Approximating Standard Cell Delay Distributions by Reformulating the Most Probable Failure Point

Dimitrios Rodopoulos*,†, Philippe Roussel‡, Francky Catthoor†,‡, Yannakis Sazeides^, and Dimitrios Soudris*
*MicroLab-ECE-NTUA, Greece, †ESAT–KU Leuven, Belgium, ‡imec, Belgium, ^Ξ Group-UCY, Cyprus
Contact Email: drodo@microlab.ntua.gr

***Abstract*—The delay distribution of a digital circuit path is crucial for the early reliability evaluation of a digital design. As transistors are shrunk to unprecedented dimensions, accurate yet fast estimation of such distributions remains a valid goal. Such distributions may not be provided or are delivered in a heavily abstracted fashion to designers, which reduces the insight into design dependability. In view of the above observations, we propose a technique that approximates the probability density function of a path of digital circuits by exending a well-known computational kernel, namely the Most Probable Failure Point (MPFP) technique. The output of this concept is the failure probability of standard cells or paths thereof for various target delays. We reformulate MPFP and establish a concise methodology for delay distribution approximation. We present simulations for an inverter and outline projections for more complex gates.**

I. Introduction

The inherent time-zero and time-dependent variability of semiconductor structures creates challenges for the efficient/accurate modeling of integrated circuit reliability [1]. Statistical Static Timing Analysis (SSTA) has been prevalent in handling delay distributions of standard cells. This is split mainly between depth-first (or path based) [3], [4] and breadth-first (or block based) [2], [5], [6] techniques.

Regardless of SSTA techniques, what is really interesting is the derivation of the primitive delay distributions of standard cells. Accurate derivation requires a large number of simulations or measurements, since rare delay events need to be accounted for. In many cases, complete distributions are not even available and only an approximated view of standard cell delay variability is provided, as in the case of the stage-based on-chip variation (OCV) [7]. In view of the above, it is important to provide an accurate and efficient technique for the approximation of a standard cell's delay distribution.

The current paper delivers the distribution of an inverter delay, based on the distributions of the involved threshold voltages (V_{th}). The numerical kernel used is an extension of the Most Probable Failure Point (MPFP), which has been used for memory cells [8], [9], [10]. We reformulate it using the χ^2 distribution and use it iteratively to get the probability mass for various inverter target delays. Project and extensions are provided for more complex standard cells and paths.

Section II presents general formulation and aspects of prior art. Simulation results are analytically presented for the case of a simple inverter in Section III. In Sections IV and V, we outline the extensions of our the scheme to more complex gates and paths. Conclusions are summarized in Section VI.

II. General Formulation & Prior Art

The manifestation of variability in a circuit can be encapsulated in a vector $\mathbf{x}$ (e.g. threshold voltage shift per involved transistor). A performance metric (e.g. delay of cell) y can be evaluated at each $\mathbf{x}$ point as $y(\mathbf{x})$. A failure occurs when the performance metric is larger than a specified target (Y), namely $y > Y$. For a specific Y, the MPFP methodology aims to find the failure, i.e. $P_{\text{fail}}(Y) = P(y > Y)$. In order to connect variability with the failure specification, it is important to isolate all the values of $\mathbf{x}$ that satisfy the failure criterion. If we use F to represent the set of x values that lead to a failurem then the failure probability is $P(\mathbf{x} \in F)$. The challenge posed is *both* isolating F *and* calculating $P(\mathbf{x} \in F)$ in a systematic way. We illustrate this situation with an inverter, which has been profiled using Synopsys NanoTime [11] according to Figure 1a. Simulation results have been fitted with MATLAB for simplicity, as shown in Figure 1b, where a level set is annotated for a specific Y (roughly 17.5 ps). For all the simulations presented herein, we have used a publicly available high performance 16 nm modelcard [12].

