Delay Uncertainty Reduction by Interconnect and Gate Splitting

Vineet Agarwal Jin Sun Alexander Mitev Janet Wang

e-mail: {vagarwal, sunj, mitev, wml}@ece.arizona.edu
Electrical and Computer Engineering Department
University of Arizona
Tucson, AZ - 85721
Tel: 520-621-2434
Fax: 520-621-8076

ABSTRACT

Traditional timing variation reduction techniques are only able to decrease the gate delay variation by incurring a delay overhead. In this work, we propose novel and effective splitting based variation reduction techniques for both interconnect and gate. We developed a new tool called TURGIS: *Timing Uncertainty Reduction by Gate-Interconnect Splitting* which reduces the timing variations of a circuit and presents little delay overhead at the primary output. It is shown that using splitting on interconnect can reduce the Chemical-Mechanical Polishing (CMP) induced *dishing* effect and can result in decrease at an average of 5% in mean interconnect delay in addition to its variation. Improvements of up to 30% are achieved on timing variation for gates of various size while reduction of 55% can be observed in interconnect delay variation.

1. INTRODUCTION

Since the advent of nanometer technology, the critical dimensions in today's digital circuits are diminishing continuously. The impact of manufacturing process such as the Chemical-Mechanical Polishing (CMP) at the present leads to random variations in circuit parameters. The standard deviation in circuit response can go as high as 15%-40% of their expected design value [1, 2]. As a result the timing profile of a circuit end up with a 'wide-spread' appearance, which in turn causes yield deterioration due to timing violations. Various techniques have been proposed in the literature to encounter the detrimental effects of these random perturbations. Among these, *Sizing* has been one of the most effective techniques. Intuitively, manufacturing uncertainties in a larger transistor or interconnect will be proportionately less as compared to that of smaller ones. *Sizing* employs the same principle and assigns various gate and interconnect sizes to minimize the delay variations at the primary outputs of the circuits.

A survey of past literature reveals that circuit timing variation reduction has been researched for several years. For example, [3] and [4] use Lancelot to do gate sizing of circuits and yield improvements of 15% are achieved on average. Lagrangian Relaxation is used in [5] to achieve similar results and improvements are also reported when compared to worst case design. [6] reduced the timing violations by using geometric programming. The aforementioned techniques were successful in improving the yield by decreasing the timing variation, but the gain came at a cost: they incurred an increase in circuit area (transistor drain-source area) and had an increase in the mean delay value at the outputs. This is because sizing up gate is the direct cause of the increase in the gate capacitance. In [7] the authors provide a trade-off between the mean value of the delay and its variance while doing gate sizing on the worst negative statistical slack path of a circuit. As one of the first papers, [8] proposed a gate splitting mechanism during the placement stage to reduce the timing variation with trade-off between variation reduction and area increase.

In this paper, we propose a interconnect-gate splitting technique to reduce the timing variations that come from manufacture process such as CMP. Our major contributions are listed as follows. 1) We include interconnect splitting to reduce the variability caused by CMP. As pointed out by [9], interconnect delay has become comparable to gate delay and its delay variation cannot be neglected. 2) We provide mathematic proofs to reveal the relationship between delay variation reduction and splitting. We demonstrate that due to correlation there exists a beneficial design space which decides whether or not to split any particular gate or interconnect. 3) The proposed methodology has been implememted in our new tool TURGIS: *Timing Uncertainty Reduction by Gate-Interconnect Splitting.* Integrated with existing statistical static timing analysis (SSTA) tool, TURGIS can provide a list of split candidates to placement algorithms such as the spin-off gate placement algorithm in [8]. And with the gate-Steiner-tree distance provided by placement algorithm, TURGIS takes into consideration the extra interconnect cost caused by gate splitting.

In the process of variation reduction TURGIS doesn't increase the transistor drain-source area. However there is a nominal increase in cell area due to increased number of gates and extra interconnect lines. Our experimental results show that when applied on local interconnects the variation was reduced by 50% while reduction of 55% was noted for global interconnects for 130nm technology. When applied on standard benchmark circuits and NAND chains improvements of nearly up to 30% in timing variation were recorded. This technique can be applied together with existing variation reduction techniques to provide trade-offs among the mean value of the delay, its variance and the area cost.

The rest of the paper is organized as follows. Section 2 introduces the idea of *splitting* and then provides its implementation on a simple inverter circuit while providing necessary mathematical modeling and proofs to support the results. Section 3 presents the TURGIS algorithm for application on digital circuits and talks about how to integrate with

1-4244-0630-7/07/$20.00 ©2007 IEEE.

Statistical Timing Analysis of a split circuit. Experimental results are presented in Section 4 while Section 5 concludes our work.

