Probabilistic Dual-Vth Leakage Optimization Under Variability

Azadeh Davoodi
Azade@eng.umd.edu

Ankur Srivastava
Ankurs@eng.umd.edu

Department of Electrical and Computer Engineering
University of Maryland, College Park
MD, USA, 20742

ABSTRACT

In this paper we address the problem of growing leakage variability through effective dual-threshold voltage assignment. We propose a probabilistic dynamic programming-based method to assign dual-threshold voltages such that the overall expected leakage is minimized under a given probability of violating the timing constraint (timing yield). The key characteristics of our strategy are two pruning criteria that stochastically identify pareto-optimal solutions and prune the sub-optimal ones. Compared to other variability-driven dual-threshold voltage assignment schemes, the main advantages of our approach are 1) considering correlations due to common sources of variation, 2) providing controllable runtime, which in one of the proposed strategies is comparable to the deterministic algorithm, and 3) performing optimization based on all the signal paths simultaneously, as opposed to one path at a time. Experimental results indicate that the proposed probabilistic scheme is significantly better than a comparable deterministic dual-threshold voltage assignment, both in terms of expected leakage and the probability of violating the timing constraint.

Categories and Subject Descriptors

B.6.2 [**LOGIC DESIGN**]: Design Aids—*Automatic synthesis, optimizati on*

General Terms

Algorithms, Performance, Design, Theory

Keywords

Leakage, Process Variations

1. INTRODUCTION

Continuous shrinking of device feature sizes has enabled scaling of the voltage, resulting in massive reduction in dynamic power. The reduction in supply voltage must be accompanied with a reduction in threshold voltage in order to address adverse effects on delay. This threshold reduction results in an exponential increase in the leakage power.

The current state of research has proposed many strategies for reducing leakage. Having multiple thresholds on chip is one of the most important among them.

The approach in [8], [12] (and many others) suggest assigning the gates on critical paths to low threshold and assigning the ones in non-critical paths to high-threshold (dual-V_{th} technology).

A major issue in leakage optimization is due to the variations caused by the manufacturing process. Fabrication variability occurs in different device parameters such as the effective channel length, oxide thickness, and doping profile. Although the scope of variation on these parameters might be small, however a large variation will be caused in leakage current due to its exponential dependence on parameters such as the the effective channel length. It has been shown that a 12.5% variation in the effective channel length can cause a 30% leakage variation in a pFET and 400% leakage variation for a netlist of a few thousand gates [11]. Such a degree of variation has to be considered while doing leakage optimization. Variability makes leakage and delay of the circuit behave as random variables.

In this paper we investigate the variability-driven leakage optimization under a timing constraint using the dual-V_{th} technology. This problem has been addressed before: The approach in [2] proposes a sensitivity-based optimization framework, in which the gates that are highly sensitive are identified to decide the appropriate threshold assignment. This is an iterative approach and could have a high execution time before convergence. The technique in [6], approaches the leakage optimization problem by addressing individual paths. However it is based on the simplifying assumption that all the paths are non-overlapping. The approach in [1] also optimizes leakage probabilistically but ignores correlations in leakage and delay of gates due to common variation sources.

We also model leakage and delay of a gate as random variables. We present a *probabilistic* dynamic programming-based strategy for assigning the threshold voltages to the gates such that the expected value of the leakage is minimized under a given timing constraint violation probability. At each node a set of pareto-optimal solutions are identified while considering process variations and the rest are pruned out. This is based on two proposed pruning criteria. The total number of pareto-optimal solutions stored at each node can be controlled. This is used to generate a tradeoff between runtime and quality of solution. Moreover, our approach is applied to a Directed Acyclic Graph that might contain over-lapping paths, while capturing spatial correlations in the delay and leakage random variables.

Experimental results show that our approach results in significantly better expected leakage and probability of meeting the timing constraint than the traditional deterministic approach.

The organization of the paper is as follows. Section 2 describes the deterministic approach. Section 3 describes the probabilistic problem and the leakage and delay models that were used under process variations. Section 4 has the probabilistic algorithm details including two proposed pruning criteria. Section 5 briefly discusses issues with the Directed Acyclic Graphs. The experimental results are in section 6.

ISLPED'05, August 8–10, 2005, San Diego, California, USA

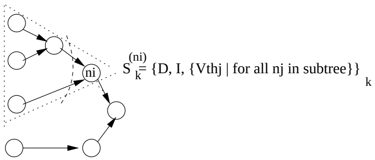

Figure 1: Solution at a node

2. TRADITIONAL APPROACH

2.1 Problem Definition

Given a gate-level netlist, the dual V_{th} leakage optimization technique decides the threshold voltage of each gate, out of two possible V_{th} choices. This allows to minimize leakage under a given timing constraint.