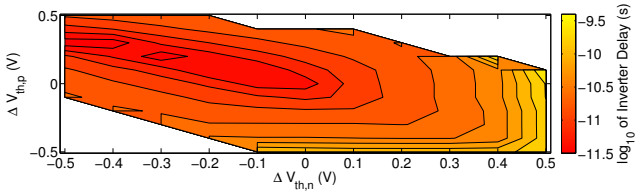
According to MPFP, the functionality criterion (which in our case is $y > Y$) is combined with a product of probabilities to approximate space F [8], [9], [10] and its probability mass, according to Equation 1 has been used to approximate F. If we cast the product of probabilities to the delay space (Figure 1c), it is clear that *even if the functionality criterion is correct, the probability mass of set F is not totally covered*. To address the above inaccuracy, we propose $P_{\text{fail}}(Y)$ calculation in two separate steps, described in Subsections III-A and III-B. There exactly lies the novelty of the current paper, in replacing the traditional MPFP formulation with the χ^2 distribution.

$$P_{\text{fail}} = \mathbf{max} \left\{ \prod_{i=0}^{N-1} P\left(|\Delta V_{th,i}| \geq x_i\right) \right\} \text{ such that } y(\mathbf{x}) > Y \tag{1}$$

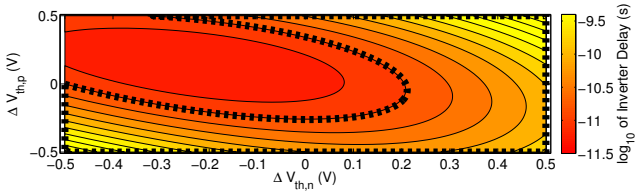
III. Adapting the MPFP – Inverter Focus

A. Minimum Identification

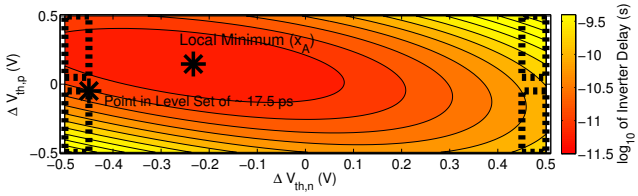
We isolate the point $\mathbf{x}_A$ which leads to minimum delay y_{min}. To achieve this, we solve the optimization problem $\mathbf{min}\{y(\mathbf{x})\}$ for $\mathbf{x}$ and also get the distance of $\mathbf{x}_A$, which we notate as r_A. For the case of the inverter, it is reasonable to expect a unique solution to this optimization problem. This statement has been verified with a series of Synopsys



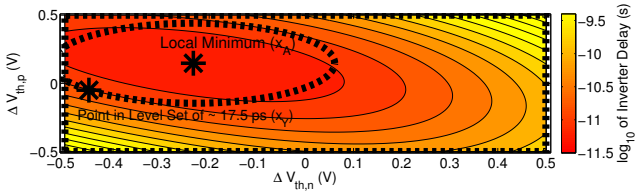
(a) Inverter delay as measured by NanoTime



(b) Fitted NanoTime measurements and bounding F with level set of 20 ps



(c) F approximation by $\prod_{i=0}^{N-1} P\left(|\Delta V_{th,i}| \geq x_i\right)$ as used in Equation 1



(d) F approximation using a hypersphere around the local minimum of inverter delay; additional probability mass is included and the χ_2 distribution applies

Fig. 1. Navigating the $\mathbf{x}$ space for accurate approximation of the inverter delay distribution. The χ^2 distribution is useful in bounding the failure region (F).

NanoTime [11] simulations (Figure 1a) and remains valid when we use a fitted expression for $y(\mathbf{x})$ as in Figure 1b.

B. Moving away from the Minimum

The procedure followed in this step is outlined in Figure 1d. First, we isolate a direction away from $\mathbf{x}_A$ along which the increase of delay is maximum. Having selected a target delay Y, we move along the above direction, until we reach $\mathbf{x}_Y$, where $y(\mathbf{x}) = Y$ at a distance equal to r_Y *from the point* x_A. At this point, the *generalized non-central Chi distribution* can be used (χ^2). This provides the probability mass of a hypersphere in the threshold voltage shift space, which is centered at $\mathbf{x}_A$ and has a radius equal to r_Y. By comparing Figures 1b and 1d, it is clear that the use of the χ^2 distribution is rather pessimistic, since $\mathbf{x}$ points that are faster than the target Y are included in space F. However, this is preferable than the formulation of Equation 1, which is highly optimistic, since it ignores a huge portion of probability mass, as we can verify from Figure 1c. On the contrary, *the* χ^2*-based formulation is guaranteed to contain all the failure points*, provided that point $\mathbf{x}_Y$ is selected based on a greatest ascent, moving away from $\mathbf{x}_A$. This ensures that the "pass" region only contains "pass" points, even though some are excluded (pessimism). It is important to note that the above statement is correct *regardless of the shape of* $y(\mathbf{x})$, as long as no other minima exist beyond the level set.

