Buffer Insertion Considering Process Variation[*]

Jinjun Xiong Kingho Tam Lei He
EE Department, University of California at Los Angeles
Los Angeles, CA, 90095

ABSTRACT

A comprehensive probabilistic methodology is proposed to solve the buffer insertion problem with the consideration of process variations. In contrast to a recent work, we point out, for the first time, that the correlation between the required arrival time and the downstream loading capacitance must be considered in order to solve the problem "correctly". We develop an efficient bottom-up recursive algorithm to calculate the joint probability density function that accurately captures the above correlation, and propose effective pruning rules to exclude probabilistically inferior solutions. We verify our buffer insertion using timing analysis with both device and interconnect variations, and show that compared to the conventional buffer insertion algorithm using nominal device and interconnect parameters, our new buffer insertion methodology can reduce the probability of timing violation by up to 30%.

1. INTRODUCTION

Ultra deep submicron process exhibits significant variations [1] [2]. One of the causes is due to critical dimension being scaled quicker than the development of its controlling process technology [3]. The variability of physical dimensions such as the effective channel length of a transistor is therefore proportionately increasing. As a result of these effects, the performance of manufactured circuits differs significantly from what a circuit simulator predicates under nominal circuit parameters.

Studies on process variations have been mainly focused on variability modeling and statistical timing analysis (STA). For example, [4] proposed three approaches to compute the probability density functions (p.d.f.) that describe the timing distribution of the circuits. [5] advocates to estimate circuit timing variations by computing bounds. However, there are very few works on design optimization that consider process variation effects. For example, essential to high performance circuit design, buffer insertion has been studied extensively in literature e.g., [6] and [7]. Yet, none of them has considered the effect of process variations. To the best of our knowledge, the only work that entails such a flavor is by [8], where a buffer insertion problem that considers the effect of wire length variation has been formulated. However, as we shall point out in Section 3, an unrealistic assumption is made in [8] in an attempt to solve such a problem.

The major contribution of this paper is two-fold: (1) we formulate a buffer insertion problem with the consideration of process variations for both devices and interconnects; and (2) we propose a comprehensive probabilistic methodology to solve the above problem efficiently. Preliminary experiment results have shown that by using our methodology, the timing constraints can be satisfied with a much higher probability than the conventional design under nominal values [6].

The rest of the paper is organized as follow. Section 2 gives the preliminary and modeling aspects of this work. We formulate the problem for buffer insertion considering process variation (BIPV) in Section 3. Section 4 discusses the details of the algorithm and the methodology for solving the BIPV problem. We present experiment results in Section 5 and conclude the paper with discussion of our future work in Section 6.

2. PRELIMINARY AND MODELING

2.1 Buffer Insertion Preliminary

For simplicity of presentation, we follow the same argument as [6] by assuming that the routing tree is given as a binary routing tree and the *legal* buffer positions (nodes) are directly after the branching points of the tree[1]. For a given buffered routing tree, we associate every legal buffer position t in the tree with two numbers: the *input loading capacitance* (or *downstream loading capacitance*) L_t and the *required arrival time* T_t. Denoting c and r as interconnect's *unit length capacitance* and *sheet resistance*, respectively, we model each interconnect segment in the routing tree with length l_i as a π model, where the resistance is given by $r \times l_i$, and the capacitance is given by $c \times l_i$. For a given technology, we associate each buffer from the library with three numbers: the *input capacitance* C_b, the *output resistance* R_b and the *intrinsic delay* T_b. Under the Elmore delay model, the L_t and T_t can be computed as follows.

If node t is obtained by adding a wire of length l_i at its direct downstream node n, then

$$\begin{aligned} L_t &= L_n + c \cdot l_i & (1) \\ T_t &= T_n - r \cdot l_i \cdot L_n - \frac{1}{2} \cdot r \cdot c \cdot l_i^2. & (2) \end{aligned}$$

If node t is obtained by adding a buffer at its direct down-

[*] This work is partially supported by NSF CAREER award CCR-0093273, a UC MICRO grant sponsored by Fujitsu Laboratories of America, Intel and LSI Logic, and a Faculty Partner Award by IBM. E-mail:(jinjun,ktam,lhe)@ee.ucla.edu.

[1] Note that the methodology to be presented in this work does not depend on these assumptions.

1530-1591/05 $20.00 © 2005 IEEE

stream node n, then

$$\begin{aligned} L_t &= C_b & (3)\\ T_t &= T_n - T_b - R_b \cdot L_n. & (4) \end{aligned}$$

If node t is obtained by merging two nodes m and n, then

$$\begin{aligned} L_t &= L_n + L_m & (5)\\ T_t &= min(T_n, T_m). & (6) \end{aligned}$$

It has been proved in [6] that the buffer insertion problem, without considering process variation, can be solved optimally via dynamic programming. Moreover, by properly defining the *dominance relationship* (or *pruning rule*) between two solutions, i.e., solution (L_1, T_1) dominates solution (L_2, T_2) if condition $L_1 < L_2$ and $T_1 > T_2$ are satisfied, [6] proved that by keeping only dominating solutions at every node, the dynamic programming approach can solve the problem in polynomial time without loosing optimality.

