Path-Based Delay Variation Models for Parallel-Prefix Adders

Kleanthis Papachatzopoulos and Vassilis Paliouras, *Member, IEEE*

Abstract—State-of-the-art static timing analysis algorithms can evaluate worst-case delay in statistical terms. In this paper, a modeling framework is introduced for the evaluation of the maximum-delay Cumulative Density Function (CDF) of an ensemble of parallel-prefix adder topologies. For moderate variations and close-to-nominal supply voltages, the maximum delay of parallel-prefix adders is practically determined by the maximum of a set of near-critical-delay paths around the nominal maximum-delay path. These paths end to the most significant and neighboring bit positions. Matrix-based path delay formulations are derived for the particular set of paths. The introduced matrix formulations are exploited to assess the maximum-delay CDF by means of a multivariate Gaussian CDF. To validate the accuracy of the introduced models, a quantitative comparison of the proposed probabilistic delay models against Spice-level Monte-Carlo simulations is offered for certain parallel-prefix adders. Threshold-voltage variations summarize several process-dependent variation-inducing mechanisms and are modeled as Gaussian variations, introduced to BSIM-4 transistor models for a 16-nm technology node. For the nominal voltage case and 10% threshold-voltage variations, the introduced models estimate the 0.95 timing yield point with a mean absolute error below 1% compared to Spice-level simulations for the 16 bit-length case. Furthermore, an extension of the proposed approach to account for multiple end points is investigated that reduces the error for the estimation of maximum delay, demonstrated for a unit delay model and certain bit-lengths of Kogge-Stone adder.

Index Terms—parallel-prefix adders, critical path delay, threshold-voltage variations, statistical static timing analysis, timing yield

1 Introduction

Process variability in advanced technology nodes of a few nanometers extends design margins, already in place to accommodate for temperature and voltage variations. This further escalates problems in timing closure and power budgeting [1]. Traditional static timing analysis considers only worst-case delay corners; it provides conservative solutions and, consequently, extensively wastes performance and power. This necessitates a change in timing analysis to account for the additional uncertainty of process-dependent variations, and efficiently evaluate the timing yield at each design iteration [2].

Delay variability is typically evaluated at the transistor level through Monte-Carlo (MC) simulations, incorporating correlated device parameters and non-Gaussian variation sources [3]. However, transistor-level Monte-Carlo simulations demand excessive computational load when extended to large-scale systems [4]. Variations of device parameters can be potentially abstracted at a higher level and modeled as delay variations, represented by Random Variables (RVs). Indicatively, the variances of cell delays and of threshold voltage are proportional [5].

Statistical Static Timing Analysis (SSTA) algorithms have been introduced as an alternative that reduces computational effort. SSTA algorithms propagate gate and interconnection delays through the circuit graph as Probability Density Functions (PDFs) and assess the maximum-delay PDF at the circuit output [3], [6], [7]. Commonly, a Gaussian distribution is preferred to model cell/gate delays [8], although log-normal and stable distributions better describe certain cell/gate PDFs in low supply voltages [9]. The use of Gaussian PDFs is preferred due to their convenient mathematical properties; furthermore, the delay along paths of sufficient logic depth can be modeled as a Gaussian due to the central limit theorem, since the path delay is conceived as a linear combination of the delays of logic along the path [10].

The computational effort required by SSTA remains significant even for designs of typical complexity [11]. EDA tools use Statistical On-Chip-Variation (SOCV) techniques that derate gate delays during timing analysis, accounting also for the input slew and load [12]. The provided accuracy is comparable with that of SSTA algorithms, rendering the SOCV the preferred sign-off timing approach at advanced nodes [13]. Regarding interconnection variability, closed-form approaches have been developed that link physical fluctuations on wires and delay metrics [14], [15]. When process variations are considered as Gaussian RVs, interconnection delay tends to have a Gaussian shape that allows integration to existing SSTA frameworks [14], [16]. Furthermore, EDA tools include interconnection variations by relying on extreme back-end-of-line (BEOL) corners for sign-off timing analysis [17]. Furthermore, analytical approaches have been proposed to assess path delay variability and provide insights on how circuit-level parameters and design knobs can control variability [5], [11], [18].

