Journal of Circuits, Systems, and Computers
Vol. 20, No. 8 (2011) 1547–1569

DOI: 10.1142/S0218126611008043

World Scientific
www.worldscientific.com

SAMPLING CORRELATION SOURCES FOR TIMING YIELD ANALYSIS OF SEQUENTIAL CIRCUITS WITH CLOCK NETWORKS*

LEE-EUN YU
Department of Electrical Engineering, KAIST Daejeon 305-701, Republic of Korea

CHANGSIK SHIN
Silicon Works, Daejeon 305-380, Republic of Korea

SEUNGWHUN PAIK
Synopsys Inc., Mountain View CA 94043, USA

JING-JIA LIOU
Department of Electrical Engineering, National Tsing Hua University, Hsinchu 30013, Taiwan

YOUNGSOO SHIN
Department of Electrical Engineering, KAIST, Daejeon 305-701, Republic of Korea

Received 13 May 2011
Accepted 29 June 2011

Analyzing timing yield under process variations is difficult because of the presence of correlations. Reconvergent fan-out nodes (RFONs) within combinational subcircuits are a major source of topological correlation. We identify two more sources of topological correlation in clocked sequential circuit: sequential RFONs, which are nodes within a clock network where the clock paths to more than one flip-flop branch out; and sequential branch-points, which are nodes within a combinational block where combinational paths to more than one capturing flip-flop branch out. Dealing with all sources of correlation is unacceptably complicated, and we therefore show how to sample a handful of correlation sources without sacrificing significant accuracy in the yield. A further reduction in computation time can be obtained by sampling only those nodes that are likely to affect the yield. These techniques are applied to yield analysis using statistical static timing analysis based on discrete random variables and also to yield analysis based on Monte Carlo simulation; the accuracy and efficiency of both methods are

*This paper was recommended by Regional Editor Eby G. Friedman.

assessed using example circuits. The sequential RFONs suggest that timing yield may be improved by optimizing the clock network, and we address this possibility.

Keywords: Statistical static timing analysis; Monte Carlo simulation; timing yield; correlation; sequential circuit; clock network.

1. Introduction

Process variations are caused by imperfections in the mask and in the semiconductor manufacturing process. They can be classified into die-to-die (D2D) and within-die (WID) variations. The effect of D2D variation is uniform across all the circuit elements on the same die, and results from lot-to-lot, wafer-to-wafer, and a portion of within-wafer variations.[1] WID variation can affect different devices on the same die to a different extent because it is caused by the limitations of fabrication equipment and the statistical variability of the process.[2] D2D variation is commonly characterized by process corners, such as the worst corner (WC), nominal corner (NC), and best corner (BC). WID variation is more difficult to handle; it is either ignored, by assuming that all the devices are governed by the same process corner, or approximated by a linear combination of the process corners.[3] For instance, at WC, the late mode corresponds to WC itself, while the early mode is located at $0.5\text{NC} + 0.5\text{WC}$. The maximum delay of the combinational logic is calculated in late mode and the clock arrival time is calculated in early mode; the setup time can then be checked by comparing the two delays so as to approximate the WID variation.

These approaches become too pessimistic as the scale of processes shrinks so that WID variation represents an increasing proportion of the total variations: e.g., 35% in 130 nm technology but 60% in 70 nm technology.[4] To reduce the pessimism, each gate delay has to be modeled as a random variable. Modeling the gate delays as a linear combination of random variables[2,5–7] is more accurate because the random and systematic components of WID variation can be explicitly identified.

Timing analysis can then be performed using either Monte Carlo simulation or statistical static timing analysis (SSTA). Monte Carlo simulation relies on repeated random sampling of the value of each random gate delay followed by traditional static timing analysis (STA). SSTA[8] propagates random variables in the form of probability distribution functions (PDFs) or cumulative distribution functions (CDFs) that describe ATs and RATs. The distributions can be represented by continuous analytic functions or by a series of discrete values. Analytical approaches typically assume a normal distribution[5–7] for the sake of computational convenience, although this represents a limitation; approaches based on discrete random variables[9,10] do not assume any particular type of distribution, and are therefore more robust but can be expected to be slower.

1.1. *Related work*

Extensive research efforts have been made to compute the timing yield, which we will review in this section.

Timing yield analysis using SSTA based on discrete random variables has been proposed.[10] It, however, computes a timing yield in combinational circuits alone and ignores correlation sources that are caused by clock networks. An analytical SSTA has been used for yield analysis of sequential circuits.[11] This method considers spatial correlations but again ignores all the topological correlation sources.

Timing yield analysis using Monte Carlo simulation has been proposed.[12] Path filtering method is used to reduce the number of paths that deserve Monte Carlo simulation for yield analysis; this method still suffers from a large processing time which is inherent in Monte Carlo simulation. Another direction of improving efficiency of Monte Carlo-based yield analysis has been proposed.[13] The method uses quasi Monte Carlo (QMC), stratified sampling, and latin hypercube sampling (LHS) to reduce Monte Carlo sample numbers.

1.2. *Motivation*

Monte Carlo simulation or SSTA can be used to compute a timing yield which is the probability that a circuit satisfies given timing constraints. This is typically done by deriving a probability density function (PDF) for the latest arrival times at the outputs (the primary outputs and the flip-flop inputs) which is then compared with the max-time constraint that consists of RATs for the primary outputs and clock period for the flip-flops. Other timing constraints such as min-time and clock pulse width also have to be taken into account in computing the timing yield.

In both SSTA and Monte Carlo simulations, ATs are derived at all nodes of a circuit. Conversely, only a handful of nodes may be checked in a yield analysis since yield is typically affected by only a small number of critical paths.[14,15] But we need to ensure that we consider the nodes that have a significant effect on the yield, and it is this sampling problem that we address in this paper. An additional challenge in SSTA and yield analysis is to allow for topological correlations which introduce extra complexity. The challenge is again one of discrimination; we want to sample significant correlation sources accurately while ensuring that the correlation of the remainder can safely be ignored.

These problems are exacerbated in clocked sequential circuits because the number of correlation sources significantly increases if we have to consider a clock network along with a primary circuit. The correlation caused by a clock network (i.e., a common clock path) is typically ignored due to the excessive processing time required to deal with it but this introduces a significant error. Figure 1 shows two timing yield CDFs for `wb_dma`, which is an open-core benchmark circuit[16]: in the first CDF the clock network correlations are taken into account but in the second they are ignored. When the clock period is 960 ps, the error in the yield is 0.2 which is substantial: like other types of yield, the timing yield directly affects a manufacturer's profit. A lot of time and resource is often spent in achieving a small increase in timing yield but this

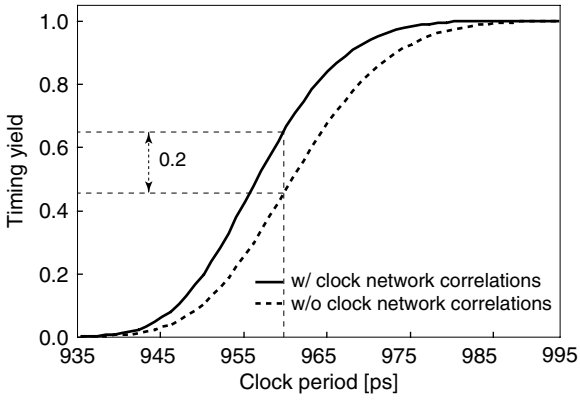


Fig. 1. Timing yield CDFs computed with and without considering clock network correlations in `wb_dma` circuit.

may be unnecessary if the timing yield is fully characterized by considering the clock network correlations as exemplified by Fig. 1.

The remainder of this paper is organized as follows: In Sec. 2, we briefly review approaches to SSTA that are based on discrete random variables, and the methods used to compute timing yield under max- and min-delay constraints; and we identify the three sources of topological correlation that are important for yield analysis. We then show how timing yield analysis based on SSTA is performed by sampling nodes and correlation sources in Sec. 3. The application of this approach to example circuits is reported in Sec. 4. We draw conclusions in Sec. 5.