2.2 Device and Interconnect Variations

Device variation and interconnect variation are considered as two typical effects of process variations in this work. Because of the process variations in gate thickness, doping density and channel length, buffers' C_b, R_b, and T_b deviate from their respective nominal values. We model such an effect by describing them as random variables that are characterized by a joint probability density function (j.p.d.f.) $g(C_b, R_b, T_b)$, whose domain is given by Ω_{C_b,R_b,T_b}. Due to variations in masking, the manufactured interconnect width and space are also different from their nominal values. We model such an effect by assuming that interconnect's c and r are random variables and that they can be described by the j.p.d.f. as $h(c, r)$, whose domain is given by $\Omega_{c,r}$. As the mechanisms of inducing variations on devices and interconnect are different, it is reasonable to assume that the device and interconnect variations are mutually independent.

3. BIPV PROBLEM FORMULATION

FORMULATION 1. **Buffer Insertion considering Process Variation (BIPV) Problem:** *Given the topology of a routing tree with parasitic capacitance and resistance, required arrival times and loading capacitances specified at all sinks, determine the placement of buffers in the routing tree such that the probability of the required arrival time at the root meeting the design specification is maximized with the consideration of process variations for both interconnect and devices, and as a secondary objective, the number of buffers used are minimized.*

Even though the conventional buffer insertion problem can be solved optimally via dynamic programming [6], solving BIPV problem "optimally" becomes much complicated. The difficulty lies in that:

- L_t and T_t are no longer fixed values, but two random variables. How do we compare two random solutions to tell that one is better than the other?
- L_t and T_t are interdependent, as both are complicated functions of all downstream random variations. Therefore, j.p.d.f., instead of their respective p.d.f., should be used to correctly characterize L_t and T_t.

Obviously, the conventional deterministic dominance relationship between two solutions is no longer applicable. A straight forward extension is to use the mean value as a metric to compare two random solutions. Despite the simplicity of this metric, however, by using the following small example, we show that this simple parametric scheme do not suffice to provide a good guideline for designers to distinguish different design alternatives.

Suppose we are given four buffer insertion solutions, and for simplicity, we assume that all solutions have the same loading capacitance L with different required arrival time T_t, which is described by a discrete probability mass function. The four solutions are: (A) (L, T_1) with $\mathrm{P}(T_1 = 100) = 1$; (B) (L, T_2) with $\mathrm{P}(T_2 = 200) = 0.5$ and $\mathrm{P}(T_2 = 0) = 0.5$; (C) (L, T_3) with $\mathrm{P}(T_3 = 1000) = 0.1$ and $\mathrm{P}(T_3 = 0) = 0.9$; (D) (L, T_4) with $\mathrm{P}(T_4 = 200) = 0.9$ and $\mathrm{P}(T_4 = -800) = 0.1$.

It is obvious that the above four solutions have very different characteristics and designers definitely want to distinguish them for a given design. However, after some simple calculations, we find that all four solutions result in the same mean value of 100 for the required arrival time T_t. Therefore, if the metric is purely based on the mean values, we cannot distinguish the above four designs at all! It may seem tempting for people to argue that they can extend the mean value metric (first order moment) by including variance (second order moment) or even higher order moments as additional metrics to compare solutions. But it is easy to construct cases such that none of the above simple metrics would work in order to distinguish different solutions with variations[2].

Another common practice that people tend to adopt is to ignore the dependence between L_t and T_t. For example, in [8] where the wire length is the only source of variation[3], it is assumed that there is no correlation between L_t and T_t, so that the pruning rules can be carried out based on simple multiplication of their respective p.d.f.s. However, this assumption does not hold in general. For example, if a downstream node n is connected to an upstream node t through an interconnect of length l, then from (1) and (2), we can clearly see that any increase in l by δ increases L_t by $c\delta$ while decreases T_t by $r\delta + \frac{1}{2}rc(2\delta l + \delta^2)$. Treating T_t and C_t as two independent parameters ignores the fact that increasing l in fact collectively decreases T_t and increases C_t, thus causing the solution to degrade much more quickly than before. Therefore, the j.p.d.f. must be used to correctly describe L_t and T_t's interdependence.

4. BIPV ALGORITHMS

4.1 Algorithm Overview

We follow the same dynamic programming paradigm as [6] to solve the BIPV problem. The overall algorithm (BIPV-ALG) is shown in the top block of Fig. 1. We first traverse the routing tree bottom-up once and build a set of dominant BIPV solutions for all sub-trees. After we reach the root, we pick an optimal solution from the set of kept dominant

[2] Readers are encouraged to verified that the last two solutions in the above small example in fact have the same variance.

