their respective input levels. The proposed method has been effective in accurately estimating the leakage mean, standard deviation and probability density function (PDF) of ISCAS-85 benchmark circuits. The average errors of our method compared with near exhaustive random vector testing for mean and standard deviation are 1.32% and 1.41% respectively.

I. INTRODUCTION

Technology scaling into the deep nanometer geometries has seen an increase in process variability due to lithographic inaccuracies and this problem magnifies with every generation. Though the advantages of scaling are evident and it is necessary to continue with this trend, it is also important to consider variability data in order to make accurate estimations of performance measures. It has become highly important to accurately estimate performance measures such as timing, leakage power and noise in the design phase itself in order to increase the parametric yield after fabrication. It has been shown that a 30% variation in circuit frequency induced by parameter variations can result in nearly 20X variation in leakage power [1]. Variations in process parameters such as channel length (L_{ch}), threshold voltage (V_{th}), channel doping concentration (N_{ch}), gate oxide thickness (T_{ox}) and gate width (W_{gate}) are the major contributors to variability in timing and leakage power. It has also been shown that this dependence on process parameters is linear in the case of timing [2] and exponential in the case of leakage power [3].

Substantial work has been done in the area of analysis, estimation and optimization of statistical static timing [2],[4],[5],[6],[7],[8] but statistical leakage power measurement and optimization is still emerging [3],[9],[10],[11].

Moreover, statistical leakage power estimation has become increasingly important because of the exponential dependence of leakage on several process parameters. The traditional methods of leakage power estimation, such as nominal analysis, underestimate leakage power by a large margin whereas corner analysis, on the other hand, overestimates leakage power dissipation. As a result of under-estimation, there may be a failure in meeting the power yield and this may also cause reliability issues in chips that do meet the power yield during their life span. Over-estimation on the other hand may result in designing unnecessary guard bands and failure to meet timing specifications [12]. Therefore, it is extremely important to accurately estimate leakage power as a function of process variations.

The terms leakage power and leakage current are synonymously used in this paper since leakage power is the product of leakage current which is the variable term and the supply voltage which is treated as a constant in this work. Leakage current comprises of several components [13] of which sub-threshold and gate leakage are the prominent ones [14]. With leakage power contributing more than 55% of the total power in present day technologies (32nm and beyond) and this trend being predicted to only increase in future technologies [15],[16], accurate estimation methods considering all the important types of leakage has become a necessity.

A chip can settle in an idle state for a considerable amount of time or can enter stand-by mode several times during its life-span. It enters an idle state or stand-by mode with a different set of input vectors each time. In this scenario, it is not assured that the leakage power specifications are met each time. Excessive leakage power dissipation for a long period of time results in a drastic increase in the thermal profile of most chips during their operation and makes them susceptible to failure. Maximum leakage power estimation provides an upper bound and guarantees that the design constraints are met irrespective of the circuit input state [17]. Maximum leakage can also be used to find hot-spots in a physical design. It can also be used to estimate the worst case battery life of a portable device [17].

Traditionally, leakage power was considered important only in the stand-by mode whereas dynamic power was considered important in the active mode of operation. But, due to the shrinking physical dimensions, the contribution of dynamic power to the total power has reduced with the

growth of leakage power [18]. With its increasing importance, leakage contribution during run-time has become a deciding factor to determine the maximum power bound for reliability of a chip and as a measure of its ever increasing thermal profile. This leakage power is called run-time leakage [19] and is becoming as important as dynamic power dissipation [20]. Runtime leakage is vector dependent and changes each time the input vector changes. It has been well established that the maximum leakage of a circuit can be greater by a few orders of magnitude than the minimum leakage and is dependent on the input vectors associated with them [21]. But, this dependence of maximum leakage power on input vectors alone is not correct anymore. In the presence of variability, the input vectors that cause the maximum or the minimum leakage current changes. Hence, maximum leakage now depends not only on the input vectors but also on process parameter variations.

Though some work has been done in the area of statistical leakage power estimation [3],[9],[10],[11], no methods have been proposed so far to estimate the run-time maximum leakage power distribution as a function of both process variations and input vectors. The work presented in this paper addresses this problem and gives an approach to accurately estimate it. This method can also be extended to determine the minimum leakage vector which can be used as a sleep vector in standby mode.

II. MOTIVATION: DEPENDENCE OF LEAKAGE ON INPUTS IN THE PRESENCE OF VARIATIONS

Consider a CMOS inverter in 32nm technology, with nominal values of process parameters given by Table I [22]. This inverter has maximum leakage current when input state is ‘0’. However, when process parameters are changed to PMOS $L_{ch} = 1.837\lambda$ and NMOS $L_{ch} = 1.985\lambda$ with other parameters at their nominal values, the maximum leakage inducing input is ‘1’. In fact, when 500 Monte Carlo simulations vary the process parameters around their nominal values with variability given by Table II [15], we observe that the mean leakage when the input vector is ‘1’ is far greater than the mean leakage when the input is ‘0’. In fact, the mean leakage for input ‘1’ is almost double when compared to the mean leakage for input ‘0’. The reason for the inverter to change its maximum leakage state is because the I_{sub} of PMOS in its OFF state (input ‘1’) exceeds that of NMOS in its OFF state (input ‘0’) when the channel length of PMOS is less than the NMOS channel length. This is because of the increased sensitivity of PMOS sub-threshold current to channel length variations as compared to NMOS [23].