2. Preliminaries

2.1. *SSTA based on discrete random variables*

The form of SSTA that we use to find a timing yield is based on discrete random variables. For a particular load capacitance and input transition time, the pin-to-pin delay of a gate is modeled as a discrete random variable, for which the probability of delay is specified by a discrete PDF; output transition times are modeled similarly. The AT and RAT of each node are also modeled as discrete random variables, and deriving their corresponding PDFs is a key element of timing analysis.

Let us consider the example in Fig. 2; note that we will overload our notation so that the same letter will refer to a node and to the arrival time at that node. Figure 2 shows how the arrival time c at the output of NAND gate is derived from the ATs a and b at its inputs along with the pin-to-pin delays d_1 and d_2. The PDF of $a + d_1$ is derived from the convolution of f_a and f_{d_1}, and c is the maximum of $a + d_1$ and $b + d_2$.

During SSTA, the correlation caused by a reconvergent fan-out is handled by a process called sampling-evaluation.[9] In Fig. 3, node a is called a reconvergent fan-out node (RFON) and we describe gate G_1 as the reconvergence gate for a. Suppose that

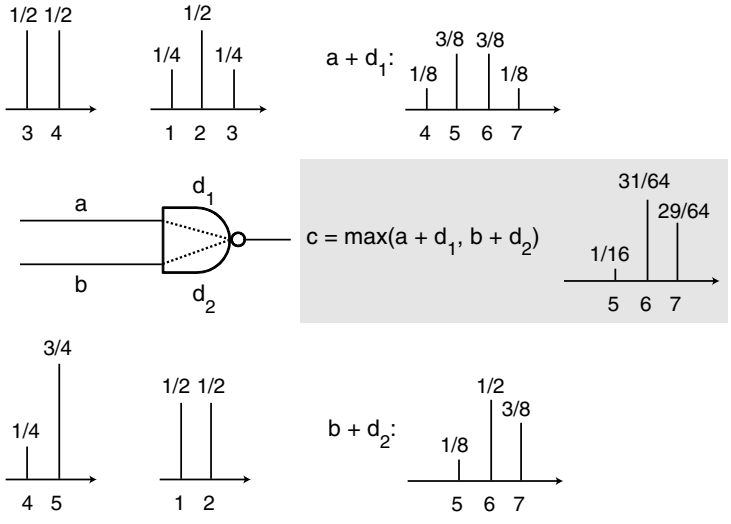


Fig. 2. ADD and MAX operations with discrete random variables.

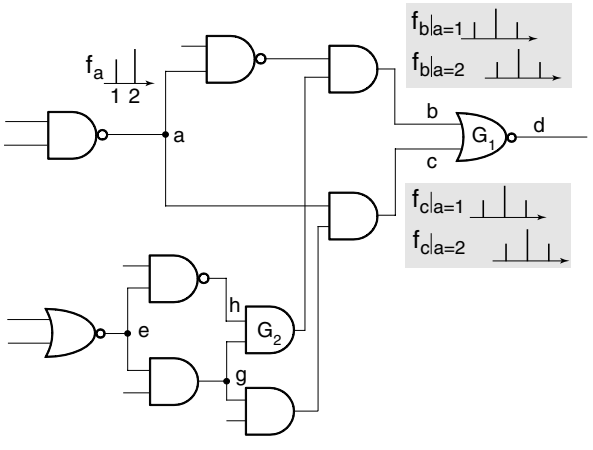


Fig. 3. Sampling-evaluation to handle correlation due to reconvergent fan-out.

a has two events in the PDF of its AT, occurring when $a = 1$ and $a = 2$. The first event causes a group of events at b, $f_b|_{a=1}$, and a group of events at c, $f_c|_{a=1}$; similarly, the second event causes $f_b|_{a=2}$ and $f_c|_{a=2}$. The PDF at b can be readily obtained from its two groups of events, by scaling the probability of the corresponding events at a, as follows:

$$f_b = f_a(a = 1) f_b|_{a=1} + f_a(a = 2) f_b|_{a=2} \,. \tag{1}$$

The distribution f_c can be obtained similarly. To derive f_d, the groups of events that occur at b and c which are caused by the same event at a have to be considered one by

one; we use $f_b|_{a=1}$ and $f_c|_{a=1}$ to construct $f_d|_{a=1}$ by propagating them through the gate G_1, and we propagate $f_b|_{a=2}$ and $f_c|_{a=2}$ to construct $f_d|_{a=2}$; we then combine $f_d|_{a=1}$ and $f_d|_{a=2}$ after they are scaled to get f_d using a similar equation to (1). Note that if we were simply to derive a single event group f_b (without explicitly deriving $f_b|_{a=1}$ and $f_b|_{a=2}$), together with f_c, and use them to derive f_d, we would implicitly be including the cases in which $f_b|_{a=1}$ and $f_c|_{a=2}$ (and $f_b|_{a=2}$ and $f_c|_{a=1}$) are propagated through G_1, which produces the wrong result.

The sampling-evaluation process is time-consuming because each event at an RFON has to be separately propagated to a reconvergence gate to construct its PDF. If more than one RFON converge to the same reconvergence gate, as a and e do in Fig. 3, all the combinations of events must be propagated. The same is true when RFONs are nested, for example, in the figure, g is within the fan-out cone of e. Heuristic approaches, such as ignoring events of low probability,[9] are therefore required to reduce the complexity of the sampling-evaluation process.

2.2. *Timing yield*

The timing yield of a sequential circuit is determined by max-delay (or setup-time) and min-delay (or hold-time) constraints. The yield under max-delay constraint can be expressed as follows:

$$\Pr\left[\max_{\forall i \rightsquigarrow j}(T_{\mathrm{cq},i} + D_{i,j} + T_{\mathrm{su},j} + S_i - S_j) \le P\right] = \Pr(\mathbf{S} \le P)\,, \tag{2}$$

where P is a clock period that is specified by the circuit designer, $i \rightsquigarrow j$ denotes the timing path from launching flip-flop i to capturing flip-flop j, $T_{\mathrm{cq},i}$ is the clock-to-Q delay of i, $D_{i,j}$ is the maximum delay of the combinational block between i and j, $T_{\mathrm{su},j}$ is the setup guard time of j, and S_i is the time at which the clock arrives at i. It should be noted that expression (2) is not restricted to timing paths between flip-flops; other types of timing path, such as those from the primary input (PI) to the primary output (PO), from the PI to a flip-flop, and from a flip-flop to the PO can also be modeled using a similar expression. For example, when i is the PI and j is the PO, then $T_{\mathrm{cq},i}, T_{\mathrm{su},j}, S_i$, and S_j become zero; P is replaced by $R_j - A_i$, where R_j is the RAT at j and A_i is the AT at i, both of which are specified by the designer as timing constraints.

The yield under min-delay constraint can be represented similarly:

$$\Pr\left[\min_{\forall i \rightsquigarrow j}(T_{\mathrm{cq},i} + d_{i,j} - T_{\mathrm{hd},j} + S_i - S_j) \ge 0\right] = \Pr(P - \mathbf{H} \le P)\,, \tag{3}$$

where $d_{i,j}$ is the minimum delay of the combinational block between i and j, and $T_{\mathrm{hd},j}$ is the hold time of j. Note that, in (2) and (3), all the parameters except P are random variables, and therefore they are associated with discrete PDFs; S_i and S_j are not independent if the paths of the clock signal to i and j share common elements in the clock network: in this case, correlations between two different timing paths

have to be taken into account in max and min operations which involve common gates.