2. SPLITTING TECHNIQUE

The Splitting techniques described in the current paper are not exactly the same as "hardware redundancy". In the traditional hardware redundancy concept, one has one component ie. p_1 to start with, and then he/she has two components in which case p_1 and p_2 with $p_1 = p_2$. The total component size doubles. Just like the name, the splitting techniques in this paper emphasize on "splitting" itself. That is, the original component p_1 splits into two equal size components p_{c1} and p_{c2}. The size of p_{c1} together with the size of p_{c2} equals the size of p_1. It is obvious to see that splitting keeps the same total component size. We provide the definition of splitting as follows:

DEFINITION 1. *Splitting is defined as the technique of substituting a parent (larger) entity by two children (smaller) entity of half the parent size and connecting them in parallel in place of the parent entity.*

2.1 Interconnect Splitting

Interconnect splitting arises directly from the CMP process. While planarizing the metal layers in the CMP step, the central portion of the wire cross section gets scooped more than the edges. The "scooping" of the cross section results in increase of interconnect resistance. This is called *dishing* of interconnect layers as the metal lines take the shape of the dish. Chang [10] described the dishing effect in detail and introduced corresponding models. Figure 1 depicts the cross section of a wire with dishing effect.

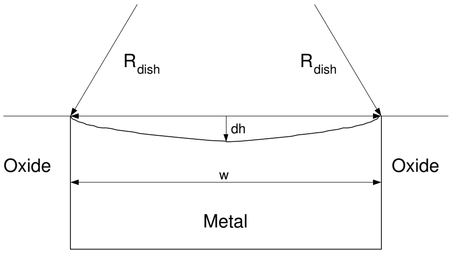

Figure 1: Metal Dishing model

R_{dish} is called the dishing radius, w is the metal width, h is the metal height and dh the height of metal lost at the center of the line. dh can be written as $dh(w, R_{dish}) = R_{dish} - \sqrt{R_{dish}^2 - \frac{w^2}{4}}$ R_{dish} is assumed to be 4×-6× of the metal width (w). In this scenario the interconnect resistance per unit length is given as

$$r_d = \frac{\rho}{wh + 0.5w\sqrt{R_{dish}^2 - \frac{w^2}{4}} - R_{dish}^2 \sin^{-1} \quad \frac{w}{2R_{dish}}} \tag{1}$$

If a wide metal line (width W_{total}) is replaced by N number of metal lines connected in parallel each of width W_{total}/N then the effect of dishing can be reduced effectively. The basic idea of interconnect splitting is shown in Figure 2. It can be seen in the figure that the dishing effect is less prominent in narrower metal lines. The parasitic capacitance of the original interconnect and the split interconnect is nearly the same. The only difference is in line resistance. Thus, if d_{org} is the delay of original wire and d_{split} is the delay of the split configuration then $d_{org} = r_d(w, R_{dish})C_{total}$ and $d_{split} = \quad r_d^1 \| \cdots \| r_d^N \quad C_{total}$ where $r_d^i = r_d(w/N_s, R_{dish})$. where N_s is the number of wires into which the original wire is split. Chang in [10] suggested $N_s = 3 \quad or \quad 4$ to achieve best performance for 0.25 μm technology. In the experimental part, we show that similar conclusion can be obtained for deep submicrion technologies.

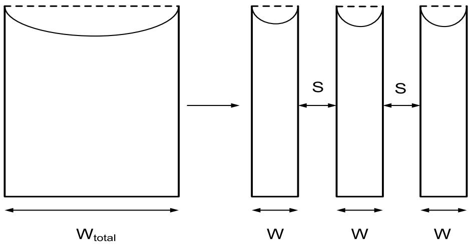

Figure 2: Interconnect splitting to subside dishing effect

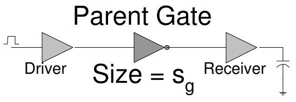

Figure 3: Original Gate before Splitting

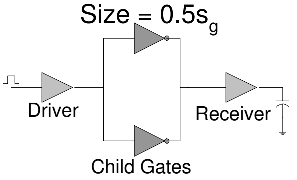

Figure 4: Gate after Splitting

2.2 Gate Splitting

Figure 3 shows a simple connection of an inverter (parent gate) of size s_g, while its split inverter equivalent is shown connected in place of parent inverter gate in Figure 4. The size of both the children inverter is $0.5s_g$. These two figures demonstrate the basic principle of gate splitting. Now the two children gates do not present any extra load on driver gate, since the total size of the children is the same as that of the parent. And similarly the two children do not provide any extra driving power to the receiver. So functionally the parent gate and children gates are equivalent.

It is important to state that gate splitting should not be viewed as multi-fingered gates. Multi-fingered gates have fixed distance between any adjancent two gates while gate splitting allows designers to adjust the distance according to the tradeoffs between area cost and delay variations. As gate effective length L_{eff} (or critical dimension) is the main reason that causes delay variaion, gate splitting in the current paper is used only when gate effective length approaches deep submicron region.

The intuition behind gate splitting mechanism is that the variance of the sum of two less-than-perfectly correlated random variables (each random variable represents the delay of each splitted gate) is always less than the variance of the sum of two perfectly correlated ones (the delay of unsplitted gate).

The following section will provide a mathematical analysis of the splitting technique and reason out the reduction in variation through it.