This example can be extended to a simple combinational circuit as shown in Figure 1. In the absence of process variations, the maximum leakage state for this circuit is ‘01’ whereas in the presence of variations, Monte Carlo simulations show that the input state for maximum leakage changes to ‘11’. This change in input vector changes the total leakage from INV (0) + INV (1) + AND (101) to 2*INV (1) + AND (001). Our experiments have shown that, for

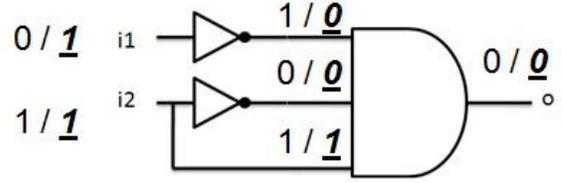


Fig. 1. Example Circuit

TABLE I
NOMINAL VALUES OF PROCESS PARAMETERS IN 32NM TECHNOLOGY

Process Parameters	Nominal Values
NMOS L_{ch}	32nm (2λ)
PMOS L_{ch}	32nm (2λ)
NMOS W_{gate}	112nm (7λ)
PMOS W_{gate}	224nm (14λ)
NMOS T_{oxe}	1.65nm
PMOS T_{oxe}	1.75nm
NMOS V_{th}	0.5088V
PMOS V_{th}	0.4500V
NMOS N_{ch}	4.12e+18
PMOS N_{ch}	3.07e+18

larger cells, the maximum leakage vector differs from the one calculated using the conventional method in cases where the effects of process variations dominate the bias voltage effects or where there is increased sensitivity to a process parameter.

This shows that the maximum leakage inducing input vector is highly dependent on the specific values of process parameters. In this paper, we are interested in finding the maximum leakage inducing vector in the presence of such process variations and use this vector to estimate the maximum leakage power distribution.

III. LEAKAGE POWER DEPENDENCE ON PROCESS PARAMETERS

Two components dominate leakage - sub-threshold (I_{sub}) and gate leakage (I_{gate}) currents. Sub-threshold leakage is the most dominant leakage mechanism occurring in the OFF state of a transistor and gate leakage is the most dominant leakage mechanism in its ON state [24]. Considering both these types of leakage and the interactions between them, the total leakage of a circuit can be approximated using (1).

$$I_{total} = I_{sub} + I_{gate} \quad (1)$$

I_{sub} and I_{gate} vary with variation in process parameters such as channel length (L_{ch}), threshold voltage (V_{th}), channel doping concentration (N_{ch}), gate oxide thickness (T_{ox}) and gate width (W_{gate}). The equation for sub-threshold leakage current is given by (2) [24]. The threshold voltage is given by (3) and the dependence of L on Vth is given by (4) [11]. For definition of variables refer [24].

$$I_{sub} = I_0 \exp\left(\frac{V_{gs} - V_{th}}{\eta V_t}\right) \left[1 - \exp\left(\frac{-V_{ds}}{V_t}\right)\right] \quad (2)$$

$$V_{th} = V_{fb} + |2\phi_p| + \frac{\lambda_b}{C_{ox}} \sqrt{2qN_{ch}\epsilon_s(|2\phi_p| + V_{sb}) - \lambda_d V_{ds}} \quad (3)$$

TABLE II
VARIABILITY DATA FOR 32NM TECHNOLOGY

Process Parameters	Variability
Critical Dimensions (Lch, Wgate, Tox)	12%
Threshold Voltage (Vth) (Including doping variability)	58%

$$V_{th,i} = V_{th0} + 0.05 - V_{dd} e^{-\delta L_{e,i}} \quad (4)$$

The equation for gate leakage current is given by (5) [16].

$$I_{gate} = WA_g \left(\frac{V_{dd}}{T_{ox}} \right)^2 \left[\exp \left(-B_b \frac{T_{ox}}{V_{dd}} \right) \right] \quad (5)$$

500 Monte Carlo simulations performed on an example ISCAS-85 benchmark circuit, C1355 with 546 gates (32nm technology using predictive models [22]) is shown in Figure 2. The nominal values for the process parameters that were varied are given by Table I. Typical variation data is shown in Table II [15]. Such significant variations in process parameters will induce large variations in performance and power.

It was observed in our experiments that, the mean of the distribution is almost four times the nominal value for leakage current in the case of this benchmark circuit. When the parameters were individually varied in hspice and Monte Carlo simulations were performed, it was observed that the leakage current of this circuit was dependent on the process parameters given by the relationships (6) and (7). This agrees with the analytical expressions and previous work in [3],[9]-[11].

$$I_{sub} \propto \frac{1}{e^{L_{ch}}}, \quad I_{sub} \propto \frac{1}{e^{V_{th}}}, \quad I_{sub} \propto \frac{1}{N_{ch}}, \quad I_{sub} \propto W \quad (6)$$

$$I_{gate} \propto \frac{1}{T_{ox}^2}, \quad I_{gate} \propto W \quad (7)$$

Due to the dependence of leakage current represented by equations (6),(7), leakage current can no longer be represented by a single nominal or a corner case value. It is in fact a log-normal probabilistic distribution with mean μ which is significantly greater than the nominal value and has standard deviation D [3].