Since expressions (2) and (3) are correlated due to the presence of common random variables and potentially common gates in the derivation of $D_{i,j}$ and $d_{i,j}$, both expressions have to be combined to obtain the exact timing yield:

$$\text{Yield} = \Pr[\max(\mathbf{S}, P - \mathbf{H}) \leq P]\,, \tag{4}$$

where

$$Y = \max(\mathbf{S}, P - \mathbf{H})\,. \tag{5}$$

Y is called a yield variable and we will refer to its PDF as a yield PDF. In Sec. 3, however, we will obtain (2) and (3) independently, and then multiply them together to approximate (4), for convenience of implementation. The correlation between $D_{i,j}$ and $d_{i,j}$, which are major components of $\mathbf{S}$ and $\mathbf{H}$ in (2) and (3), respectively, is reported to be small in practical circuits.[11] This observation, which is confirmed by the experiments reported in Sec. 4, motivates the method of approximation that we develop.

2.3. *Sources of topological correlation in a clocked sequential circuit*

In a clocked sequential circuit, there are three sources of topological correlation which are illustrated in Fig. 4.

- A combinational RFON (cRFON), which we introduced in Sec. 2.1, is an RFON within a combinational block. A cRFON affects the accuracy of $D_{i,j}$ in expression (2) and $d_{i,j}$ in (3): if sampling-evaluation is not performed at the cRFON, the values of $D_{i,j}$ and $d_{i,j}$ will be incorrect.
- A sequential RFON (sRFON) is a node within a clock network in which the clock paths to more than one flip-flop branch out, so that the clock path from a clock

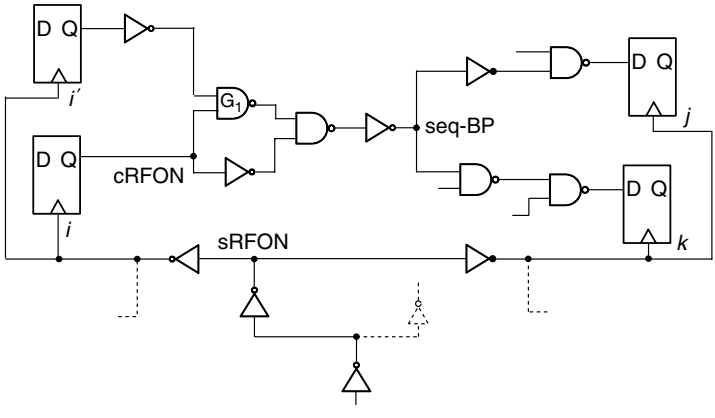


Fig. 4. Three sources of topological correlation in a clocked sequential circuit.

source to the sRFON is shared by multiple flip-flops. The sRFON that branches to launching flip-flop i and to capturing flip-flop j in Fig. 4 affects the accuracy of $S_i - S_j$ in expressions (2) and (3); if the correlation between S_i and S_j is not taken into account, the variance of the yield PDF will grow unnecessarily. When the sRFON branches to launching flip-flops i and i', as shown in Fig. 4, the correlation due to the sRFON has to be considered. When we compute the AT of the reconvergence gate G_1, this correlation affects the accuracy of $D_{i,j}$ and $d_{i,j}$ in (2) and (3), respectively.

- A sequential branch-point (seq-BP) is a node within a combinational block where combinational paths to more than one capturing flip-flop (or PO), say j and k, branch out, so that $i \rightsquigarrow j$ and $i \rightsquigarrow k$ share the same combinational path up to seq-BP, as shown in Fig. 4. This affects the accuracy of the max operation in (2) and the min operation in (3).

The effect of cRFONs on the accuracy of the PDF of the maximum arrival time at circuit outputs ($\mathbf{S}$ in (2)) is illustrated in Fig. 5(a). Note that we describe both primary outputs and capturing flip-flops as outputs, while primary inputs and launching flip-flops are called inputs. The figure compares one PDF obtained after applying sampling-evaluation at all correlation sources with another PDF obtained after applying sampling-evaluation at sRFONs and seq-BPs alone (thus ignoring the correlations due to cRFONs). Similar experiments were performed to assess the effect of sRFONs: their results are shown in Fig. 5(b), and the effect of seq-BPs is shown in Fig. 5(c). It can be readily seen that all the sources of correlation are important for the accuracy of the PDF of the maximum arrival time, and thus for the accuracy of the timing yield.

All three correlation sources could theoretically be handled by sampling-evaluation as explained in Sec. 2.1. However, the total number of cRFONs, sRFONs,

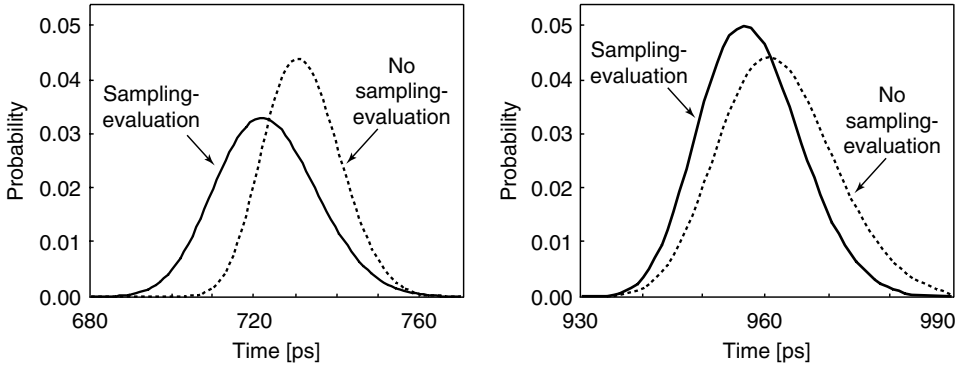
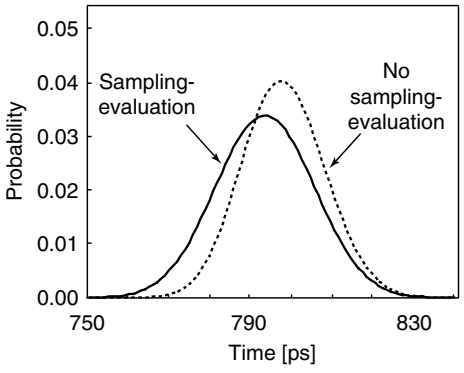


Fig. 5. Comparing PDFs of maximum arrival time at circuit outputs with and without applying sampling-evaluation: (a) at cRFONs in `s5378`, (b) at sRFONs in `wb_dma`, and (c) at seq-BPs in `s713`.



(c)

Fig. 5. (*Continued*)

and seq-BPs in practical circuits is simply too large, and each sampling-evaluation takes a long time. This makes direct application of sampling-evaluation to all correlation sources effectively impossible. Fortunately, it can be shown that many of these sources can be ignored without sacrificing too much accuracy. This leaves us with the task of searching for the minimum number of correlation sources that are important to the accuracy of the timing yield, and we will address this problem in the next section.

3. Timing Yield Analysis by Sampling Nodes and Correlation Sources

We derive a timing yield under max-delay and min-delay constraints ((2) and (3)) separately, and then multiply them together to obtain the overall timing yield as explained in Sec. 2.2. Since the process is similar for both timing constraints, we will only explain the process for the max-delay constraint.

3.1. *Structure of the algorithm*

Figure 6 illustrates the overall process of yield analysis, which consists of three steps: *sampling nodes*, *sampling correlation sources*, and *computing the yield*. The input to the algorithm is a netlist that describes a sequential circuit and its clock network.