[3] We do not model the interconnect length variation in this work as that is not a typical process variation.

solutions, optimizing for the required arrival time and the number of buffers inserted. We then back-track the chosen optimal solution to determine the solution for each sub-tree recursively. The key part to this algorithm is the bottom-up traversal of the routing tree. Therefore, we also present the detailed bottom-up algorithm in the bottom block of Fig. 1.

```
BIPV-ALG($t$)
  Input: root of the routing tree $t$.
  Output: Solution to BIPV problem.
Z = BIPV-BOTTOM-UP-ALG($t$);
$Z^*$ = PICK-BEST-SOL(Z);
BACK-TRACK-SOL($Z^*$);
```

```
BIPV-BOTTOM-UP-ALG($t$)
/* bottom-up traversal of the routing tree */
If node $t$ is sink
 /* Initialization, see section 4.2.4 */
  $Z_1$=($L_t, T_t$)=INIT-ADD-WIRE-JPDF($L_s$,$T_s$);
  $Z_2$=ADD-BUFFER-JPDF($Z_1$);
  return ($Z_1 \cup Z_2$);
Else
  /* Compute solutions from sub-trees */
  $Z_m$ = BIPV-BOTTOM-UP-ALG(t.left);
  $Z_n$ = BIPV-BOTTOM-UP-ALG(t.right);
  /* Merge two solutions */
  $Z = \emptyset$;
  For each solution ($L_m, T_m$) from $Z_m$
    For each solution ($L_n, T_n$) from $Z_n$
      /* see Section 4.2.3 */.
      ($L_t, T_t$)=MERGE-SUBNODE-JPDF(($L_m, T_m$),($L_n, T_n$));
      $Z = Z \cup (L_t, T_t)$;
  /* see Section 4.3 */
  $Z_t$ = PRUNE-JPDF($Z$);
  if node $t$ is root
    return($Z_t$);
  $Z = \emptyset$;
  For each solution ($L_n, T_n$) in $Z_t$
    /* adding a wire, see Section 4.2.1 */
    Compute ($L_1, T_1$) according to (1) and (2);
    $Z_1$ = ADD-WIRE-JPDF($L_n, T_n$);
    $Z = Z \cup Z_1$;
    /* adding a buffer, see Section 4.2.2 */
    $Z_2$ = ADD-BUFFER-JPDF($Z_1$);
    $Z = Z \cup Z_2$;
  return $Z$;
```

Figure 1: BIPV Algorithm.

4.2 Computation of J.P.D.F.

According to (1) and (2), or (3) and (4), we know that the upstream variations do not change the distribution of L_t and T_t, as L_t and T_t are only functions of node t's downstream random variations. Therefore, L_t and T_t's distributions are independent of their upstream random variations. Moreover, if (L_{t1}, T_{t1}) and (L_{t2}, T_{t2}) share some (or do not share any) common down-stream paths, then L_{t1}, T_{t1}, L_{t2}, and T_{t2} are mutually correlated (or independent).

Even thought (L_t, T_t) depend on their downstream random variations in a complicated way, the way we compute (L_t, T_t) is via the recursive equations according to either (1) and (2), or (3) and (4), or (5) and (6). Therefore, we develop an efficient algorithm to compute the j.p.d.f. of (L_t, T_t) in a recursive fashion while we traverse the routing tree. This is particularly useful in the context of dynamic programming, because only incremental computation is necessary at each node. To develop the recursive computation formula, we employ the multivariate transformation technique [9]. We denote the j.p.d.f. of node t's direct downstream nodes as f_{L_n,T_n} with its domain given by Ω_{L_n,T_n}.

4.2.1 J.P.D.F. after Adding a Wire

To compute the j.p.d.f. of (L_t, T_t) obtained from (1) and (2) via the multivariate transformation technique, we introduce two new random variables X and Y as follows:

$$\begin{aligned} X &= c; & (7)\\ Y &= r; & (8)\\ L_t &= L_n + c \cdot l_i; & (9)\\ T_t &= T_n - r \cdot l_i \cdot L_n - \frac{1}{2} \cdot r \cdot c \cdot l_i^2. & (10) \end{aligned}$$

By transforming variables, we obtain

$$\begin{aligned} c &= X; & (11)\\ r &= Y; & (12)\\ L_n &= L_t - X \cdot l_i; & (13)\\ T_n &= T_t + Y \cdot l_i \cdot L_t - \frac{1}{2} \cdot Y \cdot X \cdot l_i^2. & (14) \end{aligned}$$