For each gate, the decided threshold voltage, determines its subthreshold leakage current and its delay. The subthreshold leakage current denoted by I_l, is expressed as a function of V_{th} by the following equation:

$$I_l = I_0 e^{\frac{V_{gs}-V_{th}}{nV_T}} \tag{1}$$

Here $I_0 = \mu_0 C_{ox}(W/L)V_T^2 e^{1.8}$, where C_{ox} is the gate oxide capacitance, (W/L) is the width to length ratio of the leaking MOS device, μ_0 is the zero bias mobility. In equation 1, V_{gs} is the gate to source voltage, V_T is the thermal voltage and n is the sub-threshold swing coefficient.

The delay of a gate denoted by D, is expressed in terms of its V_{th} by the following equation:

$$D \propto \frac{C_L V_{dd}}{(V_{dd} - V_{th})^{\alpha}} \tag{2}$$

Here C_L is the load capacitance at the gate output and α is the velocity saturation index which is about 1.3 for the 0.18 μm CMOS technology.

2.2 Deterministic Approach

The given gate-level netlist is described as a Directed Acyclic Graph (DAG) where each gate is represented as a node and the nets are represented as directed edges in the graph that connect any source gate to its fanout(s). A virtual sink node is also added that has incoming edges from all the primary outputs.

A popular dynamic programming approach to solve the dual-V_{th} assignment problem, traverses the nodes in topological order from the primary inputs to the primary outputs [12]. Each node n_i contains a set of pareto-optimal solutions as it is encountered in the topological traversal. The j^{th} pareto-optimal solution, denoted by $S_j^{(n_i)}$, contains the V_{th} assignment to all the nodes that are located in the fanin subtree of n_i, including n_i itself. In addition, $S_j^{(n_i)}$ contains the arrival time at the node output denoted by $D_j^{n_i}$, and the estimated leakage of the node subtree which includes n_i, denoted by $I_j^{n_i}$. Figure 1 illustrates this. During the topological traversal, for n_i, the current encountered node, the pareto-optimal solution set is determined in the following 3 steps: 1) Initially the solutions of the children (fanins) of the node are combined to generate a new solution set. 2) This solution set is then combined with the two possibilities of V_{th} choices of n_i, therefore it's size is doubled. 3) The resulting solutions are then compared and the sub-optimal solutions are pruned out.

In step 1, every solution combination of the children of n_i is considered. For any combination, the arrival time is computed as the maximum of the arrival times of the node's children. The leakage is the summation of the leakage of the node's children for that combination. The assigned V_{th}s, for the gates in the subtree of n_i, is the union of the V_{th} assignments of all the children's solutions. This resulting solution set is denoted by $S_{fanin}^{(n_i)}$. Note that the number of solutions in $S_{fanin}^{(n_i)}$ is the multiplication of the number of solutions of the children of n_i.

The resulting solution set is combined with the two possibilities of the V_{th}s of n_i in step 2. For each solution $S_j \in S_{fanin}^{(n_i)}$, a new solution is generated. For this new solution, the arrival time is the summation of the arrival time of S_j and the delay of n_i for that V_{th}, determined by equation 2. The leakage is the summation of the leakage of S_j and n_i for that V_{th}, determined by equation 1. Finally the union of the V_{th}s of S_j and the V_{th} of n_i is the V_{th} assignment for the nodes in the fanin subtree. Number of solutions at this point is twice the size of $S_{fanin}^{(n_i)}$.

The solution set generated in step 2 is then evaluated and sub-optimal solutions are pruned out. The number of stored solutions is set to be proportional to the summation of the solutions of the node's children. This is a necessary step to avoid exponential growth of solution space, and acheive impractical runtime. To determine the stored pareto-optimal solutions, among all solutions that have the same arrival time, the one with minimum leakage is chosen. This a purely heuristic.

In the end, in the virtual primary output node, among all solutions that have an arrival time smaller than the given timing constraint, the one with minimum leakage is selected.

2.3 Deterministic Approach Under Variability

The presented deterministic approach does not account for the increasingly importance of process variations on leakage and delay of the gates. Under such variations, there are two possible ways in which a traditional approach can be modified to consider variability: It can estimate the delay and leakage of each gate with its expected value or its worst case. Such approximations are either too optimistic or pessimistic. It has been shown that estimating all gate delays by their expected values causes an under-estimation (optimistic scenario) of the arrival time since a stochastic MAX of two random variables has a higher expected value than the MAX of their expected values [4]. If the gate delays are represented by their worst case values, this may cause an over-estimation of the circuit delay.

In the next section, a variability-aware approach to this problem is presented which is probabilistic in nature and does not suffer from the shortcomings of the deterministic approach. Such a probabilistic approach assumes the delay and leakage of each gate to be random variables. The optimization is done probabilistically by estimating the arrival time and leakage as random variables for each solution.