2.3 Delay Metric with Splitting Technique

The Elmore delay model for circuit sketched in Figure 3 can be represented as:

$$d_s^p = 0.69 \quad R_D C_D + (R_D + R_w/N_s)(C_L + C_w)s_g + \frac{r}{s_g}C_R \tag{2}$$

where d_s^p is delay at the output of the parent gate. R_D is equivalent driving resistance of the driver. C_D is self loading capacitance of the driver. C_R is input capacitance of the receiver. r is driving resistance of a minimum sized inverter. C_L is Load Capacitance of a minimum sized inverter. s_g is size of the inverter. N_s is number of wire split. C_w is wire capacitance. R_w is wire resistance. Similarly the Elmore delay model for the circuit sketched in Figure 4 can be written as:

$$d_s^c = 0.69[R_D C_D + (R_D + R_w/N_s)(C_L + C_w)(s_1 + s_2) + \frac{r}{(s_1 + s_2)} C_R] \quad (3)$$

where d_s^c is the delay at the output of the split child gate, and s_1 and s_2 are size of children gate. Mean value of both s_1 and s_2 is equal to $0.5s_g$.

Now just for analysis purpose driver and receiver gates are assumed to be sized equally and to be free from perturbation. The gate size s_g is a function of L_{eff} and W width of transistors. Since $W = NL_{eff}$, where N is a integer ≥ 1. We view gate size s_g as a function of L_{eff}. If L_{eff} is modeled as a random variable, then $s_g = f(L_{eff})$ is also a random variable. We assume that s_g has any kind of gaussian or non-gaussian probability distribution. In this analysis we will assume that perturbations are normally distributed because of simplicity of expressions. But the forthcoming proofs can be easily extended to non-gaussian cases. So, let's assume that s_g is normally distributed with a *pdf* of $N(\mu_s, \sigma_s)$ and the children gates having similar *pdf* of the form $N(\mu_s', \sigma_s')$ with $\mu_s' = 0.5\mu_s$ and the correlation coefficient between them being ρ. The values of σ_s and σ_s' can be written in form of μ_s according to (6). With these simplification and disregarding the constant terms the two delay values can be re-written in the form:

$$d_s^p \simeq s_g + \frac{1}{s_g} \qquad d_s^c \simeq s_c + \frac{1}{s_c}$$

where $s_c = s_1 + s_2$. Therefore we can write $s_c \sim N(\mu_{s_c}, \sigma_{s_c})$ where $\sigma_{s_c}^2 = \sigma_{s_1}^2 + \sigma_{s_2}^2 + 2\rho\sigma_{s_1}\sigma_{s_2}$. This can be reduced to $\sigma_{s_c} = \sigma_s' \sqrt{2(1+\rho)}$ since $\sigma_{s_1} = \sigma_{s_2} = \sigma_s'$.

Now for any random variable x with *pdf* of the form $N(\mu, \sigma)$ we can write

$$E \ \frac{1}{x} = \frac{1}{\mu}\left[1 + \ \frac{\sigma}{\mu}^{2} + 3 \ \frac{\sigma}{\mu}^{4} + 15 \ \frac{\sigma}{\mu}^{6}\right] \quad (4)$$

$$\sigma \ \frac{1}{x} = \frac{\sigma}{\mu^2}\sqrt{\left[1 + 8 \ \frac{\sigma}{\mu}^{2} + 69 \ \frac{\sigma}{\mu}^{4}\right]} \quad (5)$$

The proof for this derivation can be found at the end of this section. Now using these values we can write the following:

$$E \ x + \frac{1}{x} = E(x) + E \ \frac{1}{x}$$

$$\sigma \ x + \frac{1}{x} = \sqrt{E\left[\ x + \frac{1}{x} \ ^{2}\right] - E^2 \ x + \frac{1}{x}}$$

$$= \sqrt{\sigma^2(x) + \sigma^2 \ \frac{1}{x} \ + 2 - 2E(x)E \ \frac{1}{x}}$$

If we make $x = s_g \sim N(\mu_s, \sigma_s)$ we can get the value for $E(d_s^p)$ and $\sigma(d_s^p)$ and similarly values for $E(d_s^c)$ and $\sigma(d_s^c)$. Using these values and substituting corresponding values of σ_s and σ_s' in terms of μ_s we can write:

$$\frac{\sigma(d_s^c)}{\sigma(d_s^p)} = 0.7282 - 0.0076\mu_s + 0.0077\mu_s^2 - 0.0015\mu_s^3$$

for $\rho = 0$ while for $\rho = 0.5$ we can write

$$\frac{\sigma(d_s^c)}{\sigma(d_s^p)} = 0.8485 - 0.0098\mu_s + 0.0007\mu_s^2 - 0.0001\mu_s^3$$

The derivation for this ratio is omitted here for sake of brevity. Similar expressions can be obtained for non-gaussian cases and can be shown to be less than 1 for various different values of μ_s and ρ. For example if the inverter gate sizes are assumed to be log-normally distributed then we can derive that

$$\frac{\sigma(d_s^c)}{\sigma(d_s^p)} = 0.5360 - 0.1825\mu_s + 0.0217\mu_s^2 - 0.0009\mu_s^3$$