Parameter variations are distinguished into *inter-die* and *intra-die* components when modeling completely dependent and independent variation sources, respectively. The former source from systematic variation mechanisms globally affecting parameters across a die or a lot, *e.g.*, channel

Manuscript received September 15, 2022; accepted March XX, XXXX. Date of publication May XX, XXXX; date of current version January 17, 2023.

- *The authors are with the Department of Electrical and Computer Engineering, University of Patras, 26504 Patras, Greece. E-mail: papachatz@ece.upatras.gr; paliuras@ece.upatras.gr*

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.
Digital Object Identifier

length variations. The latter are attributed to atomic-level and random higher-scale fluctuations, *e.g.,* random dopant fluctuations and line-edge roughness of bulk CMOS technologies [4]. Nonetheless, intra-die variations prevail over inter-die ones in advanced technologies for reduced logic depths and low voltages [3].

Even under nominal conditions, adders are usually timing-critical units in high-performance applications [19]. Hence, delay variations in adders play a detrimental role on the overall system delay sensitivity. Furthermore, the study of adder behavior under delay variations provides insights for more elaborate arithmetic structures. In a variation-prone context, certain arithmetic encoding schemes and hardware implementations have been proven more efficient than binary counterparts in statistical performance terms. In [20], [21], it is shown that the borrow-save encoding leads to hardware architectures that combine power efficiency and low standard deviation of maximum delay. Furthermore, Alioto *et al.* contribute simplified circuit models for Mirror, Dual-Rail, Complimentary Pass-Transistor Logic, and Transmission Gate full adders for the evaluation of delay sensitivity to supply voltage variations [19]. Bernstein *et al.* demonstrate a preliminary evaluation of certain 16-bit static and dynamic implementations of adders in normalized delay terms based on simulations [22]. Patil *et al.* observe that small logic paths contribute to increased delay variability of certain parallel-prefix adders, and rely on SSTA optimizations for the minimization of delay variations [23]. Prior work in [24] also investigates methods for the estimation of maximum delay performance of parallel-prefix structures under variations and relies on a path-based approach. In [24], it is shown that it is possible to evaluate the worst-case delay of 16-bit adders by considering only a small set of potential critical paths with an error less than 6%. However, the approach relies on MC simulations for the extraction of key statistical metrics of the underlying maximum-delay distributions, subsequently used for fitting. Thus, the approach is computationally intensive. A matrix representation for the prefix network of Knowles adders is proposed in [25]. The nominal critical-path delay is evaluated based on only three logic paths and the maximum delay error is as high as 14% compared to Spice simulations. EDA tools evaluate maximum delay under variations by considering the $\mu_{\text{nom-crit}} + 3\sigma_{\text{nom-crit}}$ delay metric, where $\mu_{\text{nom-crit}}$ ($\sigma_{\text{nom-crit}}$) is the mean delay (standard deviation) of the nominal maximum-delay critical path [13]. Previous works that evaluate the performance of adders rely solely on simulations and avoid parallel-prefix adders, while models in [25] refer only to Knowles adders. Thus, prior art does not provide closed-form models that cover a range of parallel-prefix adder design choices, are computational efficient, accurate compared to Spice simulations, and suitable for the evaluation of delay variability of long bit-length adders, where Spice-level MC simulations are prohibitive. To the best of our knowledge, this is the first research effort that addresses delay variability of parallel-prefix adders with detailed formulas. This paper contributes closed-form models that interpret the results of delay variability simulations, presented in [24]. The relative performance of investigated parallel-prefix adders, obtained by the proposed models, is in agreement with the simulation results in [23].