The purpose of sampling nodes (L1 to L4) is to filter out the nodes that are unlikely to affect timing yield, so that the subsequent and expensive statistical analysis has to deal with as few nodes as possible. To this end, we first perform STA (L2) and compute the maximum arrival time at each output j, as follows:

$$m_j = \max_{\forall i: i \rightsquigarrow j} (T_{\mathrm{cq},i} + D_{i,j} + T_{\mathrm{su},j} + S_i - S_j)\,. \tag{6}$$

All the parameters in (6) have deterministic values based on the nominal gate delay. The maximum value of m_j, which correspond to the output associated with the

Algorithm Yield-analysis

Sample nodes:

L1 **for** each of WC, NC, and BC **do**
L2 Perform STA
L3 Mark candidate outputs and inputs for yield analysis
L4 $\mathcal{G} \leftarrow$ Mark nodes within a region spanned by candidate inputs and outputs

Sample correlation sources:

L5 Find all cRFONs in $\mathcal{G}$
L6 $\mathcal{C} \leftarrow$ Effective cRFONs
L7 Determine effective outputs from $\mathcal{G}$
L8 Determine effective inputs from $\mathcal{G}$
L9 $\mathcal{C} \leftarrow \mathcal{C} \cup$ effective sRFONs $\cup$ effective seq-BPs

Compute yield:

L10 Propagate ATs with sampling-evaluation at $\mathcal{C}$
L11 Compute timing yield

Fig. 6. Overall algorithm of yield analysis.

timing-critical path, is called m^*. We then run SSTA only for the gates on the timing-critical path to obtain the standard deviation of m^*, which we denote as σ^*. The output j that satisfies $m^* - m_j \leq \alpha\sigma^*$ is then considered to be a candidate for yield analysis, and is marked (L3); the constant α is chosen empirically, and this choice is discussed in Sec. 4. In the subsequent statistical analysis, m^* and m_j will become random variables, and will thus be associated with their own PDFs. If the PDF of m_j is sufficiently far to the left of the PDF of m^* (i.e., $m^* - m_j > \alpha\sigma^*$), the output j is unlikely to affect the timing yield, and thus the accuracy of m_j is not relevant to the analysis. During yield analysis under the min-delay constraint, m_j corresponds to the earliest arrival time (see (3)); m^* corresponds to the output with the minimum m_j; and output j is marked as a candidate if $m_j - m^* \leq \alpha\sigma^*$.

From the candidate outputs, we trace backwards to find the inputs that can be reached. Out of those inputs, we mark input i if its (deterministic) slack s_i satisfies $s_i \leq \alpha\sigma^*$ (L3), on the basis that an input is likely to affect the timing yield only if it has a very small slack. The process is repeated under different process corners (L1), e.g., worst corner (WC), nominal corner (NC), and best corner (BC), to account for D2D process variations.

All the internal nodes that lie in a region spanned by the candidate inputs and outputs are put into a list $\mathcal{G}$ for subsequent statistical analysis (L4), during which the nodes not in $\mathcal{G}$ are completely ignored. This is illustrated in Fig. 7. The AT random variable z at the output of the NAND gate is approximated as the sum of x and d_2 (the delay of the timing arc from x to z), and not as $\max(x + d_2, y + d_1)$ since y is not in $\mathcal{G}$.

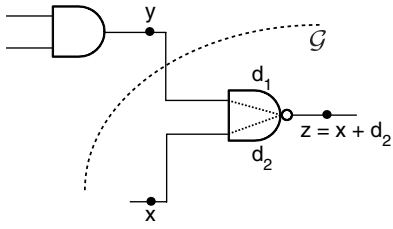


Fig. 7. Propagating AT random variables after sampling nodes.

In the second step (L5 to L9), we sample correlation sources from all the cRFONs, sRFONs, and seq-BPs in $\mathcal{G}$. The cRFONs that potentially contribute to the accuracy of the timing yield are called *effective* cRFONs (L6); finding them is addressed in Sec. 3.2. The effective sRFONs and effective seq-BPs are discovered indirectly, we identify effective outputs and inputs (L7 and L8) which are very likely to affect the timing yield; then the sRFONs and seq-BPs that are reachable from the effective outputs and inputs are labeled as effective (Sec. 3.3).

In the third step (L10 and L11), we propagate the ATs with corresponding PDFs through both the sequential circuit and its clock network, while processing the sampled effective correlation sources by sampling-evaluation (L10), so as to obtain the value of $\mathbf{S}$ from (2). This value is finally compared to P to determine the timing yield (L11).

3.2. *Effective cRFONs*

When multiple paths from a cRFON to its reconvergence gate are important in computing input ATs at the reconvergence gate, the correlation due to that cRFON is important in calculating delay PDF at the output of the reconvergence gate. Therefore, we mark the cRFON as effective, and effective cRFONs are processed by sampling-evaluation during the yield computation step (L10 in Fig. 6).

Consider cRFON a in Fig. 8. Let the pin-to-pin delays of the NAND gate be represented by the random variables x_1 and x_2, and the delays of the AND gate by y_1

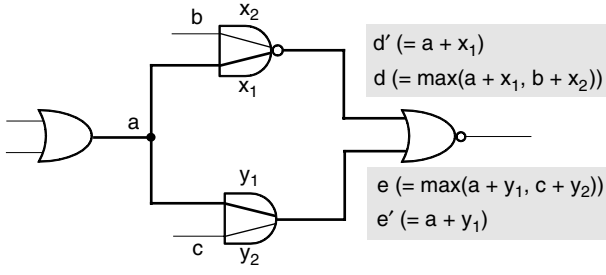


Fig. 8. Detecting effective cRFONs.

and y_2. The two inputs to the reconvergence gate are denoted by d and e; they are computed by max and add operations, as shown in Fig. 8; yielding their PDFs f_d and f_e. To determine whether cRFON a is effective or not, we introduce two additional random variables, d' and e'; their values are computed considering only the random variables along the paths from the cRFON to the inputs of the reconvergence gate. Thus $b + x_2$ is ignored when computing d', and $c + y_2$ does not contribute to e' (see Fig. 8). We declare a to be an effective cRFON if d and d' are statistically similar and e and e' are also statistically similar.

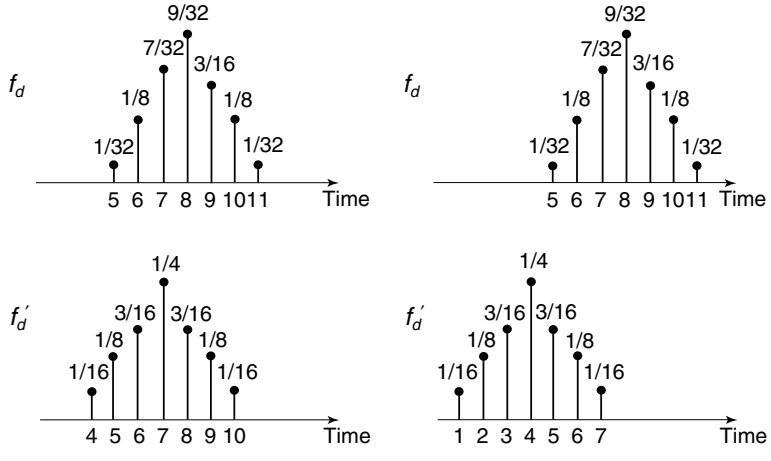
The similarity of two discrete random variables, d and d', for example, can be defined as the sum of the geometric average of the corresponding PDFs, evaluated at all the points at which they are defined[17]:

$$S(d, d') = \sum_i \sqrt{f_d(d = i) f_{d'}(d' = i)}\,. \tag{7}$$

Note that $0 \le S(d, d') \le 1$; $S(d, d')$ is 0 when there is no overlap between the two PDFs and 1 when they are exactly the same. Figure 9(a) shows an example in which two variables are quite similar, and Fig. 9(b) shows an example in which they are not. We can then apply a threshold ϵ_{cRFON} to decide whether two variables are similar:

$$d \text{ and } d' \text{ are similar} \quad \text{if } S(d, d') > \epsilon_{\text{cRFON}}\,. \tag{8}$$

Before we apply the process that we have just described, we need to determine the similarity between d and e. If $S(d, e)$ is less than a threshold ϵ'_{cRFON}, then a is not an effective cRFON because the AT of either d or e dominates the other; otherwise we can use the process described above to check whether a is effective. When the



(a) (b)

Fig. 9. Similarity between two random variables d and d': (a) $S(d, d') = 0.92$ and (b) $S(d, d') = 0.29$.