We can see the ratio of delay deviation of the split circuit and that of original circuit is less than 1 for various values of μ_s within our design range. Figure 5 plots the relative decrease in the delay variance for varying mean gate size (μ_s) and correlation coefficient (ρ). We can notice the reduction in delay variation in the split circuit, which exhibits the inherent advantage of splitting any gate. Also, not all the gates will exhibit a decrease in variation due to splitting and thereby splitting those gates can have a detrimental effect on the delay variance. We can thereby choose only those gates for splitting which render positive decrease in delay variance. The region above the dotted line in Figure 5 is called the beneficial design region. A gate is split only if it has the parameters (μ_s and ρ) which lie in this region. Now lets take a look at how to derive (6). First we introduce a size perturbation model. Due to random perturbations while fabricating a circuit, the actual fabricated value deviates from its expected mean value. In this model, the gate sizes have been modeled as normal random variables. It can be intuitively concluded that larger gates will suffer lesser amount of deviation as compared to smaller gates. But the presence of random fluctuations, forbid the decrease of deviation beyond a particular amount, even if we increase the size to large values. In this work, the amount of deviation in the mean value of a gate size of s is assumed to be

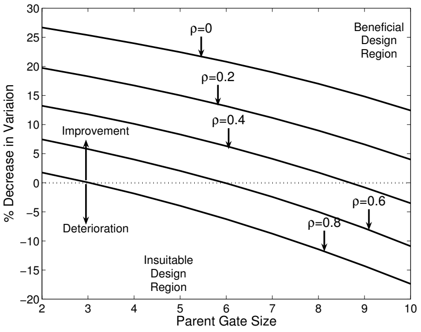

Figure 5: Change in delay variance of a simple 2 inverter chain for various values of ρ

$$\sigma_s = \frac{(31 - s)}{3} \ \% \ of \ s \quad (6)$$

This is an empirical model, based on data provided in [1, 2]. Thus a gate of size 1× will have standard deviation of 10% while a gate of size 10× will have standard deviation of 7%. While the transistors of any particular gate will lie close to each other while fabrication, we can safely assume that deviation in critical dimensions of each transistor will be approximately equal due to correlation factors.

For $x \sim N(\mu, \sigma)$ we can assume that $\sigma \leq 0.1\mu$ according to the model listed in (6). Now re-writing x as $(\mu + 3\sigma\xi)$, we can deduce that $\xi \sim N(0, 1/3)$. Now:

$$\frac{1}{x} = \frac{1}{\mu + 3\sigma\xi} = \frac{1}{\mu \ \ 1 + \frac{3\sigma}{\mu}\xi}$$

Now $\frac{3\sigma}{\mu}\xi \ \leq 1$ since max $\ \frac{\sigma}{\mu} \ \leq 0.1$, we can write:

$$\frac{1}{x} = \frac{1}{\mu} \ \ 1 + \frac{3\sigma}{\mu}\xi \ ^{-1}$$

$$\frac{1}{x} \simeq \frac{1}{\mu}\left[1 - \frac{3\sigma}{\mu}\xi + \ \frac{3\sigma}{\mu} \ ^{2}\xi^2 - \ \frac{3\sigma}{\mu} \ ^{3}\xi^3 + \ldots\infty\right]$$

Now since $\xi_i \sim N(0, 1/3)$, $E\left(\xi_i^n\right) = 0$ for $n = 2k+1$ and $E\left(\xi_i^n\right) = 1.3\ldots(n-1)\sigma_i^n$ for $n = 2k$, where $\sigma_i = 1/3$.

$$E \quad \frac{1}{x} \quad = \quad \frac{1}{\mu}\left[1 + \quad \frac{3\sigma}{\mu} \quad ^{2} \sigma_i^2 + \quad \frac{3\sigma}{\mu} \quad ^{4} (1.3)\,\sigma_i^4 + \ldots\infty\right]$$

$$E \quad \frac{1}{x} \quad = \quad \frac{1}{\mu} + \frac{1}{\mu}\sum_{k=1}^{\infty} \quad \frac{3\sigma\sigma_i}{\mu} \quad ^{2k} \frac{(2k-1)!}{2^{k-1}(k-1)!}$$

Thus now if we ignore higher order terms we can simplify as:

$$E \quad \frac{1}{x} \quad = \quad \frac{1}{\mu}\left[1 + \quad \frac{\sigma}{\mu} \quad ^{2} + 3 \quad \frac{\sigma}{\mu} \quad ^{4} + 15 \quad \frac{\sigma}{\mu} \quad ^{6}\right]$$

Following similar steps we can write:

$$E^2 \quad \frac{1}{x} \quad = \quad \frac{1}{\mu^2}\left[1 + 2 \quad \frac{\sigma}{\mu} \quad ^{2} + 7 \quad \frac{\sigma}{\mu} \quad ^{4} + 36 \quad \frac{\sigma}{\mu} \quad ^{6}\right]$$

$$E\left[\quad \frac{1}{x} \quad ^{2}\right] \quad \simeq \quad \frac{1}{\mu^2}\left[1 + 3 \quad \frac{\sigma}{\mu} \quad ^{2} + 15 \quad \frac{\sigma}{\mu} \quad ^{4} + 105 \quad \frac{\sigma}{\mu} \quad ^{6}\right]$$