This paper derives and evaluates statistical delay-variability models for certain parallel-prefix adder architectures. The introduced statistical models, provided in the form of maximum-delay Cumulative Density Functions (CDFs), are utilized for the estimation of worst-case delay at the tails of the obtained maximum-delay distributions. Specifically, in this paper

- Matrix-based formulations are derived that describe path delays which have a significant probability to become maximum-delay critical under variations. The formulation considers the delays of the nominal maximum-delay critical path and paths with comparable delays at the same bit position, thus reducing the dimensionality of the problem.
- The CDF of the maximum adder delay is approximated by exploiting the introduced matrix-based formulation. The proposed approach uses RVs, jointly described by a multivariate Gaussian CDF, to model the delays of the processing nodes.
- An algorithmic approach is demonstrated that formulates path delays of a generic parallel-prefix structure.
- By employing a unit gate delay model and performing MC simulations for a range of bit-lengths of a Kogge-Stone adder, it is shown that the proposed delay CDF perfectly matches the MC-based CDF of the maximum delay, observed at the MSB position. Furthermore, it shown that extending the approach to take into account paths to a certain set of the adder sum output bit positions, in the vicinity of the MSB, improves fitting to MC-based CDFs that refer to the complete sum word, even when assuming independence of the delays of the particular paths. Analysis considers also correlated delays.
- The effectiveness of the introduced models is further investigated by comparisons to Spice-level MC simulations. In the presence of threshold-voltage variability at a 16-nm technology node, it is validated that the introduced models estimate the timing yield at quantile points of practical interest with sufficient accuracy; *i.e.*, the maximum relative error for the estimation of worst-case delay of the investigated 16-bit parallel-prefix adders does not exceed 2.2%.

Model extension to more elaborate structures is also possible. Ling [26] and sum-propagate [27] parallel-prefix adders utilize the commonly used prefix network structures, although they differentiate on the logic implemented by the nodes. Therefore, the proposed models are applicable for such architectures as well. Furthermore, they can be used for more elaborate adders, with simple modifications, resembling the case of carry-select adders in [20].

The remainder of this paper is structured as follows: Section 2 reviews addition principles and revisits parallel-prefix adder architectures. Section 3 introduces path delay models for the estimation of maximum-delay CDF for certain parallel-prefix adders in the presence of variations, and an algorithm for the general case. Section 4 investigates the accuracy of introduced CDFs for unit delay models and delay variations, and Spice-level simulations in the presence

of threshold-voltage variations. Finally, Section 5 presents conclusions.

2 Parallel-Prefix Adder Architectures

For the addition of two n-bit binary numbers $A_n = a_{n-1}a_{n-2}\cdots a_0$ and $B_n = b_{n-1}b_{n-2}\cdots b_0$, the modulo-2 sum (XOR operation) of bits a_i and b_i, $P_i = a_i \oplus b_i$, is combined with the carry signal C_{i-1}, computed in the previous bit position, for the evaluation of sum bit S_i at the ith bit position, as

$$S_i = P_i \oplus C_{i-1}. \tag{1}$$

The propagation of incoming carry C_{i-1} enables the evaluation of sum bits at the following bit positions. The carry at the ith bit position can be computed by the recursion

$$C_i = G_i + P_i C_{i-1}, \tag{2}$$

exploiting the locally computed generate, $G_i = a_i b_i$, and propagate, P_i, signals.

Carry computation, expressed as a parallel-prefix problem, offers logarithmic-scale delays [34]. Carry-lookahead adders are typically structured in pre-processing, prefix-processing and post-processing stages. The pre-processing stage computes the bit-wise generate G_i and propagate P_i signals utilized in the prefix-processing stage. Prefix processing nodes associate group propagate and generate bit pairs, exploiting the prefix operator $\circ$, defined as

$$(G, P) \circ (G', P') = (G + PG', PP'). \tag{3}$$

The carry signal, $C_i = G_{i:0}$, is obtained by consecutive associations of group generate and propagate bits, *i.e.*, $(G_{i:k}, P_{i:k})$, for bit positions $i, i-1, \ldots, k$, using

$$(G_{i:k}, P_{i:k}) = (G_i, P_i) \circ (G_{i-1}, P_{i-1}) \circ \cdots \circ (G_k, P_k). \tag{4}$$

Then, the post-processing stage computes the sum bits based on (1).