IV. MAXIMUM LEAKAGE VECTOR IN THE PRESENCE OF PROCESS VARIATIONS

Exact determination of the input vector that induces maximum (or minimum) leakage requires exhaustive simulations with all input vectors. Several approaches have been proposed to estimate this vector with reduced complexity focusing on sleep vectors for leakage minimization in stand-by mode [25], [26], [17]. Rao et al. [27] proposed an approach where the minimum (or maximum) leakage vector is determined by taking cell functionalities into account but the effects of process variations were not considered. This technique cannot be used in the presence of process variations, as demonstrated in Section II. In this work, we propose

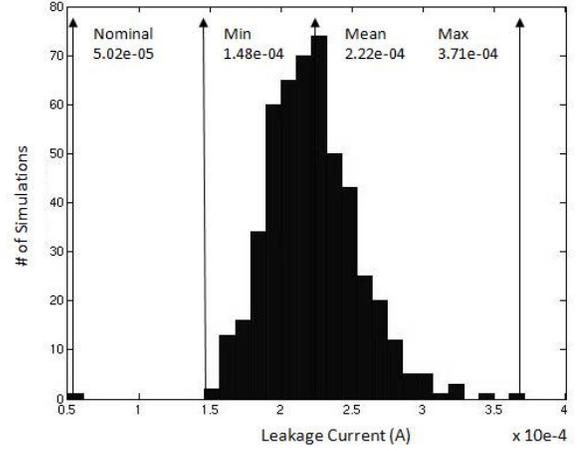


Fig. 2. Leakage power distribution - C1355

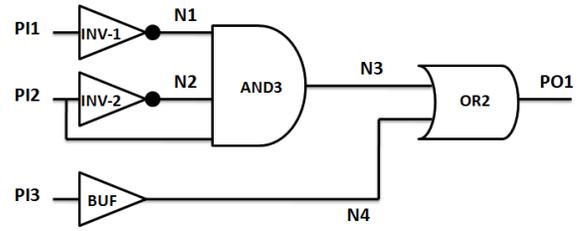


Fig. 3. Illustrative example

a heuristic to accurately estimate the maximum leakage vector considering both cell functionalities as well as process variations. The vector thus obtained is then used to determine the maximum sum leakage distribution of a circuit. This method can also be modified to find the minimum leakage vector and hence the minimum sum leakage distribution. Below, we adapt some of the definitions Rao et al. provided in [27] to accommodate process variations. The definitions are also supported by an illustrative example given by Figure 3. Only the most dominant leakage state is considered for the purposes of illustration as opposed to the implemented algorithm which considers the top two maximum leakage states. This was done to keep the illustration simple. The notations used in the illustration are given by Table III.

A. Definitions

It is assumed that the circuit under consideration can be decomposed into standard cells. A graph is constructed with cells as the nodes and nets as the edges. The leakage value associated with each state of a standard cell is calculated as a weighted sum of the mean leakage current and the standard deviation of that particular state as given by (8). The mean and standard deviation are determined by performing 500 Monte Carlo simulations by varying the process parameters as shown in Tables I and II for each input combination for each cell. The mean and standard deviation of leakage for an inverter and a two input OR gate are given by Table IV. It is seen in Figure 2 that the probability density function of a circuit has an increased average leakage value given by its mean and a spread represented by the standard deviation. When λ is chosen to be equal to 1, the heuristic chooses

TABLE III
NOTATIONS

Symbol	State
0	State '0'
1	State '1'
X	Don't Care
C	Conflict between '0' and '1'

TABLE IV

LEAKAGE MEAN AND STANDARD DEVIATION OF STANDARD CELLS

Cell	Mean (10^{-7})	Standard Deviation (10^{-7})
INV (Input '0')	2.25	5.26
INV (Input '1')	4.62	12.46
OR2 (Input '00')	7.79	14.41
OR2 (Input '01')	5.00	7.65
OR2 (Input '10')	2.52	4.54
OR2 (Input '11')	2.03	4.36

cells with greater mean values but completely ignores the standard deviation and hence the spread of the leakage profile. When λ is chosen to be equal to zero, the mean leakage value is completely ignored and the heuristic chooses cells with greater spread. This could result in choosing cells with smaller means. Hence, λ is chosen to be 0.5 in our experiments to give equal importance to both leakage mean and standard deviation.

$$P_i = \lambda \mu_i + (1 - \lambda) D_i \quad (8)$$

where, P_i is the probabilistic cell leakage in input state i , μ_i is the mean leakage of the cell in input state i , D_i is the standard deviation of leakage in input state i , λ , the weighting factor is a real number and $\lambda \in [0, 1]$

Node Controllability: The controllability of a node in a circuit is defined as the minimum number of inputs that have to be assigned to specific values in order to force the node output to a specific state. Every node is assigned two values, CC0 (controllability to force the cell output to 0) and CC1 (controllability to force the cell output to 1).