Therefore again by ignoring higher order terms we can write:

$$\sigma \quad \frac{1}{x} \quad = \quad \frac{1}{\mu}\sqrt{\left[\quad \frac{\sigma}{\mu} \quad ^{2} + 8 \quad \frac{\sigma}{\mu} \quad ^{4} + 69 \quad \frac{\sigma}{\mu} \quad ^{6}\right]}$$

3. TURGIS ALGORITHM

Interconnect splitting may be applied on any interconnect at any metal layer with very little extra cost. Splitting gates on the critical path, however, may not be the most optimal choice as we need extra interconnect lines for each gate splitting. We will follow a simple algorithm *Timing Uncertainty Reduction by Gate and Interconnect Splitting* (TURGIS) to optimally choose the splitting candidates (gates and/or interconnects) for maximum delay variation reduction. This new algorithm may be integrated with statistical static timing analysis (SSTA) to provide a list of split candidates (either gate or interconnect) for spin-off gate placement algorithm [8]. At the same time, the distance between a spin-off gate G_f and steiner tree T_i: $d_f = d(G_f, T_i)$ defined in [8] can offer delay estimation adjustment for TURGIS. For example, the delay after splitting in (3) can be modified as

$$d_s^c = 0.69[R_D C_D + (R_D + R_w/N_s)(C_L + C_w)s_1$$
$$(R_D + (r_w d_f)/N_s)(C_L + (c_w d_f))s_f + \frac{r}{(s_1 + s_f)}C_R] \qquad (7)$$

Here r_w and c_w are the per unit length wire resistance and capacitance respectively. d_f is the distance between a spin-off gate G_f and steiner tree T_i and s_f is the size of spin-off gate.

Since our aim is to reduce the overall timing variance of a circuit, we can apply splitting to those elements (gates or interconnects) which contribute most to output delay variance. The algorithmic flow of decision making on the choice of circuit elements is outlined in Algorithm 3.1. As the circuit elements on the critical path contribute most to the delay and finding critical path deterministically can lead to false critical path [7], we trace the critical path statistically from each primary output after SSTA. Once all the elements on statistically longest path are identified, sensitivity is assigned to each of those elements. And at last all the elements above the sensitivity threshold are chosen to be split. Also, gates with detrimental effect of splitting (which are not in beneficial design region) are excluded from this list. Lower the value of this threshold, more number of elements will be selected for splitting.

Algorithm 3.1: TURGIS(circuit)

```
$\mathcal{A} \leftarrow$ STA($circuit$)
$\mathcal{PO} \leftarrow$ Extract primary outputs($circuit$)
$\mathcal{PI} \leftarrow$ Extract primary inputs($circuit$)
for each $\lambda \in \mathcal{PO}$
  do
    $slp_\lambda = \{\emptyset\}$
    PUSH($slp_\lambda, \lambda$)
    parent node $\leftarrow \lambda$
    while parent node $\notin \mathcal{PI}$
      do
        fan-ins $\leftarrow$ Find Fan-In(parent node)
        $\nu \leftarrow$ FIND LARGEST(fan-ins)
        PUSH($slp_\lambda, \nu$)
        parent node $\leftarrow \nu$
for each $g \in slp$
  do
    $s_g \leftarrow$ Compute Sensitivity($g$)
    if $s_g > threshold$
      then
        $\mathcal{R} \leftarrow$ Locate($g$)
        if $\mathcal{R} \subset$ BeneficialDesignRegion
          then circuit $\leftarrow$ Split($g$)
return (circuit)
```

We require two operations in TURGIS: random variable comparison (in FIND LARGEST()) and sensitivity computation (in Compute Sensitivity( )). After running SSTA to get all the delay PDF at the output of gates, we need to identify the statistical longest path by comparing delays. As delays are random variables, we compare two random variables following Theorem 1[11].

THEOREM 1. *For any 2 random variables x and y, x is termed as statistically larger than y if $\frac{(\mu_x - \mu_y)}{\sqrt{\sigma_x^2 + \sigma_y^2}} \geq 3$ or otherwise y is termed as statistically larger than x if $\frac{(\mu_y - \mu_x)}{\sqrt{\sigma_x^2 + \sigma_y^2}} \geq 3$ and if none of the above conditions is true then $\frac{\partial\Upsilon(x,y)}{\partial x} \geq \frac{\partial\Upsilon(x,y)}{\partial y} \Rightarrow x \geq y$ where the function $\Upsilon(x,y)$ is defined as following $\alpha = \sqrt{\sigma_x^2 + \sigma_y^2}$, $\beta = (\mu_x - \mu_y)/\alpha$, $\xi_1 = \mu_x\phi(\beta) + \mu_y\phi(-\beta) + a\ \varphi(\beta)$, $\xi_2 = (\mu_x^2 + \sigma_x^2)\phi(\beta) + (\mu_y^2 + \sigma_y^2)\phi(-\beta) + (\mu_x + \mu_y)\ \alpha\ \varphi(\beta)$, $\Upsilon = \xi_2 - \xi_1^2$, $\varphi(\alpha) = \frac{1}{\sqrt{2\pi}}e^{-\frac{\alpha^2}{2}} and \phi(\alpha) = \int_{-\infty}^{\alpha}\varphi(t)dt$*

.