Leveraging the properties of associativity and idempotence of prefix operator $\circ$ [34], [35], carry-lookahead adders offer hardware implementations with efficient regularity and placement, spanning a range of number of wiring tracks, fanout and logic depth combinations [35]–[37].

For bit lengths larger than 32 bits, the selection of an optimal prefix structure is not straightforward, as the design space grows exponentially. State-of-the-art design approaches focus on efficient exploration methods to maximize area and delay benefits. Roy *et al.* in [38] exploit deep reinforcement learning for the construction of Pareto-optimal parallel-prefix adders of long bit lengths. Ene and Stine in [39] propose a tree representation that handles the pre- and post-processing stages as an integral part of the overall adder structure. The introduced optimizations lead to architectures with delay benefits compared to synthesized adders by commercial tools.

3 Proposed Maximum-Delay CDFs for Parallel-Prefix Adders

This section derives expressions for the approximation of maximum-delay CDF of certain parallel-prefix adders. The presented statistical models consider radix-2 Kogge-Stone [28], Sklansky[1] [29], Knowles [30], Han-Carlson [31], Ladner-Fischer [32], and Brent-Kung [33] topologies. An algorithm for the path delay formulation of generic parallel-prefix structures is also introduced. Indicatively, Fig. 1 demonstrates 16-bit topologies for the investigated adders. In the following, basic assumptions for variation are stated, and the notation used in the remainder of the paper is defined. Subsequently, the introduced models are derived, as the CDF of the maximum delay per adder.

3.1 Notation and Assumptions

The delay D_i of a logic cell or gate is decomposed as

$$D_i = D_{i,\text{nom}} + D_{i,\text{var}}, \tag{5}$$

where $D_{i,\text{nom}}$ is the nominal (deterministic) delay of ith logic cell or gate, and $D_{i,\text{var}}$ constitutes the respective zero-mean variation component, described by a specific PDF, of a variation source. In the case of the two correlation extremes, the notation $D_{i,\text{inter}}$ and $D_{i,\text{intra}}$ is employed to declare the respective inter-die and intra-die delay variation components, respectively. In this paper, we examine delay variations sourcing from atomic-level fluctuations of dopant atoms in the transistor channel causing variations on threshold voltage [4], [5]. Furthermore, threshold-voltage variations possibly model other less significant variations, *i.e.*, channel length, mobility and oxide thickness variations [5]. As random dopant fluctuations are manifested without correlation among devices, we consider delay variations, completely independent between cells in Spice analysis. For these reasons, D_i is also a RV with mean value $\mu_{D_i} = D_{i,\text{nom}}$ and variance $\sigma^2_{D_i} = \sigma^2_{D_{i,\text{var}}}$. The resulting cell/gate delay distribution, subjected to either threshold voltage or channel length variations, has demonstrated a skewness parameter below 0.1 (cf. [9, Fig. 1]) for supply voltage values around the nominal one. Based on this, D_i is assumed Gaussian. Hereafter, notation is as follows:

- A boldfaced variable $\boldsymbol{X}$ denotes a vector or a matrix of delays, modeled as RVs,
- $\boldsymbol{\mu}_X$ is the vector of mean values of $\boldsymbol{X}$,
- $\boldsymbol{\Sigma}_X$ is the covariance matrix of $\boldsymbol{X}$,
- $\text{cdf}_{\boldsymbol{X}}$ and $\text{pdf}_{\boldsymbol{X}}$ are the joint CDF and joint PDF of $\boldsymbol{X}$, respectively,
- σ_Y is the standard deviation of a RV Y.