3. PROBABILISTIC APPROACH

3.1 Problem Definition

Given a gate-level netlist, a constraint T_{cons} for the arrival time at the primary outputs, two choices of threshold voltages for the gates and P_{viol}, a maximum allowed timing violation probability, decide one threshold voltage for each gate, such that the expected value of the subthreshold leakage current is minimized while the probability of violating T_{cons} is at most P_{viol}.

This definition, assumes that process variation randomizes design parameters which in return varies the characteristics of a gate. Thus the delay and leakage of each gate, and the output arrival time and overall circuit leakage are also random variables. For each solution, the output arrival time violates the timing constraint T_{cons} with a certain probability expressed as: $\int_{T_{cons}}^{\infty} f_T(t)dt$, where f_T is the probability density function of T, the arrival time random variable. Based on the probabilistic definition, only the solutions that have an arrival time, which violates T with a probability of at most P_{viol}, are of interest. Among these solutions, the one with minimum expected value of leakage is the best. In addition to accounting for process variations, such a probabilistic definition provides a risk-management framework for the designer to decide the amount of risk at his/her discretion.

3.2 Modeling the Distribution of Leakage

In this section a probabilistic model from [10] is reviewed that estimates the distribution of the leakage of a MOSFET. The manufacturing process causes variation in many different parameters in the system such as the effective channel length L_{eff} and gate oxide thickness t_{ox}. To model the variation in the subthreshold leakage current, [10] assumes the variation in the system parameters cause variation in the V_{th} of a MOSFET and models this as:

$$V_{th} = V_{th0} - \sum_{\forall i} \beta_{X_i} \frac{X_{i0} - X_i}{X_{i0}} \quad (3)$$

In the above equation V_{th} is a random variable that is described as a linear function of the X_i random variables. Here X_i can express any gate parameter such as L_{eff} or t_{ox} that has variation. V_{th0} and X_{i0} are the expected values of V_{th} and X_i respectively. β_{X_i} is a constant for the MOSFET.

With the following assumption, [10] shows that the subthreshold leakage current of a MOSFET, initially described using equation 1, is written as below to consider variability:

$$I_l = I_{s0} \frac{W_{eff}}{L_{eff}} e^{\frac{V_{gs}-V_{th0}}{nV_T} \beta_{L_{eff}} \frac{L_{eff0}-L_{eff}}{L_{eff0}} \sum \forall X \, \beta_X \frac{X_0 - X}{X_0}} \quad (4)$$

Here the X_i are random variables. To ease the use of this probabilistic model in the dual V_{th} leakage optimization, one can write the Taylor series of equation 4:

$$I_{X_1,X_2,...} = I(X_{10}, X_{20}, ...) + \sum_{\forall i} (X_i - X_{i0}) \frac{dI}{dX_i}|_{X_{10},...} + ... \quad (5)$$

This is a polynomial representation of equation 4 that allows easier calculation of the variance or higher order moments.

3.3 Capturing Correlations

In the dual-V_{th} leakage optimization, while considering the delay and leakage of all the gates together, it is important to take into account the degree of correlation between the X_i variables. Correlations can arise due the to the spatial location of the gates on the chip. The gates that are more close to each other, are more probable to have similar variation in their X_i variables, hence are more correlated. The expression of equation 5 allows better representation of these correlations, by introducing common X_i variables for all the gate leakages.

We are interested to capture spatial correlation due to variation in L_{eff}. The L_{eff} of a device in gate i is modeled as:

$$L_{eff_i} = a_{i0} + \sum_{j=1}^{4} a_{ij} \times L_j \quad (6)$$

Here L_{eff_i} is the effective channel length for gate i and a_{i0} is the expected value of L_{eff_i}. In this equation, the four L_j variables are independent and have standard Gaussian distribution (N∼(0,1)). The a_{ij} coefficients are proportional to the Manhattan distance of gate i from the (left/right top/bottom) corners of the chip, assuming the gates are placed. The a_{ij} coefficients reflect the degree of spatial correlation between different gates, i.e. if two gates are physically close to each other, their corresponding coefficients will be very similar, thus their L_{eff} variables will be highly correlated. The a_{ij} coefficients are scaled such that the variance of L_{eff} is matched with statistical data. Figure 2 illustrates this. With this assumption, the Taylor series of the subthreshold leakage is simplified to:

$$I(L_{eff_i}) = I_{s0} \frac{W_{eff_i}}{a_{i0}} e^{\frac{V_{igs}-V_{ith0}}{nVT}} (1 + \frac{(\beta - 1)}{2a_{i0}} \sum_{j=1}^{4} a_{ij} L_j + ...) \quad (7)$$

The above equation is only specific to L_{eff} variation. Other variations can be captured similarly. Note that since the leakage of each gate is represented as this polynomial expression for variations in L_i, the correlations among the leakage of different gates can be computed and represented more easily. In the dual V_{th} leakage optimization problem, the leakage of each gate is represented in the above forms.