Now lets take a look at sensitivity computation. For any circuit with N elements and with size profile as $S = \{s_1, s_2, \ldots, s_N\}$, the output delay can be modeled in form of

$$d_o(S) = \alpha + \sum_{i=1}^{N}\beta_i s_i + \sum_{i=1}^{N}\gamma_i s_i^2 + \sum_{i=1}^{N-1}\sum_{j=i}^{N}\delta_{ij}s_i s_j \qquad (8)$$

The coefficients of the (8) can be found out by any technique such as Response Surface Modeling [12] or least square fitting techniques. We have found that for performing least square fit with high accuracy and with least number of sample points, one can make choice of sample points termed as collocation point, as defined in [13].

After d_o is constructed as a function of $s_1, s_2, \ldots, s_N$ the following steps can be followed to compute the sensitivity of each element (g_i) of size s_i.

$$\begin{aligned}
\mathcal{F}(S) &= f_{s_1 s_2 \cdots s_N}(s_1, s_2, \cdots, s_N) \\
\mu(d_o) &= \int\!\!\int \cdots \int d_o\mathcal{F}(S)ds_1 ds_2 \cdots ds_N \\
\sigma^2(d_o) &= \int\!\!\int \cdots \int \left[d_o - \mu(d_o)\right]^2 \mathcal{F}(S)ds_1 ds_2 \cdots ds_N \\
\mu(d_o|s_i) &= \int\!\!\int \cdots \int d_o\mathcal{F}(S|s_i)ds_1 \cdots ds_{i-1}ds_{i+1} \cdots ds_N \\
sen(g_i) &= \frac{1}{\sigma^2(d_o)}\int \left[\mu(d_o|s_i)\right]^2 f(s_i)ds_i
\end{aligned}$$

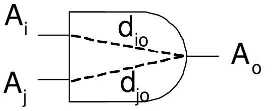

Figure 6: Original Arrival time model for SSTA

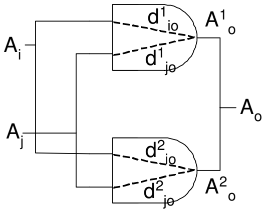

Figure 7: New Arrival time model for SSTA

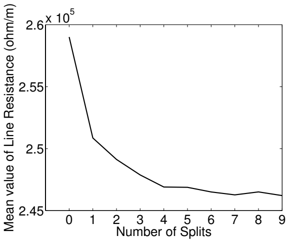

Figure 8: Mean Interconnect Resistance per unit length

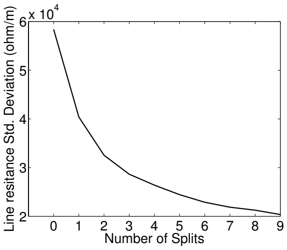

Figure 9: Standard Deviation of Interconnect Resistance per unit length

where $\mathcal{F}(S)$ is the joint probability distribution of the gate sizes and $f(s_i)$ is the probability distribution function of s_i.

To include the effects of multi-driver we add on a MIN operator to the methodology. Traditionally the SSTA was performed on circuit as sketched in Figure 6. In this case

$$A_o = \max(A_i + d_{io}, A_j + d_{jo})$$

But for performing SSTA on the split circuit as sketched in Figure 7 a simple MAX operation cannot be used. In this new case we define A_o as:

$$A^1_o = \max(A_i + d^1_{io}, A_j + d^1_{jo}) \quad A^2_o = \max(A_i + d^2_{io}, A_j + d^2_{jo})$$
$$A_o = \min(A^1_o, A^2_o)$$

where A_o is the arrival time at the output node, A^1_o and A^2_o are arrival times at the output node of the two child gates, d_{io} and d_{jo} are the delay from nodes i and j to o. Where as an superscript of 1 or 2 denote the corresponding delay for the child gates. The rationale behind using a MIN operation is that the output node starts to switch the moment any one of the child gates starts to switch. Therefore the delay at that output node becomes the minimum delay of the two inverter. This can also be confirmed using SPICE simulations.

MAX operator can be implemented as multiplication of Cumulative Distribution Function (CDF) of the two arrival times [14].

$$A_o = \max(A_a, A_b) \quad \Rightarrow \quad C_o(t) = C_a(t)C_b(t)$$

Where $C_o(t)$ is the CDF of output arrival time and $C_a(t)$ and $C_b(t)$ are CDF of input arrival times. Similarly MIN operator can be implemented as

$$A_o = \min(A_a, A_b) \quad \Rightarrow \quad C_o(t) = C_a(t) + C_b(t) - C_{ab}(t, t)$$

where $C_{ab}(t, t)$ is the joint CDF of a and b.