It is easy to verify that the mapping between (c, r, L_n, T_n) and (X, Y, L_t, T_t) is a one-to-one mapping. Therefore, the Jacobian is given by

$$\begin{aligned} J &= \begin{vmatrix} \frac{\partial c}{\partial X} & \frac{\partial c}{\partial Y} & \frac{\partial c}{\partial L_t} & \frac{\partial c}{\partial T_t} \\ \frac{\partial r}{\partial X} & \frac{\partial r}{\partial Y} & \frac{\partial r}{\partial L_t} & \frac{\partial r}{\partial T_t} \\ \frac{\partial L_n}{\partial X} & \frac{\partial L_n}{\partial Y} & \frac{\partial L_n}{\partial L_t} & \frac{\partial L_n}{\partial T_t} \\ \frac{\partial T_n}{\partial X} & \frac{\partial T_n}{\partial Y} & \frac{\partial T_n}{\partial L_t} & \frac{\partial T_n}{\partial T_t} \end{vmatrix} \\ &= \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ -l_i & 0 & 1 & 0 \\ \frac{1}{2} \cdot Y \cdot l_i^2 & l_i \cdot L_t - \frac{1}{2} \cdot X \cdot l_i^2 & Y \cdot l_i & 1 \end{vmatrix} \\ &= 1. & (15) \end{aligned}$$

As c and r are independent of L_n and T_n, the j.p.d.f. of (c, r, L_n, T_n) is given by: $f_{c,r,L_n,T_n} = h(c, r) \cdot f_{L_n,T_n}$. Therefore, we obtain the j.p.d.f. of (X, Y, L_t, T_t) as follows:

$$\begin{aligned} & f_{X,Y,L_t,T_t} = |J| \cdot f_{c,r,L_n,T_n}(c, r, L_n, T_n) & (16)\\ & = h(c, r) \cdot f_{L_n,T_n}(L_n, T_n) \\ & = h(x, y) \cdot f_{L_n,T_n}(L_t - X \cdot l_i, T_t + Y \cdot l_i \cdot L_t - \frac{X \cdot Y \cdot l_i^2}{2}) \end{aligned}$$

Then the j.p.d.f. of (L_t, T_t) is obtained by

$$f_{L_t,T_t} = \int_{\Omega_{X,Y}(L_t,T_t)} f_{X,Y,L_t,T_t} dX dY, \qquad (17)$$

where $\Omega_{X,Y}(L_t, T_t)$ is the domain for X, Y in terms of L_t and T_t. Knowing c and r's domain $\Omega_{c,r}$ and (L_n, T_n)'s domain $\Omega_{L_n,Tn}$, we can deduce $\Omega_{X,Y}(L_t, T_t)$ according to (7), (8) (9) and (10).

4.2.2 J.P.D.F. after Adding a Buffer

To compute the j.p.d.f. of (L_t, T_t) obtained from (3) and (4) via the multivariate transformation technique, we introduce three new random variables X, Y and Z as follows:

$$\begin{aligned} X &= L_n; & (18)\\ Y &= R_b; & (19)\\ Z &= T_b; & (20)\\ L_t &= C_b; & (21)\\ T_t &= T_n - T_b - R_b \cdot L_n. & (22) \end{aligned}$$

By transforming variables, we obtain

$$\begin{aligned} C_b &= L_t; & (23)\\ R_b &= Y; & (24)\\ T_b &= Z; & (25)\\ L_n &= X; & (26)\\ T_n &= T_t + Z + Y \cdot X. & (27) \end{aligned}$$

It is easy to show that the mapping between $(C_b, R_b, T_b, L_n, T_n)$ and (X, Y, Z, L_t, T_t) is a one-to-one mapping. Therefore, the Jacobian is given by

$$\begin{aligned} J &= \begin{vmatrix} \frac{\partial C_b}{\partial X} & \frac{\partial C_b}{\partial Y} & \frac{\partial C_b}{\partial Z} & \frac{\partial C_b}{\partial L_t} & \frac{\partial C_b}{\partial T_t} \\ \frac{\partial R_b}{\partial X} & \frac{\partial R_b}{\partial Y} & \frac{\partial R_b}{\partial Z} & \frac{\partial R_b}{\partial L_t} & \frac{\partial R_b}{\partial T_t} \\ \frac{\partial T_b}{\partial X} & \frac{\partial T_b}{\partial Y} & \frac{\partial T_b}{\partial Z} & \frac{\partial T_b}{\partial L_t} & \frac{\partial T_b}{\partial T_t} \\ \frac{\partial L_n}{\partial X} & \frac{\partial L_n}{\partial Y} & \frac{\partial L_n}{\partial Z} & \frac{\partial L_n}{\partial L_t} & \frac{\partial L_n}{\partial T_t} \\ \frac{\partial T_n}{\partial X} & \frac{\partial T_n}{\partial Y} & \frac{\partial T_n}{\partial Z} & \frac{\partial T_n}{\partial L_t} & \frac{\partial T_n}{\partial T_t} \end{vmatrix} \\ &= \begin{vmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ Y & X & 1 & 0 & 1 \end{vmatrix} = -1. \qquad (28) \end{aligned}$$