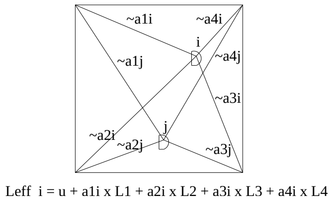

Figure 2: Capturing spatial correlations

For a set of nodes, the overall leakage can then be expressed in the same form by adding the leakage expression of each gate, by summing the corresponding coefficients in all the expressions. Next a variation-aware delay model will similarly be extracted.

3.4 Modeling the Delay Distribution of A Gate

Similar to the leakage model of the previous section, a variation-aware delay model can be extracted from equation 2 for each gate. Different sources of with-in-die variation such as L_{eff}, T_{ox}, etc. cause variation in the parameters of equation 2, hence variation in the delay of a gate. For any of these parameter variations the Taylor series expansion of a gate's delay is:

$$D_{X_1,X_2,...} = D(X_{10}, X_{20}, ...) + \sum_{\forall i} (X_i - X_{i0}) \frac{dD}{dX_i}|_{X_{10},...} + ... \quad (8)$$

Here the X_i variables could be L_{eff}, T_{ox} or any other parameter that is affected by variation. As an example consider writing the V_{th} as a linear function of L_i variables (to represent L_{eff} variability). For details see equations 3 and 6. The Taylor series expansion of the delay model of gate i is simplified to:

$$D_i = \frac{C_L V_{dd}}{(V_{dd} - V_{ith0})^{\alpha}} (1 + \frac{\alpha\beta}{a_{i0}(V_{dd} - V_{ith0})} \sum_{j=1}^{4} a_{ij} L_j + ...) \quad (9)$$

Note we consider the first-order Taylor series expansion of D_i and ignore higher order terms. This linear form captures correlations between different gate delays very well. The presented variation-aware models for leakage and delay of a gate will be incorporated in the next section in an algorithm that targets the probabilistic definition of the problem.

4. PROBABILISTIC ALGORITHM

In this section a probabilistic algorithm is presented that targets the probabilistic definition of the dual V_{th} leakage optimization problem. It's main difference with the presented deterministic approach is that the delay and leakage of the gates thus the arrival times are random variables. These random variables are correlated to each other, since they have common sources of variation.

4.1 Global Algorithm

The probabilistic algorithm is also a dynamic programming based approach where the nodes are traversed topologically from the primary inputs to the primary outputs. Each node has a set of pareto-optimal solutions where each solution S_i contains the following three quantities: the signal arrival time at the node output (denoted by T_i), the V_{th} assignment and the expected leakage of the fanin subtree of the node (denoted by $E[I_i]$).

During the topological traversal, at a node n_i, the following three steps are done: 1) Initially the solutions of the fanins of the node are combined, to generate a new solution set ($S^{(n_i)}_{fanin}$) which contains the random variable for the arrival time at the node input and the expected leakage of the fanin subtree of the node. 2) This solution set is combined with the two V_{th} possibilities of the node to get a new solution set at the output of the node. 3) Finally probabilistic pruning is done to store a limited pareto-optimal set of solutions. In step 1, for every combination of the solutions of the fanins of n_i, a specific V_{th} assignment is determined for all the nodes in the fanin-subtree of n_i.

This V_{th} assignment results in an expected leakage for each gate, which can be obtained from the probabilistic leakage model that was presented in the previous section. The expected leakage for this solution combination is computed by adding the expected leakage of each gate in the fanin-subtree of n_i. For this solution combination, the corresponding arrival time is computed by doing a probabilistic max operation as explained next.

Assume for a solution combination in a node with two fanins, the arrival times of the fanins, denoted by T_i and T_j, are represented as linear expressions as below:

$$T_i = c_{i0} + \sum_l c_{il} X_l \qquad T_j = c_{j0} + \sum_l c_{jl} X_l \tag{10}$$

In the above equations X_ls are the random variables representing the design parameters that are affected by variation. The max of these two arrival times is approximated back into the same linear form as in [4] and [5]:
$Max(T_i, T_j) \simeq c_{k0} + C \sum_{l=1}^{n} c_{kl} X_l$. The average ($c_{k0}$) and coefficients ($c_{kl}$) are computed using [3]:

$$c_{k0} = c_{i0}\phi(\alpha) + c_{j0}\phi(-\alpha) + \theta\varphi(\alpha) \tag{11}$$

$$c_{kl} = c_{il}\phi(\alpha) + c_{jl}\phi(-\alpha) \tag{12}$$

$$\theta^2 = c_{i0}^2 + c_{j0}^2 - 2c_{i0}c_{j0}\rho \quad \alpha = (c_{i0} - c_{j0})/\theta \tag{13}$$

where $\varphi(\alpha)$ and $\phi(\alpha)$ are the probability density function (pdf) and cumulative distribution function (cdf) respectively for a standard normal random variable. In equation 13, ρ is the correlation coefficient between T_i and T_j. The constant $C = \sigma_{act}/\sigma_{appr}$ is defined such that the variance of the actual distribution and the approximate distribution match. The max operation is done on two arrival times at a time. For a node with more than two fanins, this is done iteratively to get the final max result. This procedure is done for every solution combination of the fanins of the nodes to complete step 1.