4. EXPERIMENTAL RESULTS

The implementations were done in MATLAB and HSPICE while the transistor and interconnect model used were the Berkeley PTM model [15]. The timings are reported for an implementation on a computer running Windows OS with 1.5GHz clock frequency and 512Mb RAM. A simple interconnect of length 1 centimeter was chosen for split. This

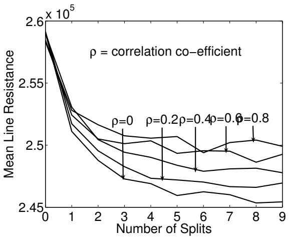

Figure 10: Effect of correlation on mean interconnect resistance

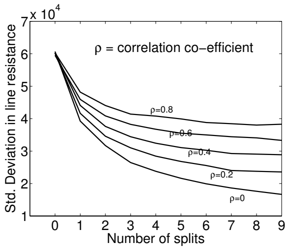

Figure 11: Effect of correlation on interconnect resistance standard deviation

setting will used only for experimentation purposes to show the advantages of splitting. Similar reductions can also be achieved for local and global interconnects of shorter lengths. In this case the wire width (w) and height (h) were assumed to be normal random variables. R_{dish} was assumed to be 4× the wire width. Figure 8 and Figure 9 show the mean and standard deviation of the resistance of the interconnect line. It is noticed that with increasing number of splits both the mean and the standard deviation of resistance decreases. Variation reduction of up to 55% is achieved. But it becomes nearly constant for splits more than three. Thus we conclude that 3 is the optimal number of splits, as increasing the number of splits will not give us more reduction and will also result in higher cell layout area. Figure 10 and Figure 11 show the effect of spatial distance between the split lines. Since a smaller spatial distance translate to higher correlation coefficient between the widths and heights of the adjacent wires, we see that with increasing correlation coefficient the amount of reduction reduces. For achieving lower correlation coefficient the wires has to be layed out far which causes extra area overhead. Thus there is a tradeoff in amount of reduction achieved and the cell area. The proposed `TURGIS` was implemented and results are demonstrated for various ISCAS benchmark circuits and Nand chains. The ISCAS circuits were pre-mapped using commercial logic synthesis tool, and then gate sizes were assigned randomly to the gates. `TURGIS` was then applied on these mapped and sized circuit.

First, the primary splitting technique was applied on a Nand chain of various lengths. The results for chain lengths of 5, 10 and 20 are tabulated in Table 1. Two different setting were used for experimentation. At first random sizes were assigned to the gates and next all gates were sized the same (6× in this example). Average improvements of 25% for random sizing and 22% for equal sizing were achieved.

Table 2 lists the various output delay statistics values of various ISCAS benchmark circuits for a given gate sizing profile. It also lists the number of gates (N) present in the circuit. Table 3 lists the delay value statistics if all the gates in the circuit are split. Improvements up to 30% in the variance is recorded while the average improvement recorded was 5.37% in mean value and 21.73% in variance. For `TURGIS` average improvements are plotted against sensitivity threshold in Figure 12. N_C is the number of gates on the statistically longest path and N_S is actual number of sensitive gates split. Increasing the sensitivity threshold causes more sensitive gates to be split and results in larger decrease in variance. This can be noticed in Figure 12 that average improvements turn out to be monotonically increasing function of decreasing sensitivity threshold or increase number of sensitive gates split. If the sensitivity threshold is decreased to very small value then generally all the gates on critical paths are selected to be split since the output variance is most sensitive to critical path perturbations and the results approach similar reduction as compared to splitting

Table 1: Results of Gate splitting on various length NAND gate chains

Number of	Sizing	Original Chain			Split Gate Chain			Improvement		
		Statistics			Statistics			(% Decrease)		
gates	Profile	μ	σ	σ/μ	μ	σ	σ/μ	μ	σ	σ/μ
5	$\Re$	87.83	1.404	0.015	87.83	1.024	0.011	0.00	27.06	26.67
5	$\aleph$	82.03	0.949	0.011	81.96	0.805	0.009	0.08	15.17	18.18
10	$\Re$	198.65	3.395	0.016	198.34	2.514	0.012	0.15	25.95	25.00
10	$\aleph$	175.54	0.995	0.005	175.42	0.847	0.004	0.06	14.87	20.00
20	$\Re$	398.09	3.322	0.008	397.61	2.719	0.006	0.12	18.15	25.00
20	$\aleph$	362.93	1.121	0.003	362.44	0.756	0.002	0.13	32.56	33.33

$\Re \Rightarrow$ Random Sizing $\aleph \Rightarrow$ All gates size = 6

Table 2: Original Output Delay statistics for ISCAS Benchmark circuits

Circuit	Original			
	statistics (ps)			
name	N_T	μ	σ	(σ/μ)
c17	6	42.97	2.98	0.069
c432	160	588.20	33.22	0.053
c499	202	962.49	133.86	0.140
c880	383	200.36	6.26	0.049
c1355	546	1212.65	95.54	0.079
c1908	880	1444.95	159.95	0.100
c2670	1193	410.43	33.38	0.110
c3540	1669	2308.53	52.19	0.032