The goal is to calculate the probability mass of random variable $\mathbf{z}^2$, as defined in Equation 2, where N is the number of involved transistors (2 in the case of the inverter) and σ_i is the standard deviation of the ΔV_{th} per involved transistor. In the current paper, we assume that all transistors exhibit the same σ_i. The *non-centrality parameter* of the utilized distribution is calculated according to Equation 2 [13]. Apart from the involved σ_i, this parameter considers the distance between point $\mathbf{x}_A$ and the origin of the axis of the $\mathbf{x}$ space, which is encapsulated as the translated mean V_{th} shift per transistor (i.e. μ_i). For the rest of current paper, we assume that these mean values are constant. In case transistor aging is assumed, actual mean V_{th} shifts become non-zero [14] and distance to $\mathbf{x}_A$ can be easily recalculated.

$$\mathbf{z^2} = \sum_{i=0}^{N-1} \frac{x_i^2}{\sigma_i^2} \text{ and } \lambda = \sum_{i=0}^{N-1} \frac{\mu_i^2}{\sigma_i^2} \tag{2}$$

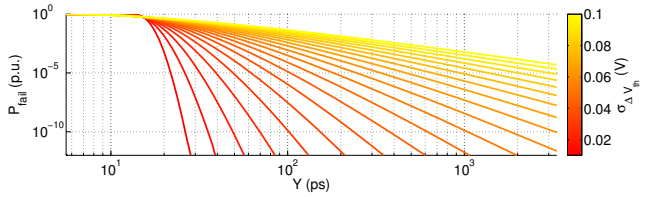
At this point, we note that the approximation of the mutli-variate distribution for the $\mathbf{x}$ vector can be improved by utilizing distribution transformations, as advised in prior art [15]. In the current paper and for the sake of simplicity, we assume no correlations between involved x_i's, hence we solely rely on the χ^2 distribution. With the above approach, we end up with the P_{fail} results of Figure 2a. As expected, a higher σ for ΔV_{th} leads to higher P_{fail} for the same target Y. Finally it is clear that, in case $f(\mathbf{x})$ has a single maximum (instead of minimum), we can alternatively start from its maximum and directly bound set F, instead of its complement. Choice between minimum/maximum depends on the shape of $f(\mathbf{x})$.

C. Getting Delay Distribution Points

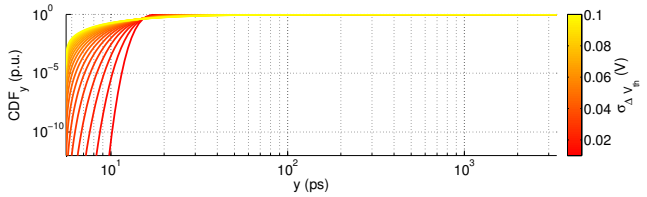
The failure probability for a target Y satisfies Equation 3, where PDF_y and CDF_y are the probability and cumulative density functions for y (inverter delay in our case).

$$P_{\text{fail}} = P(y > Y) = 1 - \text{CDF}_y(Y) = 1 - \int_{-\infty}^{Y} \text{PDF}_y dy \tag{3}$$

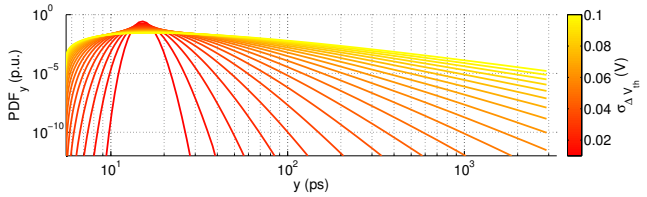
It is clear that by repeating the process of Subsection III-B for different values of Y we can isolate the cumulative probability for various delay specifications of the target circuit. Based on the P_{fail} vs. Y relation derived in Figure 2a, we easily produce the respective CDF_y data, as illustrated in Figure 2b. A simple differentiation yields the corresponding PDF_y. This effectively constitutes the *delay distribution of the inverter*. It is solely based on a NanoTime-compatible description of the inverter and uses values of the standard deviation for V_{th} shifts (multiple values inspected). This being a non-analytical