Furthermore, $\otimes$ stands for the Kronecker product, $\boldsymbol{I}_i$ is an $i \times i$ identity matrix, $\mathbf{1}_{i\times j}$ is an $i \times j$ all-one matrix, and $\mathbf{0}_{i\times j}$ is an $i \times j$ all-zero matrix.

3.2 CDF of Maximum Delay - General Form

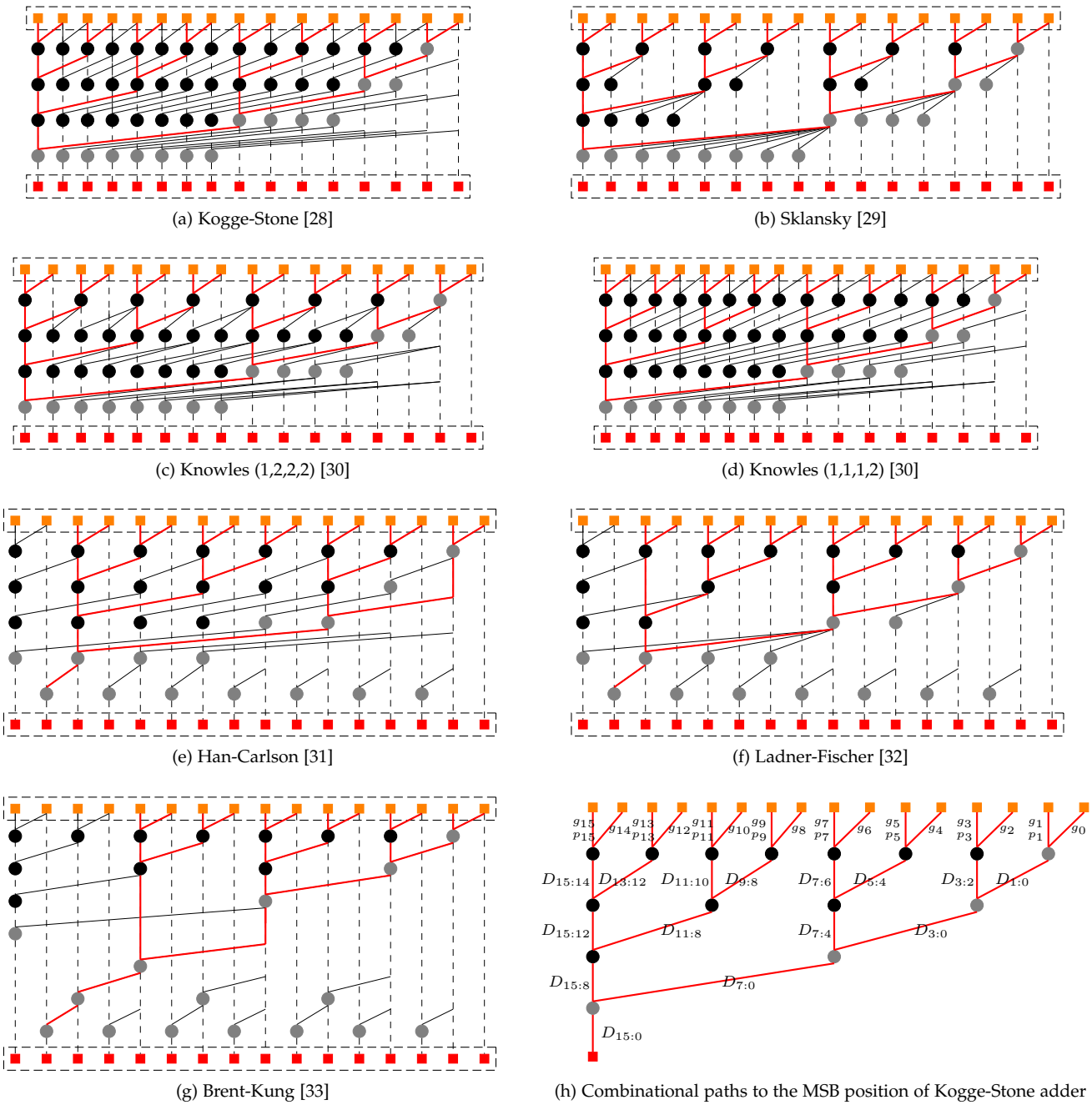
We seek the CDF of RV $d_{\max}$ that describes the maximum delay observed at the sum outputs of an n-bit parallel-prefix adder. We define as $\boldsymbol{D}$,