Controllability List: The constraints imposed on the primary input vector in order to force a node output to a specific state is defined as the controllability list. The two constraint lists associated with every node output is CC0 list and CC1 list.

For example in Figure 3, CC0 list of N3 would require either N1, N2 or PI2 to be '0'. We choose the net which has the least fanout. In case of equal fanout nodes, we pick a net randomly and set it to '0'. In this example we have chosen N1=0 which requires PI=1 and PI2=X and PI3=X on the primary input lines. Table V gives the CC0 and CC1 lists for all internal nets in the example circuit.

Probabilistic Worst Input Condition (PWIC): The worst input condition for a cell represents the minimum number of primary inputs and their specific values that force the cell to its highest leakage in the presence of process variations.

TABLE V
DETERMINATION OF CC0, CC1, CC0 LIST AND CC1 LIST

Net	CC0 list - PI1 PI2 PI3 (CC0)	CC1 list - PI1 PI2 PI3 (CC1)
N1	1XX (1)	0XX (1)
N2	X1X (1)	X0X (1)
N3	1XX (1)	0CX (C)
N4	XX0 (1)	XX1 (1)

TABLE VI

DETERMINATION OF PWIC

Cell	PWIC1
INV-1	1XX
INV-2	X1X
AND3	0CX
BUF	XX1
OR2	1X0

It was observed in our experiments that every gate has two dominant worst leakage states. The remaining states dissipate far less leakage compared to these two dominant states. It was also observed that when the top two worst leakage states were considered the accuracy of the algorithm was improved. Hence, we define Worst Input Conditions, PWIC1 and PWIC2 where PWIC1 yields the worst value for leakage and PWIC2 yields the second worst value.

For example, in the case of cell OR2, PWIC1=00 and PWIC2=01. To force the inputs of OR2 to PWIC1, N3 and N4 must be forced to '0'. This translates to CC0 of N3 & CC0 of N4 which is equal to 1XX & XX0 = 1X0. Table VI gives the PWIC1 constraints for all the cells in the example circuit.

Probabilistic Worst Leakage Advantage (PWL A): If the PWIC of a cell cannot be satisfied, the cell may settle into one of its low leakage states. The increase in leakage when a cell is forced to its PWIC can be quantified by a metric called Probabilistic Worst Leakage Advantage. PWLA is given by the difference in the leakage of the worst leakage state and the average of the low leakage states and can be represented using (9). Since we have chosen two dominant PWICs, we define PWLAs associated with each of them. Table VII shows the PWLAs for the standard cells in the illustration.

$$PWL A = P_{PWIC} - AVG(P_{PLLS}) \quad (9)$$

where, P_{PWIC} is the probabilistic cell leakage in PWIC, P_{PLLS} is the probabilistic cell leakage of low leakage states.

For example, in the case of cell OR2,

$$PWL A1 = P_{00} - 0.5(P_{11} + P_{10})$$

$$PWL A2 = P_{01} - 0.5(P_{11} + P_{10})$$

where, P_{ij} is the probabilistic leakage when the cell is in state ij .

Conflicting and Dominated Cells: When the PWIC of a cell is satisfied, it will result in certain nodes in the circuit being forced to particular states because of the way the gates

TABLE VII
DETERMINATION OF PWLA

Cell	PWLA1 (10^{-6})
INV-1	0.957
INV-2	0.957
AND3	1.363
BUF	1.028
OR2	1.549

TABLE VIII
CONFLICTING AND DOMINATED CELLS

Cell	Conflicting Cells	Dominated Cells	Infeasible?
INV-1	AND3	-	NO
INV-2	-	-	NO
AND3	INV-1, OR2	-	YES
BUF	OR2	-	NO
OR2	AND3, BUF	INV-1	NO

function. Cells C_i and C_j are said to be conflicting cells if they have opposing requirements for primary input P_m . Cell C_i is said to dominate cell C_k if the input requirements for C_k to be in its worst input state is a subset of that required for cell C_i to be in its worst input state. Table VIII shows the conflicting and dominated cells for the illustrative example. For example, OR2 and BUF in the illustration are conflicting cells because BUF requires PI3=1 to force it to its PWIC while OR2 requires PI3=0 to force it to its PWIC. On the other hand OR2 dominates INV-1 because satisfying PWIC1 of OR2 (1X0) would also mean satisfying the PWIC1 of INV-1 (1XX).

Cost Function: When the PWIC of a cell is satisfied, the PWICs of its conflicting cells are violated, while the PWICs of its dominated cells are satisfied. Therefore, the cost of satisfying the PWIC of a cell C_i is calculated as given by $Cost(C_i)$. As a special case, costs of cells which have infeasible requirements for a primary input are assigned a large negative value, as they can never be forced to their PWIC. The costs of the standard cells in the illustration is given by Table IX.

$$Cost(C_i) = PWLA(C_i) + \sum(PWLA(DominatedCells(C_i))) - \sum(PWLA(ConflictingCells(C_i)))$$

For example, $Cost(BUF) = PWLA(BUF) - PWLA(OR2)$. The costs associated with both the PWICs are calculated in our method.