Table 3: Output Delay statistics for ISCAS Benchmark circuits split on all the gates

Circuit	Split on All gates				Improvements				CPU
	Statistics (ps)				$\frac{N}{N_T}$	(% Decrease)			time
name	N	μ	σ	(σ/μ)		$\Delta\mu$	$\Delta\sigma$	(σ/μ)	(sec)
c17	6	42.32	2.31	0.054	1	1.51	22.48	21.73	0.02
c432	160	558.26	24.01	0.041	1	5.09	27.72	22.64	0.22
c499	202	915.06	116.42	0.129	1	4.92	13.02	7.85	0.35
c880	383	193.45	5.06	0.038	1	3.44	19.16	22.44	0.51
c1355	546	1156.67	82.55	0.072	1	4.61	13.59	8.86	0.89
c1908	880	1332.45	111.48	0.075	1	7.78	30.30	25.00	2.52
c2670	1193	385.22	23.85	0.085	1	6.14	28.55	22.73	4.21
c3540	1669	2089.71	42.26	0.028	1	9.47	19.02	12.50	19.95
	Average Improvements				0	5.37	21.73	17.97	

all gates.

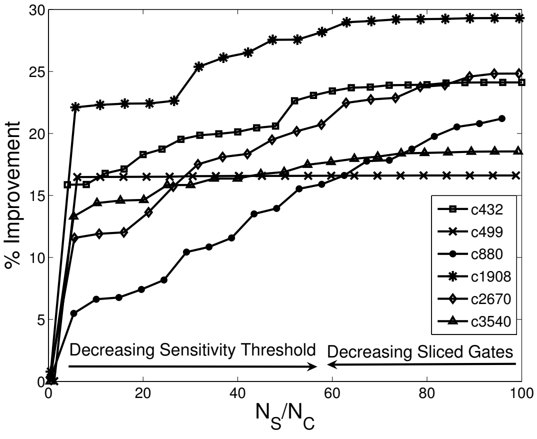

Figure 12: Average Improvements in Output delay variance on ISCAS benchmarks by TURGIS

5. CONCLUSION

In this work, a simple yet effective technique of gate-interconnect splitting was introduced. Interconnect splitting was shown to reduce the variation of the delay by 55%. Preliminary improvements of splitting were demonstrated on inverter gate and disadvantages of previous techniques were overcome. Average improvements of nearly up to 30% were achieved for ISCAS benchmarks and up to 35% were achieved for arbitrary sized Nand gate chains.

6. REFERENCES

[1] S. Nassif, “Delay variability: sources, impacts and trends,” in *IEEE International Conference on Solid - State Circuits*, 2000, pp. 368 – 369.

[2] S. R. Nassif, “Within chip variability analysis,” in *IEDM Technical Digest*, 1998, p. 283.

[3] S. Raj, S. Vrudhula, and J. Wang, “A methodology to improve timing yield in the presence of process variations,” in *DAC*, 2004, pp. 448–453.

[4] E. Jacobs and M. Berkelaar, “Gate sizing using a statistical delay model,” in *DATE*, 2000, pp. 283–291.

[5] S. H. Choi, B. C. Paul, and K. Roy, “Novel sizing algorithm for yield improvement under process variation in nanometer technology,” in *DAC*, 2004.

[6] J. Singh, V. Nookala, Z.-Q. Luo, and S. Sapatnekar, “Robust gate sizing by geometric prog.” in *DAC*, 2005.

[7] O. Neiroukh and X. Song, “Improving the process - variation tolerance of digital circuits using gate sizing and statistical techniques,” in *DATE*, 2005.

[8] D. Wu, G. Venkataraman, J. Hu, Q. Li, and R. Maphapatra, “Dicer: Distributed and cost-effective redundancy for variation tolerance,” in *International Conference on Computer Aided Design*, 2005.

[9] “National technology roadmap for semiconductors,” 2004.

[10] R. Chang, “Integrated cmp metrology and modeling with respect to circuit performance,” in *Ph.D Thesis, UC. Berkeley*, 2004.

[11] C. Clark, “The greatest of a finite set of random variables,” *Operations Research*, vol. 9, no. 2, 1961.

[12] R. B. Brawhear, N. Menezes, C. Oh, L. T. Pillage, and M. R. Mercer, “Predicting circuit performance using circuit-level statisticaltiming analysis,” in *European Design and Test Conference,*, 1994, pp. 332 – 337.

[13] S. Isukapalli and P. Georgopoulos, “Stochastic response surface methods (srsms) for uncertainty propagation : Application to enviromental and biological systems,” in *Risk Analysis*, 1998, pp. 351–363.

[14] A. Devgan and C. Kashyap, “Block - based static timing analysis with uncertainty,” in *ICCAD*, 2003, p. 607.

[15] “Berkeley ptm.” [Online]. Available: http://www-device.eecs.berkeley.edu/~ptm/