Algorithm Effective-cRFON (cRFON a)

L1 $\mathcal{I} \leftarrow$ list of inputs of reconvergence gate of a
L2 $j \leftarrow \text{argmax}_{d \in \mathcal{I}} \sum_i i f_d(d = i)$
L3 **for** each $d \in \mathcal{I} - \{j\}$ **do**
L4 **if** $S(j, d) > \epsilon'_{\text{cRFON}}$ **then** go to L6
L5 **return** a is not effective
L6 **for** each $d \in \mathcal{I}$ such that $S(j, d) > \epsilon'_{\text{cRFON}}$ **do**
L7 **if** $S(d, d') < \epsilon_{\text{cRFON}}$ **then** go to L5
L8 **return** a is effective

Fig. 10. Algorithm to detect an effective cRFON.

reconvergence gate has more than two inputs, we pick the input with the maximum mean AT, $\sum_i i f_d(d = i)$ where d is one of the inputs. Then we compute the similarity between d and all the other inputs; if all the similarity values are smaller than ϵ'_{cRFON}, the RFON is declared not to be effective. Figure 10 summarizes the procedure used to detect whether cRFON a is effective, in which we determine the similarity between the dominant input j, determined at L2, and all the other inputs to the reconvergence gate (L3 and L4); this is followed by the same test at each input (L6 and L7). Note that only the inputs d that are statistically similar to j, so that $S(j, d) > \epsilon'_{\text{cRFON}}$ are tested.

The thresholds ϵ_{cRFON} and ϵ'_{cRFON} are chosen empirically; their values reflect a trade-off between the accuracy of the timing yield analysis and runtime. This trade-off is explored in Sec. 4.2.1 to provide guidelines for choosing the empirical parameters. In our experiments with 45-nm technology, $\epsilon_{\text{cRFON}} = 0.7$ and $\epsilon'_{\text{cRFON}} = 0.5$ seemed reasonable choices in all the examples that we tried without sacrificing the accuracy of the timing yield analysis too much.

Example 1. Consider Fig. 3, which shows how we determine the reconvergence gates and their associated reconvergent nodes. G_1 is a reconvergence gate for a, e, and g; G_2 is a reconvergence gate for e. We propagate the ATs by computing the PDF of the AT at each node from the primary inputs toward the primary outputs. Once we reach G_2, we have to decide whether e is effective. We can then compute $S(g, h)$. Suppose that $S(g, h)$ is larger than ϵ'_{cRFON}; we then compute $S(g, g')$ and $S(h, h')$, and compare each of them to ϵ_{cRFON}; if both are larger than ϵ_{cRFON}, e is an effective cRFON. Once we reach G_1, we perform a similar procedure to decide which of a, e, and g are effective.

3.3. *Effective sRFONs and effective seq-BPs*

We can find out which sRFONs and seq-BPs require sampling-evaluation indirectly by identifying effective outputs (capturing flip-flops and POs) and effective inputs (launching flip-flops and PIs), which correspond to L7 and L8 of Fig. 6, respectively.

3.3.1. *Effective outputs*

While we are determining the effectiveness of the cRFONs, we propagate the ATs from the inputs toward the outputs, which are reached at the end of the process. This allows us to compute the latest arrival time at each output $j \in \mathcal{G}$:

$$m_j = \max_{\forall i: i \rightsquigarrow j} (T_{\mathrm{cq},i} + D_{i,j} + T_{\mathrm{su},j} + S_i - S_j)\,, \tag{9}$$

where $T_{\mathrm{cq},i} = S_i = 0$ if i is a PI and $T_{\mathrm{su},j} = S_j = 0$ if j is a PO. From all the outputs in $\mathcal{G}$, we select the one which is most likely to have a latest arrival time which violates its timing constraint, so that $\Pr(m_j > P)$ when both i and j are flip-flops. Let the latest arrival time of that output be denoted by m^*. We then derive the similarity $S(m_j, m^*)$ between m_j and m^* for each output j; if $S(m_j, m^*)$ is larger than the chosen threshold ϵ_{out}, we declare j to be effective.

If flip-flops i and j share a common clock path, the correlation between S_i and S_j is not taken into account by (9) because it is not yet known whether the sRFON which is the source of their correlation is effective. But our approach is conservative: if an output is effective when the correlations of the relevant sRFONs are taken into account, it is also effective when these correlations are ignored; although the opposite is not true. Figure 11(a) shows the PDFs of m_j and m^* when the correlations caused by the sRFONs are taken into account and Fig. 11(b) when they are not. Since variance of PDF increases while mean remains almost the same when correlations are not taken into account, $S(m_j, m^*)$ is larger in Fig. 11(b) than in Fig. 11(a).

3.3.2. *Effective inputs*

From the effective outputs, we trace back to the inputs; and those inputs that cannot be reached are declared ineffective. For each of the inputs i that can be reached, we compute its slack

$$s_i = r_i - a_i\,, \tag{10}$$

where a_i is a random variable of the AT, such that $a_i = S_i + T_{\mathrm{cq},i}$ if i is a flip-flop and $a_i = A_i$ (which is a designer-specified AT) if i is a PI. To derive the random variable

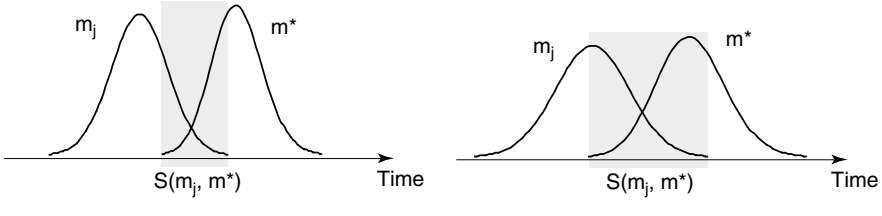


Fig. 11. Computing $S(m_j, m^*)$ (a) when correlations from sRFONs are taken into account and (b) when they are not.

corresponding to a RAT r_i, we first set the RAT of each effective output j to $r_j = \max_{\forall i: i \rightsquigarrow j}(T_{\mathrm{cq},i} + D_{i,j} + T_{\mathrm{su},j} + S_i)$, and propagate them toward the inputs, which eventually yields r_i. If it is sufficiently likely that the negative slack at the input i will be greater than the chosen threshold, so that $\Pr(s_i < 0) > \epsilon_{\mathrm{in}}$, then the input is declared effective.

3.3.3. *Effective sRFONs and effective seq-BPs*

Once the effective inputs and effective outputs have been determined, the effective sRFONs and effective seq-BPs are implicitly selected (see Fig. 4). Effective sRFONs are nodes within a clock network that branch out to multiple effective flip-flops. Effective seq-BPs are nodes within a combinational block that are located between effective inputs and effective outputs that branch out to multiple effective outputs.

3.4. *Speedup techniques for sampling-evaluation*

Even after reducing the number of correlation sources by sampling, the sampling-evaluation process dominates the total runtime required for yield analysis. This is especially true when more than one effective cRFONs is associated with the same reconvergence gate. In Fig. 3, for example, G_1 is a reconvergence gate for a, e, and g. If all three cRFONs are effective, N^6 event combinations have to be propagated to G_1, where each cRFON has two PDFs, each of which consists of N events, one for the rise arrival times and another for the fall AT. Therefore, we perform Monte Carlo simulation instead of sampling-evaluation if more than two effective cRFONs are associated with the same reconvergence gate in a region of a circuit spanned by the effective cRFONs and the reconvergence gate. We do not apply the same heuristic approach to sRFONs because the region of a circuit spanned by sRFONs and a reconvergence gate is, in general, too large for Monte Carlo simulation to be practicable.