B. Heuristic to Determine the Maximum Leakage Vector

This section gives the outline of HeuristicMax, the heuristic proposed to determine the maximum leakage vector in the presence of process variations. The heuristic is given by Algorithm-1.

The complexity of determining the controllability and controllability lists in HeuristicMax was reduced by sorting the nets in the increasing order of its depth. A cell in the higher level was processed only after processing all the cells in the lower levels. By sorting, the need to back traverse all

Algorithm HeuristicMax(Circuit Graph)

Input: CMOS circuit represented by a graph with cells as nodes and nets as edges

Output: Primary input vector for maximum leakage in the presence of process variations

Variables: Cells[cells], Nets[nets], Inputs[PI]

```

forall Nets do
    Sort in increasing order of depth from the primary
    inputs;
end
for  $i=1$  to nets do
    Determine CC0 and CC1 lists;
end
for  $i=1$  to cells do
    Generate PWICs;
    Compute PWLAs using table lookup;
end
for  $i=1$  to cells do
    Generate conflicting cell list;
    Generate dominated cell list;
end
Make all primary inputs undefined;
Create a cell list with all cells;
while (cell list is not empty) and (primary inputs
undefined) do
    Compute cost functions;
    Assign a large negative value to the cost of cells
    with infeasible inputs;
    Satisfy the input constraint of the cell with the
    highest cost;
    Remove the selected cell and its dominated cells
    from the cell list;
    Remove the conflicting cells from the cell list and
    push them to a violated cell list;
    Update conflicting and dominated cell lists for the
    remaining cells in the cell list;
end
for  $i=1$  to PI do
    if Inputs( $i$ ) = X then
        Determine Leakage(1) = Total leakage
        contributed by cells with unsatisfied PWIC
        when Inputs( $i$ )=1;
        Determine Leakage(0) = Total leakage
        contributed by cells with unsatisfied PWIC
        when Inputs( $i$ )=0;
        if Leakage(1) > Leakage(0) then
            Inputs( $i$ ) = 1;
        else
            Inputs( $i$ ) = 0;
        end
    end
end

```

Algorithm 1: Algorithm HeuristicMax

TABLE IX
DETERMINATION OF CELL COST

Cell	Cost (μA)
INV-1	-0.406
INV-2	0.957
AND3	LARGE NEG
BUF	-0.520
OR2	0.114

TABLE X
UPDATED VALUES

Cell	Conflicting Cells	Dominated Cells	Cost
INV-1	AND3	-	-0.406 μ
AND3	INV-1, OR	-	LARGE NEG
BUF	OR	-	-0.520 μ
OR2	AND3, BUF	INV-1	0.114 μ

the way to the primary inputs was eliminated and hence the complexity was reduced.

In the illustrative example, INV-2 is selected and its input constraint - X1X is satisfied in the first iteration. The conflicting and dominated cell lists are updated and the new costs are determined for the remaining cells as given by Table X. In the second iteration OR2 is selected and its input constraint - 1X0 is satisfied which finally defines the primary inputs as 110.

V. ESTIMATION OF TOTAL LEAKAGE CURRENT DISTRIBUTION

The leakage current distribution of a standard cell in state i can be represented by a log-normal distribution with mean μ_i and standard deviation D_i . This distribution has a corresponding normal distribution having mean m_i and standard deviation σ_i obtained by taking the natural logarithm of all the points in the log-normal distribution [28].

A. Sum of Log-normals

A standard cell in state i with Gaussian mean m_i and Gaussian standard deviation σ_i , has probability density function represented by (10).

$$f(x) = \frac{1}{x\sigma\sqrt{2\pi}} e^{-\frac{(\ln(x)-m_i)^2}{2\sigma_i^2}} \quad (10)$$

Given the probability density function of all the standard cells, we can determine the total leakage power distribution of a circuit. In theory, the sum of log-normals is not known to have a closed form. An approximation of the same can be made using the Fenton-Wilkinson's method of estimating the sum of several log-normal distributions [28]. Given $f(x)$ of all the standard cells in the circuit, the sum leakage distribution is given by equating the first two moments, $\alpha 1$ and $\alpha 2$. Equation (11) gives the relationship between $\alpha 1$, $\alpha 2$, m_i and σ_i . Equations (12)-(13) give the relationship between μ_i , D_i , m_i and σ_i . The mean μ and variance D^2 of the sum log-normal distribution are given by (14)-(15) and the distribution function is given by (16).

$$\alpha 1 = e^{m_i} e^{\sigma_i^2/2}, \quad \alpha 2 = e^{2m_i} e^{2\sigma_i^2} \quad (11)$$

$$\mu_i = \alpha 1 = e^{m_i} e^{\sigma_i^2/2} \quad (12)$$

$$D_i^2 = \alpha 2 - \alpha 1^2 = e^{2m_i} e^{\sigma_i^2} (e^{\sigma_i^2} - 1) \quad (13)$$

$$\mu = \mu_1 + \mu_2 + \dots + \mu_i + \dots + \mu_n \quad (14)$$

$$D^2 = D_1^2 + D_2^2 + \dots + D_i^2 + \dots + D_n^2 \quad (15)$$

$$f(x) = \frac{1}{x\sigma\sqrt{2\pi}} e^{-\frac{(\ln(x)-m)^2}{2\sigma^2}} \quad (16)$$

B. Overall Approach

1. Calculate the maximum leakage vector using the heuristic described in Section IV.B.
2. Set the primary inputs of the circuit to the maximum leakage vector.
3. Forward-propagate the primary inputs and define the input states of all the gates in the circuit.
4. Estimate the probabilistic leakage distribution corresponding to the maximum vector using the method described in Section V.A.