As C_b, R_b, and T_b are independent of L_n and T_n, the j.p.d.f. of $(C_b, R_b, T_b, L_n, T_n)$ is given by: f_{C_b,R_b,T_b,L_n,T_n} $=g(C_b, R_b, T_b) \cdot f_{L_n,T_n}$. Therefore, we obtain the j.p.d.f. of (X, Y, Z, L_t, T_t) as follows:

$$\begin{aligned} f_{X,Y,Z,L_t,T_t} &= |J| \cdot f_{C_b,R_b,T_b,L_n,T_n} \\ &= g(C_b, R_b, T_b) \cdot f_{L_n,T_n} \qquad (29) \\ &= g(L_t, Y, Z) \cdot f_{L_n,T_n}(X, T_t + Z + Y \cdot X). \end{aligned}$$

Then j.p.d.f. of (L_t, T_t) is obtained by

$$f_{L_t,T_t} = \int_{\Omega_{X,Y,Z}(L_t,T_t)} f_{X,Y,Z,L_t,T_t} dX dY dZ, \qquad (30)$$

where $\Omega_{X,Y,Z}(X, Y, Z)$ is the new domain for X, Y and Z in terms of L_t and T_t. Knowing C_b, R_b and T_b's domain Ω_{C_b,R_b,T_b} and (L_n, T_n)'s domain $\Omega_{L_n,Tn}$, we can deduce $\Omega_{X,Y,Z}(X, Y, Z)$ according to (18), (19), (20), (21) and (22).

4.2.3 J.P.D.F. after Merging Two Solutions

To compute the j.p.d.f. for (L_t, T_t) after merging two solutions according to (5) and (6), we introduce two new random variables as follows:

$$\begin{aligned} X &= L_m & (31)\\ Y &= T_n + T_m & (32)\\ L_t &= L_n + L_m & (33)\\ T_t &= min(T_n, T_m) & (34) \end{aligned}$$

Because of the min-function, there is no one-to-one mapping relation between (L_m, T_m, L_n, T_n) and (X, Y, L_t, T_t), thus we cannot use the multivariate transformation technique to compute the j.p.d.f. of (L_t, T_t) directly. But we note that the original domain Ω_{L_m,T_m,L_n,T_n} for (L_m, T_m, L_n, T_n) can be divided into two disjoint sub-domains Ω_1 and Ω_2, where Ω_1 is the sub-domain with $T_m \le T_n$ and Ω_2 is the other sub-domain with $T_m > T_n$. For each disjointed sub-domain, we can show that there exists a one-to-one mapping between (L_m, T_m, L_n, T_n) and (X, Y, L_t, T_t), thus we can compute the j.p.d.f. for each sub-domain by using the multivariate transformation technique. Then by combining the two sub-domains' j.p.d.f., we can compute the j.p.d.f. for the whole domain [9].

We first find the j.p.d.f. for sub-domain Ω_1 with $T_m \le T_n$. We have:

$$T_t = min(T_m, T_n) = T_m \qquad (35)$$

After transformation of variables, we have

$$\begin{aligned} L_m &= X; & (36)\\ L_n &= L_t - X; & (37)\\ T_m &= T_t; & (38)\\ T_n &= Y - T_t. & (39) \end{aligned}$$

The Jacobian is given by

$$\begin{aligned} J &= \begin{vmatrix} \frac{\partial L_m}{\partial X} & \frac{\partial L_m}{\partial Y} & \frac{\partial L_m}{\partial L_t} & \frac{\partial L_m}{\partial T_t} \\ \frac{\partial L_n}{\partial X} & \frac{\partial L_n}{\partial Y} & \frac{\partial L_n}{\partial L_t} & \frac{\partial L_n}{\partial T_t} \\ \frac{\partial T_m}{\partial X} & \frac{\partial T_m}{\partial Y} & \frac{\partial T_m}{\partial L_t} & \frac{\partial T_m}{\partial T_t} \\ \frac{\partial T_n}{\partial X} & \frac{\partial T_n}{\partial Y} & \frac{\partial T_n}{\partial L_t} & \frac{\partial T_n}{\partial T_t} \end{vmatrix} \\ &= \begin{vmatrix} 1 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & -1 \end{vmatrix} = 1. \qquad (40) \end{aligned}$$