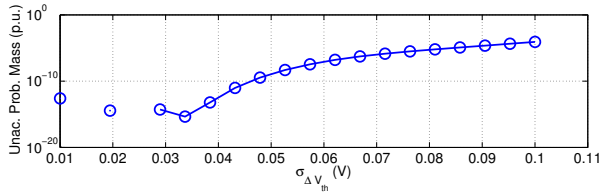
(a) Inverter failure probability for various values of V_{th} shift spread



(b) Cumulative density function of inverter delay for various values of σ



(c) Probability density function of inverter delay for various values of σ



(d) Unaccounted probability mass using our approximation

Fig. 2. Inverter delay analysis and approximation of set F for P_{fail} extraction

approach, it is important to highlight the probability mass that is unaccounted for. This corresponds to events that are not included in the presented PDF$_y$, as in the case of very high target delays (which can be safely assumed as negligibly rare). The unaccounted probability mass is illustrated in Figure 2d. Evidently, *the higher the spread of V_{th} shifts is, the higher the unaccounted probability mass is using our technique. However, when considering σ values for ΔV_{th} that are relevant to current technologies [14], our technique behaves with sufficient accuracy*. The delay of the inverter is lower bounded (Figure 1b). This means that delay distribution of the inverter does not have a tail on the left. However, as σ increases, the right-hand tail extends uncontrollably.

IV. Generalizing to Complex Gates

Standard cells with more than one transistor in the pull-up/-down branches, require a systematic way of identifying the delay minimum (of maximum). In the general case, one has no information about $y(\mathbf{x})$, which is evaluated with NanoTime for each iteration. We implement coordinate descent [16] according to Algorithm 1 for the case of a NAND gate.

Algorithm 1: Coordinate descent used in the current paper, based on iteration limit and V_{th} step equal to s

```
while (itNum<Limit) {
  for i{∈ 0,1,...,N−1} {
    s =find descenting direction for x_i
    if y(x_0,...,x_i+s,...,x_N−1) <prev_delay
      {Update x with x_i+s}
    else {Proceed to next transistor}
  }
}
```

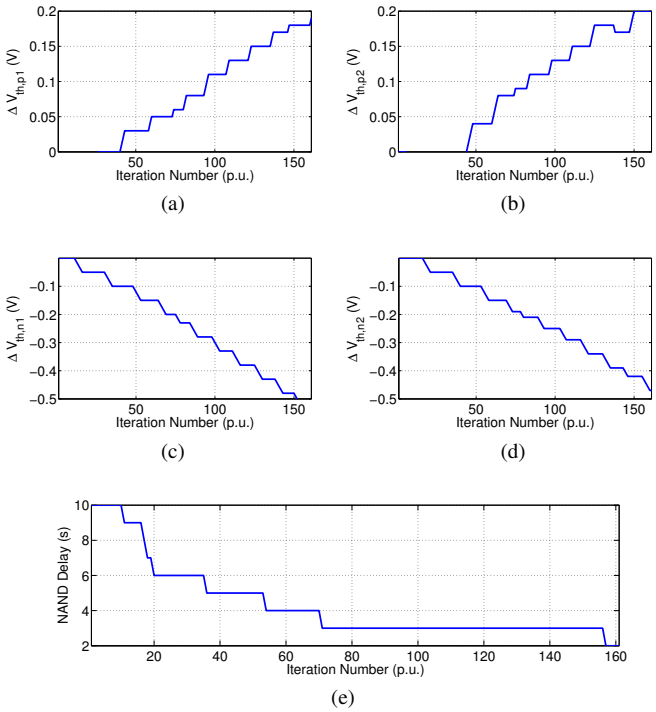


Fig. 3. Results of the coordinate descent implementation of Algorithm 1. One of the minimum delay points is identified within $\sim$ 160 NanoTime iterations.