VI. RESULTS

The approach described in this paper was implemented using C++ and was tested on ISCAS-85 benchmark circuits using predictive models for 32nm technology [22]. The nominal values for the process parameters that were varied are given by Table I. Typical variation data is shown in Table II. Vdd for 32nm technology is 0.9V. The cells were characterized using 500 Monte Carlo simulations in hspice.

A. Leakage Power Distribution Corresponding to the Maximum Leakage Vector in the Presence of Variations

This sub-section gives the results for the heuristic implemented to determine the leakage power distribution corresponding to the maximum leakage vector in the presence of process parameter variations. The results are given by Table XI. This method was compared against random vector testing using 100,000 random vectors except for C17 which has 5 primary inputs, was exhaustively tested with 2^5 input vectors. The average error for mean leakage current and standard deviation was found to be 1.32% and 1.41% respectively. Only the smaller benchmarks were verified against Monte Carlo simulations using 100 random vectors due to the large run-time of Monte Carlo simulations. The results are shown in Table XII. In the presence of process parameter variations, the mean leakage was several times larger than the nominal analysis as expected. The comparison results are given in Table XIII. The mean leakage power obtained using our method was on an average 3.4X greater than the leakage current obtained using the method in [27]. It was observed that when the top two worst leakage states were considered instead of one worst leakage state, the accuracy

TABLE XI

COMPARISON OF MEAN AND STANDARD DEVIATION OBTAINED USING OUR HEURISTIC (H) WITH RANDOM VECTOR TESTING (R)

Benchmark	H μ (μ A)	R μ (μ A)	Error (%)	H D (10^{-5})	R D (10^{-5})	Error (%)
C17	2.4	2.4	0.0	0.27	0.27	0.0
C432	65.8	70.5	6.6	1.28	1.38	7.2
C499	123.3	127.5	3.3	1.52	1.60	5.0
C880	144.6	154.7	6.5	1.77	1.85	4.5
C1908	320.4	325.9	1.7	3.01	3.01	0.0
C2670	530.2	526.0	-0.1	3.46	3.50	1.3
C3540	709.7	731.3	2.9	4.14	4.21	1.6
C5315	997.9	979.1	-1.9	4.87	4.79	-1.7
C6288	653.2	642.2	-1.7	2.84	2.82	-0.1
C7552	1451.9	1394.4	-4.1	6.02	5.80	-3.7
Average	-	-	1.32	-	-	1.41

TABLE XII

COMPARISON OF HEURISTIC MEAN WITH MONTE CARLO MEAN

Benchmark	H μ (μ A)	MC μ (μ A)	Error(%)	Runtime Saving(X)
C17	2.4	2.4	0.0	30,965
C432	65.8	66.5	1.1	59,520
C499	123.3	125.2	1.5	195,818
C880	144.6	148.5	2.6	337,042

of the algorithm improved on an average by 3.2% for mean and 2.7% for standard deviation.

B. Pessimistic Approach

A pessimistic approach was also implemented in which the maximum sampled value from the Monte Carlo simulations was used to calculate the maximum bound on leakage rather than the mean and standard deviation of the leakage profile. The results obtained using this approach are given by Table XIV and the comparison between the pessimistic approach and the algorithm in [27] is given by Table XV.

C. Maximum Leakage Vector in the Presence of Variations

The maximum input vector obtained using the approach in [27] was used to calculate the leakage of the benchmark circuits in the presence of parameter variations. This leakage current was compared with the leakage current obtained using the vector computed by our heuristic which considers the effect of variations. The comparison results are given by Table XVI. The average error in the mean leakage current was found to be 7.7% when parameter variations were not considered for the determination of maximum leakage vector.

D. Complexity and Usage

The implemented heuristic is quadratic in complexity. Leakage estimation using exhaustive hspice Monte Carlo simulations to determine the maximum leakage vector and the leakage power associated with it takes a few hours to days depending on the number of random vectors considered and the size of the benchmarks. Our method can estimate the same with less than 1.5% error in a matter of few seconds or a few minutes depending on the size of the benchmark. The runtime savings when compared to Monte Carlo simulations with a sweep value of 500 for 100 random vectors is given

TABLE XIII

COMPARISON OF MEAN OBTAINED USING OUR APPROACH (H) WITH THE LEAKAGE CURRENT OBTAINED USING [27]

Benchmark	Gates	PIs	[27] (μ A)	H μ (μ A)	Increase
C17	6	5	0.57	2.4	4.20X
C432	160	36	19.64	65.8	3.35X
C499	202	41	38.55	123.3	3.19X
C880	383	60	44.43	144.6	3.25X
C1908	880	33	97.60	320.4	3.28X
C2670	1193	233	163.96	530.2	3.23X
C3540	1669	50	215.07	709.7	3.30X
C5315	2307	178	297.79	997.9	3.35X
C6288	2416	32	225.36	653.2	2.90X
C7552	3512	207	431.21	1451.9	3.37X