In step 2, the solution set at the input of n_i is combined with the two V_{th} possibilities of n_i. For each possibility, the expected value of leakage is computed by adding the expected value of the the leakage of n_i with the expected leakage of the solution at the input of n_i. Also for each solution combination, the delay of n_i is expressed as in equation 9. Equation 9 only considers variability in L_{eff}, however as discussed in section 3.4 the linear delay models can be obtained similarly for other parameter variation. For a solution combination assume the delay of n_i, denoted by D_i and the arrival time at the input of n_i, denoted by T_j are written as in equation 10. The resulting arrival time, T_k is represented in a linear form as:
$T_k = (c_{i0} + c_{j0}) + \sum_{l=1}^{n}(c_{il} + c_{jl})X_l$.

By the end of step 2, all solution combinations due to the fanins of n_i and the V_{th} of n_i are generated. At this stage pruning is done to store a limited number of pareto-optimal solutions. This is necessary to avoid exponential growth of the number of solutions and achieve practical run-times. For each node, the number of stored solutions is proportional to the summation of the number of solutions of its fanins. This desired number of solutions is denoted by K through out the paper. By changing K we can obtain a trade-off between the quality of solution and run-time. In the next section, we will explain our pruning criteria when the arrival time and leakage of each solution is a random variable.

Once the topological traversal is finished, at the virtual primary output sink, among all the solutions that are meeting the timing constraint T with a probability that is smaller than P_{viol}, the one with minimum expected leakage is chosen as the best solution which in return determines the V_{th} assignment to all the gates.

4.2 Probabilistic Pruning Criteria

In this section two different probabilistic pruning criteria are proposed. The objective of pruning is to remove the sub-optimal solutions from the solution space.

Pruning Criterion 1:

Each solution S_i at a node is characterized by two random variables: the arrival time at the node output denoted by T_i and the overall leakage of the node fanin subtree denoted by I_i. In the first pruning criterion, the goal is to identify and remove suboptimal solutions and therefore be left with the pareto-optimal ones. Consider two solutions S_i and S_j. Here S_j is an inferior solution if both its leakage and arrival time are inferior to S_i. This is formulated as:

$$Prob(I_i \leq I_j \& T_i \leq T_j) = 1 \tag{14}$$

The above equation says that with probability of 1, S_i will have a smaller arrival time and smaller leakage, thus it prunes S_j. This pruning criterion is based on computing the "pruning" probability in the above equation to identify inferior solutions. Initially this probability is computed between all pairs of solutions. In order to compute this probability, for each solution S_i, the arrival time and leakage are represented as first-order polynomials as explained in section 3.2. The overall leakage (I_i) for the fanin subtree of the node is obtained by summing the linear leakage expressions for each gate (expressed in equation 5). Two solutions $S_i = (I_i, T_i)$ and $S_j = (I_j, T_j)$ are:

$$I_i = b_{i0} + \sum_k b_{ik} X_k \qquad T_i = c_{i0} + \sum_k c_{ik} X_k$$

$$I_j = b_{j0} + \sum_k b_{jk} X_k \qquad T_j = c_{j0} + \sum_k c_{jk} X_k \tag{15}$$

Each solution S_i is pruned out, as soon as one solution S_j is found that prunes S_i with a probability close to 1. Since by the end of this procedure it is not guaranteed that K pareto-optimal solutions are left, next we compare the pair-wise probabilities among the remaining solutions in another round, however this time we decrease the pruning probability from 0.99 to a smaller value (such as 0.60). As soon as the number of remaining solutions becomes K the procedure is stopped. Please note that obtaining the pair-wise probabilities among solutions is done only once and in different rounds only these already-computed probabilities are compared to prune sub-optimal solutions. In theory, the following procedure can be repeated many rounds, each time the limit of the pruning probability is decreased to a smaller value than the previous round until the desired number of solutions are obtained. However due to run-time concerns, it is necessary to have a bound on the number of rounds that the solutions are compared. In our experiments we found two rounds of comparison with probabilities of 0.99 and 0.6 to be suitable.