Algorithm 1 has been implemented as Perl wrapper around NanoTime. We initially sweep the $y(\mathbf{x})$ function for the NAND case, in a 100 mV granularity ($> 14,000$ NanoTime iterations). This is a crude estimation of the minimum value and the value of the corresponding V_{th} shifts (i.e. estimation of $\mathbf{x}_A$. Using Algorithm 1, we succeed in identifying the minimum point in $\sim$ 160 NanoTime iterations (Figure 3).

Given the multitude of transistors in the pull-up/-down branches of standard cells, *multiple minima/maxima may exist* for $y(\mathbf{x})$ ($i = 0, 1, ..., M-1$). Given the symmetry of such cells, we may treat only one of these points ($\mathbf{x}_{Ai}$) with the technique of Subsection III-B. The cumulative probability around $\mathbf{x}_{Ai}$ can be multiplied by M to provide the total probability mass of the "pass" event at delay Y. This is, conceptually, the complement of the F set ("failure" region). By subtracting from one, we get P_{fail}, which is substituted in Equation 3. Repeating this for different Y values (Subsec-

tion III-C) yields the delay distribution of the standard cell.

Clearly, in case $y(\mathbf{x})$ has a finite number of maxima (instead of minima), a dual approach can be maintained. This leads to bounding set F, instead of the latter's complement. The choice between the two courses of action can be resolved with a high-level view of $y(\mathbf{x})$, e.g. by crudely sweeping the $\mathbf{x}$ space.

The techniques of minimum/maximum identification (i.e. Subsection III-A and Algorithm 1) and probability mass calculation (i.e. Subsection III-B) need to be generalized for an arbitrary number of transistors (N) in the standard cell. The two step technique of Subsections III-A and III-B should account for plateaus in $y(\mathbf{x})$ and non-global minima/maxima. All these enhancements constitute points for future work.

V. Generalizing to Standard Cell Paths

Given a (sufficiently) accurate PDF approximation for the delay distribution of a set of standard cells, it is quite easy to provide the delay distribution for a path of standard cells. Given that we target the sum of the delays of the involved standard cells, the respective distribution is produced with convolution of the delay distributions [17]. In Figure 4 we present the results for a chain of four inverters, each one being identical to the one used in Section III. The chain of operations is exactly inverted: we convolute PDF_y the appropriate amount of times (four) and produce Figure 4a, namely the delay PDF for the path of inverters (PDF_{y4}). A simple integration yields the CDF_{y4} (Figure 4b) and subtraction from one provides the failure probability of the simple four-inverter path for various target delays (Y), as illustrated in Figure 4c. We note that the resulting failure probabilities span a wider Y range in comparison to the single-inverter equivalent. Also, there is a general transposition of the nominal delay in comparison to Figure 2a, given the connection of inverters in series.

VI. Conclusions

In the current paper we disclose an iterative technique used to approximate the delay distribution of a standard cell. Complete reduction to practice has been achieved for the case of a simple inverter and extensions are discussed for more complex gates and paths of standard cells. The proposed technique starts from the Most Probable Failure Point (MPFP) concept, which has been traditionally used in prior art for reliability modeling of memory cells. In the current paper, we extendMPFP to improve probability mass coverage, using the χ^2 distribution and coordinate descend.

Acknowledgments

Partial support by European Commission project FP7-612069-HARPA. Prof. Julius Georgiou (UCY, CY) acknowledged for EDA support. Dr. Pieter Weckx (KUL, BE) and Prof. Ramon Canal (UPC, ES) acknowledged for inspiring discussions.