TABLE XIV

COMPARISON OF MAXIMUM LEAKAGE OBTAINED USING THE PESSIMISTIC APPROACH WITH RANDOM VECTOR TESTING

Benchmark	Heuristic Max(mA)	Random Max(mA)	Error(%)
C17	0.065	0.068	4.4
C432	1.743	1.718	-1.4
C499	4.285	4.287	0.1
C880	4.380	4.562	3.9
C1908	10.114	10.590	4.5
C2670	17.568	18.128	3.1
C3540	22.387	21.152	-5.8
C5315	31.379	31.139	-0.1
C6288	15.915	15.361	-3.6
C7552	46.334	45.898	-0.1

by Table XII. The CPU runtime of the overall approach is given by Table XVI.

HeuristicMax accepts an RTL description of the CMOS circuit. It is assumed that all the cells in the standard cell library have been pre-characterized for leakage mean and standard deviation using Monte Carlo hspice simulations and are available to HeuristicMax in a look up table. The pre-characterization is a one time effort. The value of λ is specified by the user. Using these inputs, HeuristicMax determines the maximum leakage vector and the leakage power associated with it. It can easily be used to accurately estimate the maximum leakage power of large circuits because of its small runtime when compared to the large run-time of Monte Carlo simulations.

E. Determination of Minimum Leakage Power Vector

HeuristicMax can be modified to determine the minimum leakage vector in the presence of process variations. In this modified approach, a probabilistic best input condition (PBIC) is defined. PBIC puts the standard cell into its least leakage state. The penalty for a cell for not settling into its PBIC, is defined as probabilistic cell leakage penalty (PCLP), given by the difference in the average of the high leakage states and the average of the low leakage states. Conflicting and dominated cells are defined in a similar way as HeuristicMax, but with reference to the PBIC. The calculation of costs is also similar to HeuristicMax but involves the usage of PCLP instead of the PWLA. The modified heuristic to determine the minimum leakage vector tries to minimize the cost penalty and hence finds the leakage vector corresponding to the minimum leakage power.

TABLE XV

COMPARISON OF LEAKAGE OBTAINED BY THE PESSIMISTIC APPROACH (P) WITH LEAKAGE OBTAINED BY [27]

Benchmark	Gates	# PIs	[27] (μA)	P (μA)	Increase
C17	6	5	0.569	64.7	113.7X
C432	160	36	19.643	1742.8	88.7X
C499	202	41	38.554	4284.7	111.1X
C880	383	60	44.437	4380.0	98.6X
C1908	880	33	97.607	10114.3	103.6X
C2670	1193	233	163.965	17568.0	107.2X
C3540	1669	50	215.071	22387.1	104.1X
C5315	2307	178	297.791	31378.8	105.4X
C6288	2416	32	225.360	15915.1	70.7X
C7552	3512	207	431.215	46334.1	107.4X

TABLE XVI

COMPARISON OF LEAKAGE MEANS USING THE MAX VECTOR OBTAINED BY OUR APPROACH (H) WITH THE MAX VECTOR OBTAINED BY [27]

Benchmark	H μ (μA)	[28] μ (μA)	(%) Error	Runtime (s)
C17	2.40	2.20	9.1	0.44
C432	65.80	63.75	3.2	8.37
C499	123.33	114.69	7.5	14.29
C880	144.57	133.72	8.1	59.94
C1908	320.40	297.16	7.8	184.54
C2670	530.23	498.42	6.4	2363.53
C3540	709.73	641.03	10.7	983.59
C5315	997.96	923.39	8.1	6022.29
C6288	653.28	601.64	8.6	1522.49
C7552	1451.9	1341.3	8.3	16283.24
Average	-	-	7.77	-

VII. CONCLUSION AND FUTURE WORK

In this work, we have developed a heuristic to accurately estimate the maximum run-time leakage power bound as a function of both the input vectors and variations in process parameters. The implemented method has been effective in accurately estimating the PDF, mean and standard deviation of the total leakage current distribution of ISCAS-85 benchmark circuits and the average errors when compared with exhaustive random vector testing for mean and standard deviation are 1.32% and 1.41% respectively. The algorithm in [27] is found to under-estimate the leakage power by a factor of 3.4X as it does not consider the effect of parameter variations. In this work, process variations were considered to be random. The effects of spatial correlations can be included to increase the accuracy of this approach. Layout level analysis further helps to determine the accuracy of the proposed estimation technique.