$$\boldsymbol{D} = \left[D_{1,1}\ D_{1,2} \cdots D_{1,m_1} \cdots D_{n,1}\ D_{n,2} \cdots D_{n,m_n} \right]^T, \tag{6}$$

the vector of path delays, where $D_{i,j}$ is the delay of jth path to the ith sum bit output. Furthermore, let m_i denote the

1. Sklansky topology belongs to the family of Ladner-Fischer adders. In this paper, we refer to a Ladner-Fischer adder with the highest logical branching as a Sklansky adder.

This article has been accepted for publication in IEEE Transactions on Emerging Topics in Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TETC.2023.3242555



(a) Kogge-Stone [28] (b) Sklansky [29]

(c) Knowles (1,2,2,2) [30] (d) Knowles (1,1,1,2) [30]

(e) Han-Carlson [31] (f) Ladner-Fischer [32]

(g) Brent-Kung [33] (h) Combinational paths to the MSB position of Kogge-Stone adder

Fig. 1. 16-bit topologies of parallel-prefix adders. Orange nodes refer to first-stage propagate-generate cells, black to group propagate-generate cells, gray to group generate cells, and red to XOR cells. Prefix paths considered by the introduced models are marked red.

number of paths to the ith sum output. Since $d_{\max}$ is the maximum of path delays in $\boldsymbol{D}$, it holds that

$$\begin{aligned} \mathrm{cdf}_{d_{\max}}(D_{\max}) &= \mathrm{P}(d_{\max} \leq D_{\max}) && (7)\\ &= \mathrm{P}(D_{1,1} \leq D_{\max} \wedge D_{1,2} \leq D_{\max} \\ &\qquad \cdots \wedge D_{n,m_n} \leq D_{\max}) && (8)\\ &= \mathrm{cdf}_{\boldsymbol{D}}(D_{\max}, D_{\max}, \ldots, D_{\max}), && (9) \end{aligned}$$

where $\mathrm{P}(\cdot)$ denotes the probability of an event and $D_{\max}$ is any value that $d_{\max}$ can assume. Let k be the bit position where the nominal maximum-delay critical path ends, and define $\boldsymbol{D}_k$,

$$\boldsymbol{D}_k = \left[D_{k,1} \; D_{k,2} \cdots D_{k,m_k} \right]^T, \qquad (10)$$

as the vector of the corresponding path-delay RVs. The delays that compose $\boldsymbol{D}_k$ refer to potentially maximum-delay critical paths in the presence of delay variations. These paths are likely to traverse all prefix stages. Path selection under nominal circumstances is a technique commonly utilized in path-based SSTA algorithms and MC simulations that approximate the maximum-delay PDF [1], [4].