4. Experimental Results

The timing yield analysis with sampling (TYAS) outlined in Fig. 6 was implemented in SIS.[18] We performed experiments on a set of sequential circuits taken from ISCAS and open cores.[16] They were synthesized with SIS and mapped into a 45-nm gate library, which we built with a predictive model.[19] For each circuit, a clock tree was constructed by grouping eight flip-flops together, and connecting each group to a leaf-stage clock buffer; clock buffers were then also grouped into eights and connected to the next stage of buffers; this was repeated until only one buffer remained, which was routed to a clock source. The clock period was set to the maximum flip-flop to flip-flop delay after performing STA with nominal gate delay. The arrival times of all the PIs were set to zero, and the required arrival times of all the POs were set to the clock period.

The threshold voltage (V_t) was chosen as a source of WID process variation. We assumed that the threshold voltage of an nMOS device at the nominal process corner would have a normal distribution with a mean (μ_w) of 200 mV and a standard deviation (σ_w) of 20 mV, with a V_{dd} of 1.0 V; a pMOS device is assumed to follow the same normal distribution with a mean of −200 mV. We also took $N(200, 20^2)$ to be the D2D process variation, which we will use in Sec. 4.3; this yields an overall standard deviation of $\sqrt{2}\sigma_w$, which is 28 mV. This is based on a 15% variation in the nominal V_t, as predicted from Ref. 20. To model the pin-to-pin delay of each gate, a SPICE simulation was performed for seven different values of V_t ($\mu_w - 3\sigma_w$, $\mu_w - 2\sigma_w$, $\mu_w - \sigma_w$, μ_w, $\mu_w + \sigma_w$, $\mu_w + 2\sigma_w$, and $\mu_w + 3\sigma_w$), yielding a discrete PDF consisting of seven events. Note that SIS approximates a gate delay as $D_i + Cd_i$, where D_i is an intrinsic delay, d_i is the increment of delay per unit capacitance, and C is the load capacitance. We derived a single PDF, which corresponds to D_i, while d_i remains constant. But many commercial STAs use a delay lookup table requiring an array of PDFs to be constructed and indexed by load capacitance and input transition time.

4.1. *Sampling nodes*

As described in Sec. 3.1, an initial netlist, along with timing information from the STA, is submitted to node sampling, which returns a set of candidate nodes for statistical analysis. This is done by marking every output j that satisfies $m^* - m_j \leq \alpha\sigma^*$ under the max-delay constraint (or $m_j - m^* \leq \alpha\sigma^*$ under the min-delay constraint), and every input i that can be reached from the marked outputs, which also satisfies $s_i \leq \alpha\sigma^*$; all nodes within a region spanned by the marked inputs and outputs are considered to be candidates. We used $\alpha = 2$, and this value was chosen empirically.

After sampling nodes, only 10% of gates are left for further analysis on average in the yield analysis under the max-delay constraint, and 3% in the yield analysis under the min-delay constraint. In yield analysis under the min-delay constraint, more outputs tend to be left out because there are, in general, more short paths whose delay is comparable to the shortest delay than long paths whose delay is comparable to the longest delay. In some exceptional circuits very few gates are left after node sampling, for example, less than 1% for circuit `s38584`. In this case, the maximum delay is dominated by one critical path, corresponding to a single output, with a delay of 1174 ps while the next critical path has 1122 ps of delay. As a result, only one output (out of 1647) and 14 inputs (out of 1458) remain after node sampling.

4.2. *Accuracy and runtime of TYAS*

4.2.1. *Effective correlation sources*

The number of effective cRFONs for yield analysis under the max-delay constraint is shown in the third column of Table 1, and the corresponding number for the

Table 1. Counts of total and effective cRFONs, sRFONs, and seq-BPs for computing the timing yield under max-delay (setup-time) and min-delay (hold-time) constraints.

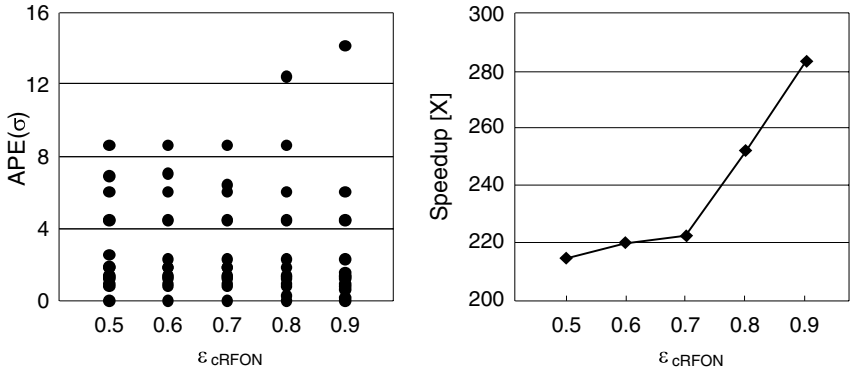
	# cRFONs			# sRFONs			# seq-BPs			# Effective Inputs/Outputs	
		Effective			Effective			Effective			
Name	Total	Setup	Hold	Total	Setup	Hold	Total	Setup	Hold	Setup	Hold
s349	33	3	0	3	1	5	30	0	1	2	12
s713	35	3	0	4	1	1	26	1	1	3	3
s1196	130	0	0	4	0	2	26	0	1	2	4
s1423	161	9	0	13	1	1	134	0	0	2	2
s1494	126	3	2	3	1	1	70	0	0	2	2
s5378	323	12	0	25	2	17	268	2	1	4	34
s15850	913	26	0	77	1	1	682	0	0	2	2
s38584	2485	0	0	205	2	2	2482	0	0	4	4
i2cm	75	3	0	8	2	4	49	0	0	3	8
ps2	478	21	0	28	1	2	252	0	0	3	4
wb_dma	270	4	0	30	5	1	328	1	0	7	2
usbc	310	5	0	59	1	2	477	1	0	3	4
t400	510	20	0	26	1	37	430	0	0	2	66
ac97	701	6	0	316	3	9	2430	6	0	12	18

min-delay constraint is shown in the next column: each of these is a fraction of the total number of cRFONs listed in the sixth column. The accuracy of the mean (μ) and standard deviation (σ) of the yield PDF obtained TYAS are assessed as absolute percentage error (APE). Thus

$$\mathrm{APE}(\mu) = \frac{|\mu_{\mathrm{mc}} - \mu|}{\mu_{\mathrm{mc}}} \times 100\%\,, \tag{11}$$

where μ_{mc} correspond to the mean of the yield PDF obtained by Monte Carlo simulation; and APE(σ) is defined accordingly. We explored the trade-off between APE(σ) and speedup (expressed as a factor of runtime of Monte Carlo simulation) by varying $\epsilon_{\mathrm{cRFON}}$, which is shown in Fig. 12. We did not plot APE(μ) because it was smaller than 0.3% even for $\epsilon_{\mathrm{cRFON}} = 0.9$. We used $\epsilon_{\mathrm{cRFON}} = 0.7$ and $\epsilon'_{\mathrm{cRFON}} = 0.5$ for all example circuits to maintain APE(σ) below 10% while improving the runtime.

The total number of effective inputs and outputs are shown in the last two columns of Table 1; again there is one for each constraint. Each number of effective inputs and outputs in Table 1 is also a fraction of the total number of inputs and outputs, which include the flip-flops listed in the third column and PIs/POs in the fifth column. The number of effective sRFONs and seq-BPs resulting from effective inputs and effective outputs are listed in columns 5−10. As we noted for node sampling in Sec. 4.1, more outputs are declared effective under the min-delay constraint creating a larger number of effective sRFONs. Table 1 shows clearly how the number of correlation sources is greatly reduced, which in turn reduces the complexity of sampling-evaluation. We used $\epsilon_{\mathrm{out}} = 0.8$ as the threshold to identify effective outputs, and $\epsilon_{\mathrm{in}} = 0.3$ to identify effective inputs. We performed Monte



(a) (b)

Fig. 12. (a) APE(σ) and (b) speedup of TYAS, which is expressed as a factor of runtime of Monte Carlo simulation, while varying ϵ_{cRFON}.