The following pruning criterion can identify suboptimal solutions with very high accuracy, since it takes the arrival time and leakage of each solution as random variables and does not solely look at the variance or expected value of each quantity. However the main challenge here is to compute the probability in equation 14 which is explained next. For the two solutions S_i and S_j, the pruning probability of equation 14 is: $Prob((I_i - I_j) \leq 0$ & $(T_i - T_j) \leq 0)$, where:

$$I = I_i - I_j = (b_{i0} - b_{j0}) + \sum_k (b_{ik} - b_{jk}) X_k$$

$$T = T_i - T_j = (c_{i0} - c_{j0}) + \sum_k (c_{ik} - c_{jk}) X_k \tag{16}$$

The variables I and T are written as linear combination of X_k variables which are assumed be standard normally-distributed (N∼(0,1)), therefore T and I will have a bivariate normal density function which directly falls from the definition of multivariate normal distribution. The probability of equation 14 is then simplified to:

$$P = Prob(I \leq 0 \& T \leq 0) = \int_{-\infty}^{0}\int_{-\infty}^{0} f_{I,T}(i,t)\,di\,dt \tag{17}$$

where $f_{I,T}$ is the joint density function of the bivariate normal distribution between I and T which is determined by computing the expected value and variance of these two variables and also by computing ρ, their correlation coefficient.

For computing the bivariate normal probability integral, [9] has reviewed many approximation methods. Here we will be using the standard procedure using tetrachroric series:

$$P = \phi_I(a)\phi_T(b)+\varphi_I(a)\varphi_T(b)\sum_{k=0}^{\infty}\frac{1}{(k+1)!}H_{e_k}(a)H_{e_k}(b)\rho^{k+1} \tag{18}$$

where φ_I, φ_T are the pdf and ϕ_I and ϕ_T are cdf of the normally distributed I and T random variables, and $a = -\frac{\mu_I}{\sigma_I}$, $b = -\frac{\mu_T}{\sigma_T}$. The expression $H_{e_k}(x)$ is the Hermite polynomials given by:

$$H_{e_k}(x) = \sum_{m=0}^{[\frac{k}{2}]}\frac{k!}{m!(k-2m)!}(-1)^m 2^{-m} x^{k-2m} \tag{19}$$

It has been shown that the above polynomial, estimates the probability P with an accuracy of 1% if expanded until the 5^{th} order [7]. Therefore, the computation of the double integral is hugely simplified to computation of a 5^{th} order polynomial.

This ease of computation of the pruning probabilities among solutions makes our first pruning criterion to be very exact and yet within a limited run-time. Please note that the assumption of the linear form for the leakage and arrival time for a solution which results in their distributions to be normal is the key reason that allows ease of computation of the pruning probability. Next we will present a very fast alternative pruning criterion.

Pruning Criterion 2:

In this pruning criterion, each solution is characterized by three fields. For a solution S_i, the first two fields are the expected value and the variance of the arrival time which are denoted by $E[T_i]$ and $V[T_i]$. The third field is the expected value of the leakage of the fanin subtree of the node which is denoted by $E[I_i]$. In this pruning criterion all the solutions are compared based on these three charactering fields.

Initially, all the generated solutions are considered in a 3-dimensional space as illustrated in figure 3. This 3-dimensional space has $E[T_i]$, $V[T_i]$ and $E[I_i]$ as its axes. The pareto optimal solutions are then extracted from this set. Two solutions S_i and S_j are pareto-optimal if any of the following conditions hold:

$((V[T_i] > V[T_j])$ and $(E[I_i] < E[I_j]))$ or

$((E[I_i] > E[I_j])$ and $(E[T_i] < E[T_j]))$ or

$((E[T_i] > E[T_j])$ and $(V[T_i] < V[T_j]))$

Figure 3(a) illustrates one such pareto-optimal curve.

Several techniques in computational geometry can be used to generate this pareto-optimal set. The major concern with such techniques is that the total number of identified pareto-optimal solutions could be very large and therefore they would be impractical in our context. As indicated earlier, at each node in the topological traversal, we select K pareto-optimal solutions. These selected solutions should best represent the solution space and the actual pareto-optimal curve at that node. By increasing K we can improve the quality of the algorithm at the cost of runtime. In our approach we fixed K to be proportional to the summation of the solutions of the node's children.

Inorder to identify the K pareto-optimal solutions, all the solutions are considered in the 3-dimensional space. The two axes corresponding to the expected arrival time and variance of the arrival time are each divided into $\sqrt{K}$ uniform units. These define K regions in the plane that is specified by these two axes as illustrated in figure 3(b).

Next these K regions are traversed and among all solutions that fall in the same region, the one with minimum expected value of leakage is chosen. The number of selected solutions will be at most K. Pareto-optimal solutions are then selected from these chosen solutions according to the criterion defined above. This is illustrated in figure 3(b). The above procedure guarantees uniform sampling of the desired 3-dimensional tradeoff-curve.