To make the estimation of $\mathrm{cdf}_{d_{\max}}(D_{\max})$ computationally tractable, we reduce the number of involved paths and approximate $\mathrm{cdf}_{d_{\max}}(D_{\max})$ of (9) as

$$\mathrm{cdf}_{d_{\max}}(D_{\max}) \approx \mathrm{cdf}_{\boldsymbol{D}_k}(D_{\max}). \qquad (11)$$

In the following, we exploit two choices for the $\mathrm{cdf}_{\boldsymbol{D}_k}(D_{\max})$, namely, a multivariate Gaussian and a mul-

tivariate log-normal. The introduced treatment can be straightforwardly extended to other choices of the multivariate CDF. We present models for the computation of $\mathrm{cdf}_{\boldsymbol{D}_k}(D_{\max})$ relying on a multivariate Gaussian distribution, given by

$$\begin{aligned}\mathrm{cdf}_{\boldsymbol{D}_k}(D_{\max}) &= \mathrm{P}(D_{k,1} \leq D_{\max} \wedge D_{k,2} \leq D_{\max} \\ &\qquad \cdots \wedge D_{k,m_k} \leq D_{\max}) \qquad (12)\\ &= \mathrm{cdf}_{\boldsymbol{D}_k}(D_{\max}, D_{\max}, \ldots, D_{\max}) \qquad (13)\\ &= \underbrace{\int_{-\infty}^{D_{\max}} \int_{-\infty}^{D_{\max}} \cdots \int_{-\infty}^{D_{\max}}}_{m_k \text{ integrals}} \mathrm{Gauss}_{\boldsymbol{D}_k}(\boldsymbol{D}_k, \boldsymbol{\mu}_{D_k}, \boldsymbol{\Sigma}_{D_k}) d\boldsymbol{D}_k, \qquad (14)\end{aligned}$$

where $\mathrm{Gauss}_{\boldsymbol{D}_k}(\cdot)$ is the m_k-variate Gaussian joint PDF of $\boldsymbol{D}_k$. The evaluation of $\mathrm{cdf}_{\boldsymbol{D}_k}(D_{\max})$ requires m_k integrals with respect to the variables $D_{k,i}$ for $i = 1, 2, \ldots, m_k$ in the interval $(-\infty, D_{\max})$.

To evaluate $\mathrm{cdf}_{\boldsymbol{D}_k}$, path-delay RVs in $\boldsymbol{D}_k$ are expressed as the sum of primary bit-wise generate/propagate and group generate/propagate cell delay RVs. We denote as g_i and p_i the delay RV of bit-wise generate and propagate cell at the ith bit position, respectively, and $D_{i:j}$ the delay RV of the $i:j$ group generate/propagate cell. To form a compact model, $\boldsymbol{D}_k$ is expressed as the sum of three matrices, denoting distinctly the delays of the pre-processing, prefix-processing and post-processing operations. Matrix-based formulations are initially derived for $\frac{m_k}{3}$ paths referring only to prefix nodes and captured by $\boldsymbol{D}_{\text{prefix}}$. Then, every path delay of $\boldsymbol{D}_{\text{prefix}}$ is extended to three separate paths, starting from a certain generate or propagate cell of the pre-processing stage. Specifically, we write the path-delay RV vector as

$$\boldsymbol{D}_k = \boldsymbol{F} + (\boldsymbol{I}_{\frac{m_k}{3}} \otimes \boldsymbol{1}_{3\times 1})\boldsymbol{D}_{\text{prefix}} + \boldsymbol{1}_{m_k\times 1} f_{\text{XOR}} + \boldsymbol{T}, \quad (15)$$

where $\boldsymbol{F}$ comprises the delay RVs of the pre-processing stage cells, *i.e.*,

$$\boldsymbol{F} = \begin{bmatrix} g_{k-1} & p_{k-1} & g_{k-2} & g_{k-3} & p_{k-3} & g_{k-4} & \cdots & g_0 \end{bmatrix}^T, \quad (16)$$

$\boldsymbol{D}_{\text{prefix}}$ is composed of the path delay RVs of the prefix-processing stage and f_{XOR} is the delay RV of the XOR cell which computes the kth sum output and is common for all paths. Given that interconnection delay can be also approximated as a Gaussian RV [14], the delay vector $\boldsymbol{T}$ of (15) accounts for the interconnection delay contribution. In [40], it is shown that the contribution of the interconnection delay on the total delay of parallel-prefix adders is smaller than 10%. Therefore, interconnection delays [12] and the related variability are assumed negligible and $\boldsymbol{T} = \boldsymbol{0}$ for the derived models.