Carlo simulation on paths that are fan-in cones of the effective outputs and obtained its yield PDF while varying ϵ_{out}. Then ϵ_{out} was selected as the maximum value that maintains its APE(σ) below 10%. Similarly, to determine ϵ_{in}, we performed Monte Carlo simulation on paths between effective inputs and effective outputs.

4.2.2. *Comparison to Monte Carlo simulation*

We chose Monte Carlo simulation as a reference of comparison. It has been shown[10] that the required number of simulation runs n is determined by the expected confidence c, the expected accuracy a, and the target timing yield y, as follows:

$$n \geq \frac{1-y}{y} \frac{1}{(1-c)a^2} \,. \tag{12}$$

We chose $c = 0.95, a = 0.01$, and $y = 0.9$ which makes $n \equiv 20{,}000$ so that we need about 20,000 runs of Monte Carlo simulation to predict the target yield (0.9) with $\pm 1\%$ error and 95% confidence. The parameter with the greatest effect on n in the above inequality is a; our choice of $\pm 1\%$ error seems to be a reasonable compromise between accuracy and the number of simulation runs required.

The mean and standard deviation of the yield PDF (recall (5)) after Monte Carlo simulation are shown in the second and third columns of Table 2, and the simulation runtime is given in the following column. The corresponding figures for TYAS are listed in the next three columns. Note that, in TYAS, the PDFs of $\mathbf{S}$ and $P - \mathbf{H}$ are derived separately and are then multiplied to approximate the yield PDF, based on the assumption that $\mathbf{S}$ and $P - \mathbf{H}$ are not strongly dependent, which is confirmed by the very small values of APE(μ) and APE(σ). The seventh column gives the factors expressing the reduction in runtimes, and the average factor is 222. There is, however, some potential to reduce the runtime of Monte Carlo simulation. The number of

Table 2. Comparison of the yield PDF and runtime for Monte Carlo simulation and TYAS. The last two columns correspond to the yield PDF obtained by SSTA while ignoring all sources of correlation.

	Monte Carlo			TYAS			No correlations	
Name	Mean (ps)	SD (ps)	Runtime (s)	APE (μ)	APE (σ)	Speedup ($\times$)	APE (μ)	APE (σ)
s349	621.4	9.2	30	0.0	1.9	6	0.5	3.8
s713	838.8	11.1	49	0.0	1.4	44	0.3	11.1
s1196	705.7	12.6	131	0.0	0.0	196	0.0	0.0
s1423	2172.3	17.8	145	0.0	4.5	241	0.2	8.1
s1494	646.3	11.4	175	0.1	6.1	500	0.1	13.2
s5378	725.2	12.2	392	0.1	8.6	44	1.7	28.5
s15850	1357.3	11.2	1240	0.0	6.4	26	1.0	25.3
s38584	1198.0	15.2	3631	0.1	1.3	1435	0.1	1.3
i2cm	891.4	9.3	85	0.0	1.2	251	0.1	6.0
ps2	988.5	10.6	593	0.0	1.9	17	0.6	10.2
wb_dma	961.1	8.8	487	0.1	1.0	31	0.0	0.2
usbc	712.3	12.0	628	0.1	2.3	237	1.6	21.4
t400	1200.7	12.4	722	0.0	0.8	12	0.0	0.9
ac97	816.1	18.4	4277	0.0	4.5	67	0.6	18.3
Average				0.0	3.0	222	0.5	10.6

runs, which is derived from inequality (12), may be excessive because it is based on Chebyshev's inequality which applies to any type of distribution but is not specific to a binomial distribution in yield analysis[10]; however, no tighter bound on the number of runs is available.

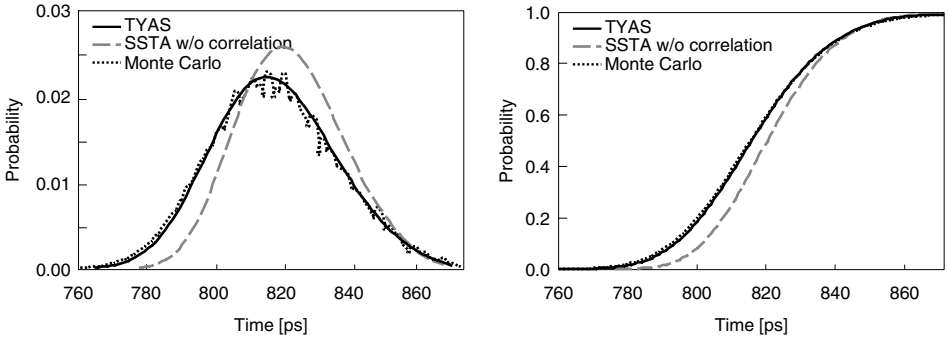
The last two columns of Table 2 characterize the yield PDFs obtained by SSTA when all the sources of correlation are ignored. The APEs deteriorate as expected. In particular, the APE of σ increases substantially, which implies that unnecessarily small values of yield will be reported for a given clock period, which increase the effort required to improve the yield.

We used `ac97`, which is one of circuits from open core, to compare the three methods to analyze the yield from Table 2; the yield PDF in Fig. 13(a) and the yield CDF in Fig. 13(b). The difference between TYAS and Monte Carlo simulation is very small. Ignoring the sources of correlation shifts the PDF and CDF to the right; and then the yield for a given clock period is unnecessarily reduced.

4.3. *Yield analysis under D2D and WID process variations*

4.3.1. *Analysis method*

The timing yield analysis in Sec. 3 only considers WID process variation, and thus a particular instance of D2D variation (i.e., a particular process corner). However, it can readily be extended to deal with both D2D and WID variations. Let D2D variation be denoted by a discrete random variable v with an associated PDF denoted by f_v. Recalling (5), let the yield variable corresponding to the ith value of v be denoted by Y_i. A yield variable which takes account for both D2D and WID



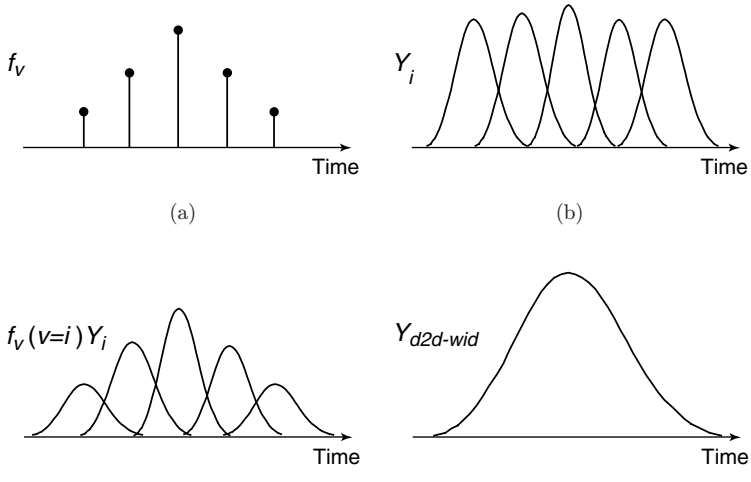
(a) (b)

Fig. 13. Comparison of (a) yield PDFs and (b) yield CDFs, for circuit `ac97`, by Monte Carlo simulation, TYAS, and SSTA that ignores correlation sources.

variations can then be formulated as follows:

$$Y_{\text{d2d}-\text{wid}} = \sum_i f_v(v = i) Y_i \,. \tag{13}$$

Figure 14 illustrates the process. Timing yield analysis under WID variation is performed five times at each instance of D2D variation, as shown in Fig. 14(b); each yield variable is then scaled by the probability corresponding to each D2D variation, as shown in Fig. 14(c). Summing all the scaled yield variables produces the aggregate yield variable shown in Fig. 14(d).



(a) (b) (c) (d)

Fig. 14. (a) D2D process variation modeled by v and its PDF f_v, (b) the yield variable Y_i for different instances of D2D variation, (c) Y_i scaled by $f_v(v = i)$, and (d) the aggregate yield variable $Y_{\text{d2d}-\text{wid}}$.