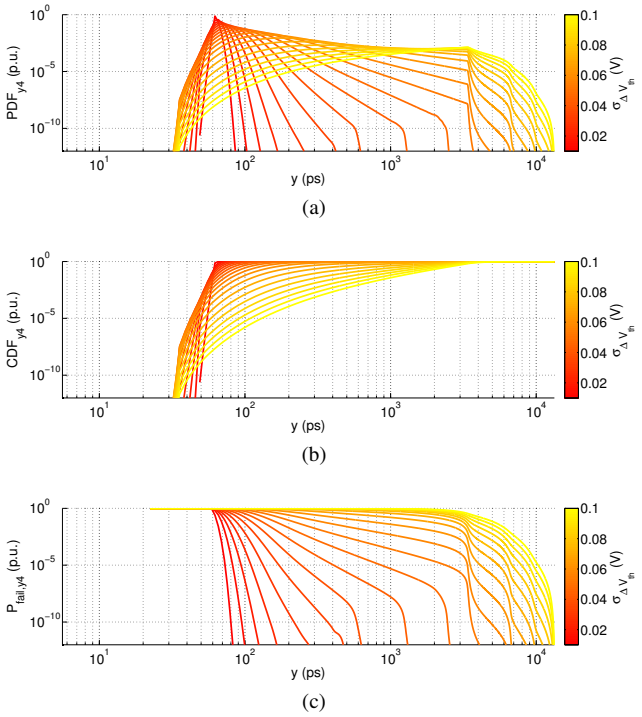


Fig. 4. Iterative convolution of PDF_y from Figure 2c provides the delay PDF, CDF and, eventually, the P_{fail} for a chain of four inverters.

References

[1] D. Rodopoulos, P. Weckx, M. Noltsis, F. Catthoor, and D. Soudris, "Atomistic pseudo-transient bti simulation with inherent workload memory," *IEEE TDMR*, vol. 14, no. 2, pp. 704–714, June 2014.

[2] Visweswariah, C. et al., "First-order incremental block-based statistical timing analysis," *IEEE TCAD*, vol. 25, no. 10, pp. 2170–2180, Oct 2006.

[3] Jess, J.A.G. et al., "Statistical timing for parametric yield prediction of digital integrated circuits," *IEEE TCAD*, vol. 25, no. 11, pp. 2376–2392, Nov 2006.

[4] M. Orshansky and K. Keutzer, "A general probabilistic framework for worst case timing analysis," in *Design Automation Conference, 2002. Proceedings. 39th*, 2002, pp. 556–561.

[5] J.-J. Liou, K.-T. Cheng, S. Kundu, and A. Krstic, "Fast statistical timing analysis by probabilistic event propagation," in *Design Automation Conference, 2001. Proceedings*, 2001, pp. 661–666.

[6] A. Agarwal, D. Blaauw, V. Zolotov, and S. Vrudhula, "Computation and refinement of statistical bounds on circuit delay," in *Design Automation Conference, 2003. Proceedings*, June 2003, pp. 348–353.

[7] A. Dunsmoor and J. ao Geada, "Applications and use of stage-basd ocv," *EE Times*, May 2012.

[8] Khalil, DiaaEldin et al., "Sram dynamic stability estimation using mpfp and its applications," *Microelectron. J.*, vol. 40, no. 11, pp. 1523–1530, Nov. 2009. [Online]. Available: http://dx.doi.org/10.1016/j.mejo.2009.01.015

[9] Ganapathy, S. et al., "Informer: An integrated framework for early-stage memory robustness analysis," in *DATE*, March 2014, pp. 1–4.

[10] Rodopoulos, D. et al., "Sensitivity of sram cell most probable snm failure point to time-dependent variability," in *IEEE SELSE Workshop, Austin–Texas*, 2015.

[11] Synopsys, Inc., "Nanotime – transistor-level static timing analysis solution for custom designs," https://www.synopsys.com/Tools/Implementation/SignOff/Documents/nanotime_ds.pdf, Tech. Rep., 2010.

[12] "Predictive technology model (ptm)," http://ptm.asu.edu/.

[13] N. L. Johnson, S. Kotz, and N. Balakrishnan, *Continuous Univariate Distributions, Volume 2*, May 1995.

[14] Weckx, P. et al., "Implications of bti-induced time-dependent statistics on yield estimation of digital circuits," *IEEE TED*, vol. 61, no. 3, pp. 666–673, March 2014.

[15] M. Miranda, P. Roussel, and L. Brusamarello, "Response characterization of an electronic system under variability effects," Aug. 24 2011, eP Patent App. EP20,100,189,434.

[16] S. J. Wright, "Coordinate Descent Algorithms," *ArXiv e-prints*, Feb. 2015.

[17] C. M. Grinstead and J. L. Snell, *Introduction to Probability*, 2003.