As (L_m, T_m) is independent of (L_n, T_n), the j.p.d.f. of (L_m, T_m, L_n, T_n) is given by: f_{L_m,T_m,L_n,T_n} $=f_{L_m,T_m} \cdot f_{L_n,T_n}$, Therefore, we can obtain the j.p.d.f. of (X, Y, L_t, T_t) in sub-domain Ω_1 as follows:

$$\begin{aligned} f_{\Omega_1} &= |J| \cdot f_{L_m,T_m,L_n,T_n} \\ &= f_{L_m,T_m} \cdot f_{L_n,T_n} \\ &= f_{L_m,T_m}(X, T_t) \cdot f_{L_n,T_n}(L_t - X, Y - T_t). \quad (41) \end{aligned}$$

Similarly, for the sub-domain Ω_2 with $T_m > T_n$, We have:

$$\begin{aligned} f_{\Omega_2} &= |J| \cdot f_{L_m,T_m,L_n,T_n} \\ &= f_{L_m,T_m}(X, Y - T_t) \cdot f_{L_n,T_n}(L_t - X,\ T_t). \quad (42) \end{aligned}$$

Combining (41) and (42) together, we obtain the The j.p.d.f. for (X, Y, L_t, T_t) in the whole domain Ω_{X,Y,L_t,T_t} as follows [9]:

$$\begin{aligned} f_{X,Y,L_t,T_t} &= f_{\Omega_1}(X, Y, L_t, T_t) + f_{\Omega_2}(X, Y, L_t, T_t) \qquad (43) \\ &= f_{L_m,T_m}(X, T_t) \cdot f_{L_n,T_n}(L_t - X, Y - T_t) + \\ &\quad f_{L_m,T_m}(X, Y - T_t) \cdot f_{L_n,T_n}(L_t - X, T_t) \end{aligned}$$

Then the j.p.d.f. of (L_t, T_t) is obtained by

$$f_{L_t,T_t}(L_t, T_t) = \int_{\Omega_{X,Y}(L_t,T_t)} f_{X,Y,L_t,T_t} dX dY, \quad (44)$$

where $\Omega_{X,Y}(L_t, T_t)$ is the new domain for (X, Y) in terms of L_t and T_t. Knowing (L_m, T_m)'s domain Ω_{L_m,T_m} and (L_n, T_n)'s domain Ω_{L_n,T_n}, we can deduce $\Omega_{X,Y}(L_t, T_t)$ from (31),(32), (33) and (34).

4.2.4 J.P.D.F. Initialization

To carry on the above recursive computation of j.p.d.f., we have to set up the initial conditions starting from sinks. As sink's L_s and T_s are constants and buffers are always inserted at sink's upstream node with a wire between them, (1) and (2) should be used for j.p.d.f. initialization. After transformation of variables, we have:

$$c = (L_t - L_s)/l_i; \quad (45)$$

$$r = \frac{2(T_s - T_t)}{(L_t + L_s)l_i}. \quad (46)$$

Therefore, the Jacobian is given by

$$\begin{aligned} J &= \begin{vmatrix} \frac{\partial c}{\partial L_t} & \frac{\partial c}{\partial T_t} \\ \frac{\partial r}{\partial L_t} & \frac{\partial r}{\partial T_t} \end{vmatrix} \\ &= \begin{vmatrix} 1 & 0 \\ \frac{-2(T_s - T_t)}{(L_t+L_s)^2} & \frac{-2}{(L_t+L_s)l_i} \end{vmatrix} = \frac{-2}{(L_t + L_s)l_i}. \quad (47) \end{aligned}$$

As the j.p.d.f. for (c, r) is already known as $h(c, r)$, we can obtain the j.p.d.f. of (L_t, T_t) as follows:

$$\begin{aligned} f_{L_t,T_t} &= |J| \cdot h(c, r) \\ &= \frac{2}{(L_t + L_s)l_i} \cdot h(\frac{L_t - L_s}{l_i}, \frac{2(T_s - T_t)}{(L_t + L_s)l_i}) \quad (48) \end{aligned}$$

4.3 Pruning Rules

A straight-forward way to extend the conventional dominance relationship between two solutions in the presence of process variation is as follows: solution (L_1, T_1) is said to dominate solution (L_2, T_2) if condition $P(L_1 \leq L_2) = 1$ and $P(T_1 \geq T_2) = 1$ are satisfied. In other words, solution (L_1, T_1) always results in a larger required arrival time but with a less loading capacitance when compared to solution (L_2, T_2). The physical interpretation of this criterion is well understood. However, there are two problems when it comes to practical implementation. First, for a continuous j.p.d.f. for two random variables (L_t and T_t), the domain is usually defined over the whole feasible region: i.e., $0 \leq L_t \leq \infty$ and $0 \leq T_t \leq \infty$. Therefore, it is almost impossible to satisfy the conditions of $P(T_1 \geq T_2) = 1$ and $P(L_1 \leq L_2) = 1$ for any two given solutions. Second, assuming the first problem can be solved, there is no guarantee that the number of solutions after pruning will increase polynomially. Chances are that it will, most likely, grow exponentially, and this has been experimentally confirmed by [8]. In the following, we propose a new set of pruning rules that not only solves the above problems, but also supports designers' intuition.