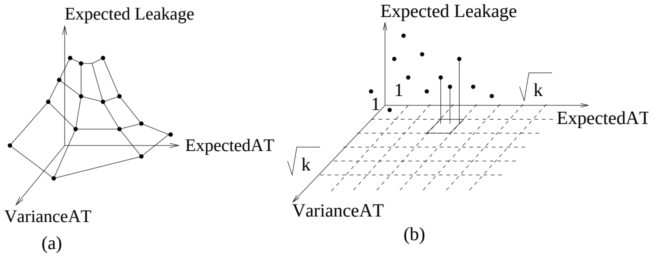

Figure 3: Solution space in the pruning criterion 2

The advantage of this pruning criterion is that it is very fast in nature. For each solution, the required three fields can be obtained very fast. The run-time in this approach is well comparable to the traditional deterministic approach. In the next section we will present experimental results to compare the different presented techniques and pruning criteria.

5. HANDLING OF DAGS

Directed Acyclic Graphs (DAGs) contain nodes with multiple fanouts whose paths converge later on in the graph. The nodes where such paths converge are refereed as reconvergent nodes. In DAGs, there will be shared subtrees among the fanins of the reconvergent nodes. These shared subtrees cause the following problem in the dynamic programming approach: While merging a combination of solutions of the fanins at a reconvergent node, different V_{th}s might be assigned to the same node(s) in the shared subtree(s). Different solutions have been proposed to handle DAG issues. In one approach, the graph is broken into trees and the problem is solved separately for each tree. The solutions of these trees are then combined.

In another approach, every time a conflict happens in V_{th} assignment of a gate, only one V_{th} for the gate is chosen for the conflicted solutions. This could either be the higher or smaller V_{th}. Every time such an adjustment is done the incremental computation of the arrival times and leakage will be incorrect, since the estimated arrival times of the node fanins and their leakage is based on an un-modified V_{th} assignment in their subtrees. Therefore, every time such a re-assignment is done while merging solutions, the delay and leakage has to be computed again by doing a timing analysis for the node's subtree and adding the leakage of every gate in the node subtree. This is implemented in our implemented approaches.

6. EXPERIMENTAL RESULTS

In order to estimate the variation in delay and leakage, we assume the presence of variability in L_{eff} of a device, where L_{eff} for each device is described as in equation 6 to capture spatial correlation for a placed netlist. In order to capture the spatial correlation among the gate parameters, we placed the experimental benchmarks using CAPO placement tool and evaluated the coefficients of the parameters based on the technique presented in section 3.3. We assumed a variance of 10% in the L_{eff} for all the gates.

In the deterministic method the delay and leakage of each gate is estimated with its expected value. This method is denoted by DetExp. Two probabilistic approaches were also implemented that only differed in their pruning criterion which are denoted by Prob1 and Prob2.

1. DetExp: Deterministic using expected value estimates

2. Prob1: Probabilistic using pruning criterion # 1

3. Prob2: Probabilistic using pruning criterion # 2

These methods were tested on a set of MCNC benchmarks, where for a given timing constraint, in the deterministic case the solution that meets the timing constraint with minimum leakage (based on the deterministic estimate i.e. expected value) was chosen.

	T	P_{vio}	DetExp.		Prob#1		Prob#2	
			$E[I]$	$P_v(T)$	$E[I]$	$P_v(T)$	$E[I]$	$P_v(T)$
C432	33.0	0.20	890	0.14	890	0.14	890	0.14
C499	17.5	0.20	1779	0.15	1176	0.19	1277	0.12
C880	32.0	0.20	1479	0.13	1077	0.19	1117	0.17
C1355	17.0	0.20	2030	0.19	1277	0.19	1327	0.16
C1908	29.0	0.20	1388	0.17	1186	0.18	1186	0.18
C3540	42.0	0.20	2991	0.17	2791	0.19	2791	0.19
C5315	31.0	0.20	4857	0.16	4211	0.16	4261	0.16
alu2	8.0	0.20	977	0.03	982	0.05	931	0.04
alu4	12.0	0.20	2230	0.07	1752	0.07	1752	0.07
too-large	12.0	0.20	1063	0.20	862	0.20	912	0.20

Table 1: Probability of meeting required time constraint at root

	DetExp	Prob#1	Prob#2
C432	49	331	50
C499	212	1267	257
C880	41	571	52
C1355	212	1401	256
C1908	181	1399	207
C3540	653	2600	708
C5315	1488	4827	1707
alu2	72	1018	94
alu4	279	1808	331
too-large	20	469	21

Table 2: Run-time (in seconds) of different techniques

In the probabilistic methods, among all the solutions that violate the timing constraint with a probability of at most P_{viol}, the one with minimum leakage is chosen. A solution generated by the deterministic method defines a V_{th} assignment for all the gates. For this assignment, we did a statistical timing analysis to obtain the distribution of timing. We then determined the probability of violating the timing constraint for the deterministic solution.