4.3.2. *Experiments*

The threshold voltage (V_t) was again chosen as a source of D2D process variation. It follows a normal distribution with a mean (μ_d) of 200 mV for an nMOS device and −200 mV for a pMOS device, and a standard deviation (σ_d) of 20 mV in both cases. The variable v can take seven different values: $\mu_d - 3\sigma_d, \mu_d - 2\sigma_d, \mu_d - \sigma_d, \mu_d$, $\mu_d + \sigma_d$, $\mu_d + 2\sigma_d$, and $\mu_d + 3\sigma_d$.

The results shown in Table 3 again compare TYAS to Monte Carlo simulation. The probability of the nominal corner, i.e., the probability that v takes a median value, is typically much larger than the probabilities associated with faster or slower corners; the probability of NC is 0.4 while that of the fastest or slowest corner is 0.004 in this experiment. Therefore, the yield analysis in process corners other than the NC does not need to be as accurate as in the NC. The runtime required for yield analysis is highly sensitive to the number of effective outputs (see the last two columns of Table 1), which is determined by ϵ_{out} (see Sec. 3.3.1). Therefore, when $v = i$, we adjust the value of ϵ_{out} as follows:

$$\epsilon_{\text{out},i} = \epsilon_{\text{out}} + (1 - \epsilon_{\text{out}})\left(1 - \frac{f_v(v = i)}{f_v(v = v_{\text{nom}})}\right), \tag{14}$$

where v_{nom} is the value of v at the NC. Note that $\epsilon_{\text{out},i} = \epsilon_{\text{out}}$ when $i = v_{\text{nom}}$; and $\epsilon_{\text{out},i}$ approaches 1 as $f_v(v = i)$ becomes smaller, which reduces to the number of outputs designated as effective, thus reducing the runtime required for yield analysis.

This heuristic is responsible for the speedup listed in the last column of Table 3 being larger than those achieved by yield analysis with WID variation alone (column 7 of Table 2). With circuit `s1196` the speedup increases from 196 to 491, with `s5378`

Table 3. Comparison of yield PDF and runtime between Monte Carlo simulation and TYAS under D2D and WID variations.

	Monte Carlo			TYAS		
Name	Mean [ps]	SD [ps]	Runtime [s]	APE(μ)	APE(σ)	Speedup [×]
s349	622.9	50.3	210	0.1	0.6	4
s713	840.4	70.0	352	0.1	1.5	53
s1196	706.8	58.7	947	0.0	1.2	491
s1423	2177.7	176.5	1036	0.1	2.7	182
s1494	647.9	53.7	1258	0.1	1.4	371
s5378	726.9	56.7	2810	0.3	2.3	180
s15850	1360.4	104.7	8599	0.0	0.9	25
s38584	1201.2	102.6	25033	0.2	2.5	1121
i2cm	894.2	76.1	603	0.0	1.0	248
ps2	990.7	79.9	4165	0.1	0.6	25
wb_dma	963.6	80.6	3402	0.1	0.8	116
usbc	714.2	57.5	4403	0.0	1.8	207
t400	1203.4	97.3	5013	0.1	1.1	9
ac97	817.8	57.3	22222	0.1	2.7	101
Average				0.1	1.5	246

it is 180 instead of 44, with `wb_dma` it is 116 instead of 31, and with `ac97` it is 101 instead of 67; all these circuits benefit from fewer effective outputs in the slower and faster corners as $\epsilon_{\text{out},i}$ is increased by (14). However, the speedup is decreased in some circuits, in particular, the result for `s1494` changes from 500 to 371, and that for `s38584` changes from 1435 to 1121; the number of effective outputs does not decrease in these circuits as $\epsilon_{\text{out},i}$ increases, while the number of effective inputs does increase as it works out, which in turn increases the number of sRFONs. A similar heuristic could be applied to Monte Carlo simulation, which would allow us to reduce the number of runs in slower or faster corners; we did not explore this direction, and the appropriate number of runs in those process corners remains to be investigated.

5. Conclusion

We have demonstrated that sRFONs and seq-BPs, in addition to cRFONs, are important in the accurate determination of timing yield for sequential circuits with clock networks. In a practical circuit, there are too many correlation sources to handle efficiently, and we have therefore presented a heuristic method of sampling only the effective sources that are likely to affect yield. We have also proposed a heuristic method of sampling a subset of nodes deserving statistical analysis to reduce the computation time. The method of sampling that we propose has been applied to yield analysis using SSTA based on discrete random variables, and also to yield analysis using Monte Carlo simulation.

References

1. K. Bowman, S. Duvall and J. Meindl, Impact of die-to-die and within-die parameter fluctuations on the maximum clock frequency distribution for gigascale integration, *IEEE J. Solid-State Circuits* **37** (2002) 183–190.
2. A. Agarwal, D. Blaauw and V. Zolotov, Statistical timing analysis for intra-die process variations with spatial correlations, *Proc. Int. Conf. Computer Aided Design* (2003), pp. 900–907.
3. M. Weber, My head hurts, my timing stinks, and I don't love on-chip variation, *Proc. Synopsys User Group Meeting* (2002), Boston.
4. P. Zuchowski *et al.*, Process and environmental variation impacts on ASIC timing, *Proc. Int. Conf. Computer Aided Design* (2004), pp. 336–342.
5. H. Chang and S. Sapatnekar, Statistical timing analysis considering spatial correlations using a single PERT-like traversal, *Proc. Int. Conf. Computer Aided Design* (2003), pp. 621–625.
6. C. Visweswariah *et al.*, First-order incremental block-based statistical timing analysis, *Proc. Design Automation Conf.* (2004), pp. 331–336.
7. L. Zhang *et al.*, Statistical timing analysis with extended pseudo-canonical timing model, *Proc. Design, Automation and Test in Europe Conf. and Exhibition* (2005), pp. 952–957.
8. D. Blaauw *et al.*, Statistical timing analysis: From basic principles to state of the art, *IEEE Trans. Comput.-Aided Des.* **27** (2008) 589–607.

9. J. Liou *et al.*, Fast statistical timing analysis by probabilistic event propagation, *Proc. Design Automation Conf.* (2001), pp. 661−666.
10. S. Naidu, Timing yield calculation using an impulse-train approach, *Proc. Asia South Pacific Design Automation Conf.* (2002), pp. 219−224.
11. M. Pan, C. Chu and H. Zhou, Timing yield estimation using statistical static timing analysis, *Proc. Int. Symp. Circuits and Systems* (2005), pp. 2461−2464.
12. S. Naidu, Speeding up Monte Carlo simulation for statistical timing analysis of digital integrated circuits, *Proc. Int. Conf. VLSI Design* (2007), pp. 265−270.
13. V. Veetil, D. Sylvester and D. Blaauw, Efficient Monte Carlo based incremental statistical timing analysis, *Proc. Design Automation Conf.* (2008), pp. 676−681.
14. L. Scheffer, The count of Monte Carlo, *ACM/IEEE Int. Workshop on Timing Issues in the Specification and Synthesis of Digital Systems*, February 2004.
15. S. Raj, S. Vrudhula and J. Wang, A methodology to improve timing yield in the presence of process variations, *Proc. Design Automation Conf.* (2004), pp. 448−453.
16. Opencores, http://www.opencores.org/.
17. A. Bhattacharyya, On a measure of divergence between two statistical populations defined by their probability distributions, *Bull. Calcutta Math. Soc.* **35** (1943) 99−109.
18. E. Sentovich *et al.*, SIS: A system for sequential circuit synthesis, UCB/ERL M92/41, U. C. Berkeley, Technical Report (1992).
19. W. Zhao and Y. Cao, New generation of predictive technology model for sub-45nm design exploration, *Proc. Int. Symp. Quality Electronic Design* (2006), pp. 585−590.
20. C. Chiang and J. Kawa, *Design for Manufacturability and Yield for Nano-Scale CMOS* (Springer, New York, 2007).