We recognize that for designers, there is always a design goal in their minds when they compare different design alternatives. For the same example as shown in Section 3, if the arrival time for a design is required to be 100, then obviously solution (A) is the best choice, as it satisfies the required arrival time constraints without any uncertainty. On the other hand, if the arrival time is required to be 200, then solution (D) would be the best choice, as this gives the highest probability to satisfy the design requirement.

Based upon the above observations, we give the following definition of dominance relationship between two solutions, which is closely related to designers' willingness to accept uncertainty for a given design. Recall that the $(100\alpha)^{th}$ percentile of a p.d.f. $f(x)$ is a number π_α such that the area under $f(x)$ to the left of of π_α is α [10]. That is,

$$\alpha = \int_{-\infty}^{\pi_\alpha} f(x)dx. \quad (49)$$

In other words, π_α gives a measure of designers preference for certainty in choosing the design parameter x in the presence of variations, such that the final design would have x less than π_α with $(100\alpha)\%$ certainty.

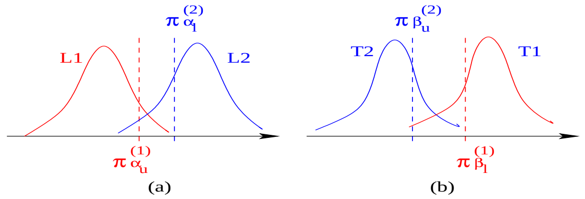

Figure 2: Graphic interpretation of dominance relationship between (L_1, T_1) and (L_2, T_2), where (a) refers to (50) and (b) refers to (51).

Suppose designers choose π_{α_l} and π_{α_u} as L_t's two percentiles with $0 \leq \alpha_l < \alpha_u \leq 1$, and π_{β_l} and π_{β_u} as T_t's two percentiles with $0 \leq \beta_l < \beta_u \leq 1$, which reflect designers preference for certainty in choosing different solutions (L_t, T_t) to the BIPV problem. Then solution (L_1, T_1) is said to dominate solution (L_2, T_2) if the following conditions are satisfied:

$$\pi_{\alpha_u}^{(1)} < \pi_{\alpha_l}^{(2)} \quad (50)$$

$$\pi_{\beta_l}^{(1)} > \pi_{\beta_u}^{(2)} \quad (51)$$

Another way to look at this dominance relationship is that solution (L_1, T_1) has a high probability of producing solutions with a larger required arrival time and a smaller loading capacitance. A graphical interpretation of this dominance relationship is shown in Fig. 2. Knowing α_l, α_u, β_l, and β_u, we can compute π_{α_l}, π_{α_u}, π_{β_l}, and π_{β_u} according to (49), which requires us to know the p.d.f. of L_t and T_t, respectively. As we already know the j.p.d.f. of (L_t, T_t) from section 4.2, the p.d.f. of L_t and T_t can be computed as two marginal p.d.f.'s of the j.p.d.f., respectively [10].

5. EXPERIMENT

A preliminary version of the above BIPV algorithm is implemented in C++ on a Linux machine. Because there is no existing work that can take into account the accurate computation of j.p.d.f. for (L_t, T_t) in the context of buffer insertion, direct comparison of our work with existing works is not possible. The only work that considered process variation for buffer insertion is [8], which does not consider either the interconnect parasitic variation or the device variation. Moreover, it is assumed that all random variables are independent, which is not correct in general. Therefore, in the following, we compare our BIPV design with the conventional design under nominal design parameters (DUN) similar to [6].

Parameter	Mean	Standard Deviation
Buffer input capacitance	0.04 pF	0.01 pF
Buffer output resistance	180 Ω	30 Ω
Buffer intrinsic delay	20 ps	5 ps
Unit wire capacitance	0.08 $fF/\mu m$	0.02 $fF/\mu m$
Unit wire resistance	0.08 $\Omega/\mu m$	0.02 $\Omega/\mu m$

Table 1: Device and interconnect variations.

For a given technology, foundry needs to characterize the j.p.d.f. for both devices and interconnects with consideration of process variations. Without loss of generality, we assume that the j.p.d.f. of both interconnect and devices are

1	2	3	4	5	6	7	8
Test Case	Terminals #	Segment Lengths (μm)	Delay Constraint (ps)	Conventional DUN Design		BIPV Design	
				Timing Violation %	Buffer #	Timing Violation %	Buffer #
1	5	1000-1400	2450	41%	4	33%	5
2	9	400-800	2200	35%	9	10%	9
3	9	600-1000	2300	32%	9	9%	10
4	9	800-1200	2400	43%	9	13%	9
5	9	1000-1400	2500	39%	9	15%	10

Table 2: Experimental result.

available to us as user inputs in the following. Normal distribution has been assumed for all test cases, and all parasitic values are adapted from [7] as shown in Table 1.