Table I compares the final solutions generated by these methods. Here column 2 is the timing constraint (in nsecs). For each method, we have reported the expected leakage ($E[I]$) and probability of violating the timing constraint ($P_v(T)$). It can be seen that both of the probabilistic methods consistently resulted in a smaller leakage when compared to the deterministic method. This is since the leakage and arrival time for each solution are assumed to be random variables in presence of variability. In addition the probabilsitic methods use effective pruning criteria to identify good solutions at each stage of optimization. When comparing the two probabilistic methods with each other, in most of the cases, Prob1 generated a solution with slightly smaller leakage.

For each method we also looked at the set of pareto-optimal solutions in the virtual primary output. This is the set from which the final solution is chosen from. These pareto-optimal curves are plotted in figure 4 for the C432 and C880 benchmarks. In these curves, the x-axis is the probability of violating the timing constraint while the y-axis is the expected leakage. From these curves it can be seen that for the same probability of timing violation, the deterministic approach always generates a solution with higher expected leakage. On the other hand, for a fixed expected leakage, the probabilistic approaches always generated a solution with smaller probability of violating the timing constraint.

Finally Table II compares the run-time of the different methods. It can be seen that the run-time of the second probabilistic method (Prob2) is well comparable to the deterministic method, while the run-time of Prob1 always stands at a higher rate than the other two methods. While Prob1 and Prob2 generate similar trade-off curves, Prob2 is superior since pruning is done much faster when compared to Prob1.

We also implemented a worst-case deterministic estimate in which the delay and leakage of each gate was estimated with its worst case value (average + 3σ). However such a deterministic technique was never able to generate a feasible solution for our given timing constraints, since it was over-pessimistic.

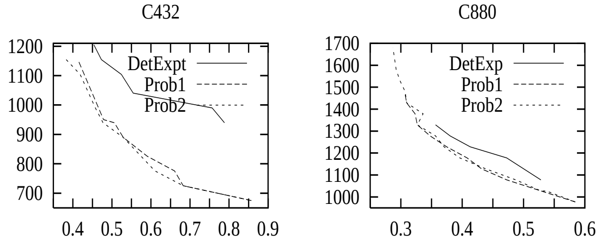

Figure 4: Tradeoff curves of different techniques for two benchmarks (T=30 nsec) x-axis: Timing violation probability, y-axis: Expected leakage (pA)

7. CONCLUSION

In this paper we propose a probabilistic dynamic programming-based approach to the leakage optimization using dual-threshold technology. Our technique effectively considers correlations, probabilistically indentifies pareto-optimal solutions, and results in better expected leakage and probability of satisfying the timing constraint. Experimental results shows the superiority of our approach over determinisitic dual-threshold schemes.

8. REFERENCES

[1] A. Davoodi, V. Khandelwal, A. Srivastava. Variability inspired implementation selection problem. In *ICCAD*, 2004.

[2] A. Srivastava, D. Sylvester, D. Blaauw. Statistical optimization of leakage power considering process variations using dual-vth and sizing. In *DAC*, pages 773–776, 2004.

[3] C. E. Clark. The greatest of a finite set of random variables. In *Operations Research, vol. 9*, pages 85–91, 1961.

[4] C. Visweswariah, K. Ravindran, K. Kalafala, S. G. Walker, S. Narayan. First-order incremental block-based statistical timing analysis. In *DAC*, pages 331–336, June 2004.

[5] H. Chang, S. S. Sapatnekar. Statistical timing analysis considering spatial correlations using a single pert-like traversal. In *ICCAD*, pages 621–628, 2003.

[6] M. Liu, W-S. Wang and M. Orshansky. Leakage power reduction by dual-vth design under probabilitic analysis of Vth variation. In *ISLPED*, Aug. 2004.

[7] O. A. Vasicek. A series expansion for the bivariate normal integral. In *Journal of Computational Finance, Vol.1, No.4*, 1998.

[8] Q. Wang and S.B.K. Vrudhula. Static power optimization of Deep Sub Micron CMOS Circuits for DUAL Vt technology. In *ICCAD*, pages 490–496, Nov 1998.

[9] S. S. Gupta. Probability integrals of multivariate normal and multivariate t. In *Annals of Mathematical Statistics, Vol. 34*, pages 792–828, 1963.

[10] S. Zhang, V. Wason, K. Banerjee. A probabilistic framework to estimate full-chip subthreshold leakage power distribution considering within-die and die-to-die variations. In *ISLPED*, 2004.

[11] S. Zhang, V. Wason, K. Banerjee. Statistical analysis of subthreshold leakage current for VLSI circuits. In *TVLSI, Vol. 2*, pages 131–139, February 2004.

[12] V. Sundararajan, K. Parhi. Low power synthesis of dual threshold voltage CMOS VLSI circuits. In *ISLPED*, pages 139–144, 1999.