REFERENCES

- [1] S. Borkar, T. Karnik, S. Narendra, J. Tschanz, A. Keshavarzi, and V. De, "Parameter variations and impact on circuits and microarchitecture," in *DAC*, 2003, pp. 338–342.
- [2] A. Agarwal, D. Blaauw, V. Zolotov, S. Sundareswaran, M. Zhao, K. Gala, and R. Panda, "Statistical delay computation considering spatial correlations," in *ASP-DAC*, 2003, pp. 271–276.
- [3] R. Rao, A. Srivastava, D. Blaauw, and D. Sylvester, "Statistical analysis of subthreshold leakage current for vlsi circuits," *IEEE Trans. on VLSI Systems*, vol. 12, pp. 131–139, Feb 2004.
- [4] D. Blaauw, K. Chopra, A. Srivastava, and L. Scheffer, "Statistical timing analysis: From basic principles to state of the art," *IEEE Trans. on CAD of Integrated Circuits and Systems*, pp. 589–607, 2008.
- [5] H. Chang and S. Sapatnekar, "Statistical timing analysis considering spatial correlations using a single pert-like traversal," in *ICCAD*, 2003, pp. 621–625.
- [6] A. Agarwal, D. Blaauw, V. Zolotov, and S. Vrudhula, "Statistical timing analysis using bounds and selective enumeration," in *Proc. TAU*, 2002, pp. 29–36.
- [7] A. Devgan and C. Kashyap, "Block-based static timing analysis with uncertainty," in *ICCAD*, 2003, pp. 607–614.
- [8] M. Orshansky and A. Bandyopadhyay, "Fast statistical timing analysis handling arbitrary delay correlations," in *Proc. ACM/IEEE DAC*, 2004, pp. 337–342.
- [9] A. Srivastava, S. S. Shah, K. Agarwal, D. Sylvester, D. Blaauw, and S. Director, "Accurate and efficient parametric yield estimation considering correlated variations in leakage power and performance," in *Proc. ACM/IEEE DAC*, 2005, pp. 535–540.
- [10] H. Chang and S. Sapatnekar, "Full-chip analysis of leakage power under process variations, including spatial correlations," in *DAC*, 2005, pp. 523–528.
- [11] S. Bhardwaj, Y. Cao, and S. Vrudhula, "Statistical leakage minimization through joint selection of gate sizes, gate lengths and threshold voltage," in *IEEE ASPDAC*, 2006, pp. 953–958.
- [12] S. Bhardwaj, S. Vrudhula, and Y. Cao, "Lotus: Leakage optimization under timing uncertainty for standard-cell designs," in *IEEE Proc. Int. Symp. Quality Electron. Design*, 2006.
- [13] K. Roy, S. Mukhopadhyay, and H. MahmoodiMeimand, "Leakage current mechanisms and leakage reduction techniques in deep-submicrometer cmos circuits," *Proceeding of IEEE*, vol. 91, pp. 305–327, 2003.
- [14] S. Mukhopadhyay, A. Raychowdhury, and K. Roy, "Accurate estimation of total leakage current in scaled cmos logic circuits based on compact current modeling," in *DAC*, 2003, pp. 169–174.
- [15] (2007) Semiconductor industry association, international technology roadmap for semiconductors. [Online]. Available: <http://www.itrs.net/>
- [16] A. Chandrakasan, W. Bowhill, and F. Fox, *Design of high-performance micro-processor circuits*. IEEE press, 2001.
- [17] S. Bobba and I. Hajj, "Maximum leakage power estimation for cmos circuits," in *Proceedings of the IEEE Alessandro Volta Memorial Workshop on Low Power Design*, 1999, p. 116.
- [18] S. Borkar, "Design challenges of technology scaling," *IEEE Micro*, vol. 19, pp. 23–29, jul 1999.
- [19] D. Lee, D. Blaauw, and D. Sylvester, "Runtime leakage minimization through probability-aware optimization," *IEEE Trans. on Very Large Scale Integration Systems*, vol. 14, pp. 1075–1088, 2006.
- [20] Y.-S. Lin and D. Sylvester, "Runtime leakage power estimation technique for combinational circuits," in *ASPDAC*, 2007, pp. 660–665.
- [21] R. S. Guindi and F. Najm, "Design techniques for gate-leakage reduction in cmos circuits," in *IEEE International Symposium on Quality Electronic Design (ISQED)*, 2003, pp. 61–65.
- [22] Arizona state university, predictive technology model. [Online]. Available: <http://www.eas.asu.edu/ptm/>
- [23] S. Zhang, V. Wason, and K. Banerjee, "A probabilistic framework to estimate full-chip threshold leakage power distribution considering within-die and die-to-die p-t-v variations," in *Proc. of ISLPED*, 2004.
- [24] Bsim3 home page. [Online]. Available: <http://www-device.eecs.berkeley.edu/bsim3/>
- [25] M. C. Johnson, D. Somasekhar, and K. Roy, "Models and algorithms for bounds on leakage in cmos circuits," in *IEEE Trans. on CAD*, 1999, pp. 714–725.
- [26] A. Abdollahi, F. Fallah, and M. Pedram, "Leakage current reduction in cmos vlsi circuits by input vector control," in *IEEE TVLSI*, 2004, pp. 140–154.
- [27] R. M. Rao, F. Liu, J. L. Burns, and R. B. Brown, "A heuristic to determine low leakage sleep state vectors for cmos combinational circuits," in *ICCAD*, 2003, pp. 689–692.
- [28] L. F. Fenton, "The sum of lognormal probability distributions in scatter transmission systems," *IRE Trans. Commun. Syst.*, vol. CS-8, pp. 57–67, 1960.