Different combinations of α_l, α_u, β_l, and β_u values have been tested for pruning. Because the choice of these values only reflects designers' preference to certainty and does not significantly affect our conclusions, we only report experiment results with $\alpha_l = \beta_l = 0.2$ and $\alpha_u = \beta_u = 0.8$ in the following.

Global interconnects with multiple terminals are used in our experiment. For simplicity, every test case is assumed to have a uniform required arrival time specified at different sink terminals. To make the test case challenging, we set the required arrival time at sinks to be half of the critical path delay without buffers. Results from five test cases are reported in Table 2. Column 2 gives the total number of terminal pins for test cases that are typical for global interconnects. The range of interconnect length between two adjacent Steiner points in the routing tree is reported in column 3. Column 4 gives the required arrival time specified at all sink terminals. Column 5 and 6 are the results measured at the root of the buffered routing trees from the conventional DUN design, while column 7 and 8 are the results from our BIPV design. Column 5 and 7 are the probabilities of the designed buffered routing tree that would violate the required timing constraints. The number in column 6 and 8 are the total number of buffers inserted.

According to Table 2, it clearly shows that the conventional DUN design exhibits very large timing violation when process variations are considered, while our process variation aware BIPV design consistently results in smaller timing violation than the DUN design. Moreover, when the test case becomes larger, the improvement of BIPV design over the DUN design is more significant. For example, for test case 4, the conventional DUN design gives 43% timing violation, while our BIPV design results in only 13% violation, and the relative reduction is 30%. In terms of the total number of buffers inserted in the final buffered routing tree, the BIPV design uses sightly more buffers than the conventional DUN design, but there is never more than one extra buffer in all experiments. This further confirms that the BIPV algorithm inserts buffers at locations such that the probability of achieving timing constraints is optimized.

6. CONCLUSION AND FUTURE WORK

We have proposed a probabilistic methodology to solve the problem of buffer insertion considering process variations for both devices and interconnects. A very recent work solved the buffer insertion with wire length uncertainty by assuming that there is no correlation between the variations of required arrival time and downstream loading capacitance. We have shown that this assumption does not hold in general and that the j.p.d.f. should be employed to accurately capture the above correlation. We have developed an efficient algorithm to calculate the j.p.d.f. recursively. Our preliminary experiment results have shown that compared to the conventional buffer insertion under the nominal parameters, the proposed variation aware buffer insertion can reduce the probability of timing violation by up to 30%.

Note that the proposed methodology is very general and can be extended to consider other process variation effects. For example, we have assumed that there is no correlation between device and interconnect variations, and that wires not sharing downstream paths are independent. However, in a modern process, the variation between devices and interconnects may be interdependent, and wires nearby may also exhibit correlation due to chemical-mechanical planarization (CMP). One of our future work is to extend the proposed methodology to handle more general process variations, like across-chip-line-width variation (ACLV). We will also study the process variation aware routing topology generation and device sizing problems in the future.

7. REFERENCES

[1] S. Borkar, T. Karnik, S. Narendra, J. Tschanz, A. Keshavarzi, and V. De, "Parameter variations and impacts on circuits and microarchitecture," in *DAC 03*, Jun 2003.

[2] K. A. Bowman, S. G. Duvall, and J. D. Meindl, "Impact of die-to-die and within-die parameter fluctuations on the maximum clock frequency distribution," in *ISSCC 01*, Feb 2001.

[3] C. Visweswariah, "Death, taxes and failing chips," in *DAC 03*, Jun 2003.

[4] J. Jess, K. Kalafala, S. Naidu, R. Otten, and C. Visweswariah, "Statistical timing for parametric yield prediction of digital integrated circuits," in *DAC 03*, Jun 2003.

[5] A. Agarwal, D. Blaauw, V. Zolotov, and S. Vrudhula, "Computation and refinement of statistical bounds on circuit delay," in *DAC 03*, Jun 2003.

[6] L. P. P. P. van Ginneken, "Buffer placement in distributed RC-tree networks for minimal Elmore delay," in *Proc. IEEE Int. Symp. on Circuits and Systems*, pp. 865–868, 1990.

[7] C. J. Alpert and A. Devgan, "Wire segmenting for improved buffer insertion," in *DAC 97*, Jun 1997. 588-593.

[8] V. Khandelwal, A. Davoodi, A. Nanavati, and A. Srivastava, "A probabilistic approach to buffer insertion," in *ICCAD 03*, 2003.

[9] R. Hogg and A. Craig, *Introduction to Mathematical Statistics*. Macmillan, 1995.

[10] R. Hogg and E. Tanis, *Probability and Statistical Inference*. Prentice Hall, 2001.