Due to the linear transformation (15), the mean value of $\boldsymbol{D}_k$, denoted as $\boldsymbol{\mu}_{D_k}$, is

$$\boldsymbol{\mu}_{D_k} = \boldsymbol{\mu}_F + (\boldsymbol{I}_{\frac{m_k}{3}} \otimes \boldsymbol{1}_{3\times 1})\boldsymbol{\mu}_{D_{\text{prefix}}} + \boldsymbol{1}_{m_k\times 1}\mu_{f_{\text{XOR}}}. \quad (17)$$

In the general case, the covariance matrix $\boldsymbol{\Sigma}_{D_k}$ of $\boldsymbol{D}_k$ is computed as

$$\begin{aligned}\boldsymbol{\Sigma}_{D_k} = \boldsymbol{\Sigma}_F &+ \boldsymbol{K}\boldsymbol{\Sigma}_{D_{\text{prefix}}}\boldsymbol{K}^T + \boldsymbol{\Sigma}_{V_{\text{XOR}}} + \mathrm{Cov}(\boldsymbol{F}, \boldsymbol{D}_{\text{prefix}})\boldsymbol{K}^T \\ &+ \boldsymbol{K}\,\mathrm{Cov}(\boldsymbol{D}_{\text{prefix}}, \boldsymbol{F}) + \mathrm{Cov}(\boldsymbol{F}, \boldsymbol{V}_{\text{XOR}}) \\ &+ \mathrm{Cov}(\boldsymbol{V}_{\text{XOR}}, \boldsymbol{F}) + \mathrm{Cov}(\boldsymbol{V}_{\text{XOR}}, \boldsymbol{D}_{\text{prefix}})\boldsymbol{K}^T \\ &+ \boldsymbol{K}\,\mathrm{Cov}(\boldsymbol{D}_{\text{prefix}}, \boldsymbol{V}_{\text{XOR}}), \qquad (18)\end{aligned}$$

where $\mathrm{Cov}(\boldsymbol{A}, \boldsymbol{B})$ is the covariance matrix of $\boldsymbol{A}$ and $\boldsymbol{B}$, and $\boldsymbol{K} = (\boldsymbol{I}_{\frac{m_k}{3}} \otimes \boldsymbol{1}_{3\times 1})$, $\boldsymbol{V}_{\text{XOR}} = \boldsymbol{1}_{m_k\times 1} f_{\text{XOR}}$ and $\boldsymbol{\Sigma}_{V_{\text{XOR}}} = \boldsymbol{1}_{m_k\times 1}\sigma^2_{f_{\text{XOR}}}$.

When there is no correlation between cell delay RVs in $\boldsymbol{F}$, $\boldsymbol{D}_{\text{prefix}}$ and f_{XOR}, (18) is reduced to

$$\begin{aligned}\boldsymbol{\Sigma}_{D_k} = \boldsymbol{\Sigma}_F &+ (\boldsymbol{I}_{\frac{m_k}{3}} \otimes \boldsymbol{1}_{3\times 1})\boldsymbol{\Sigma}_{D_{\text{prefix}}}(\boldsymbol{I}_{\frac{m_k}{3}} \otimes \boldsymbol{1}_{3\times 1})^T \\ &+ \boldsymbol{1}_{m_k\times 1}\sigma^2_{f_{\text{XOR}}}. \qquad (19)\end{aligned}$$

The simplified form of (19) captures only spatial correlations among delays of $\boldsymbol{F}$ ($\boldsymbol{D}_{\text{prefix}}$) through $\boldsymbol{\Sigma}_F$ ($\boldsymbol{\Sigma}_{D_{\text{prefix}}}$). Thus, a potential process- and/or environmental-induced parameter that introduces correlation among gate delays, *e.g.*, channel length, temperature, or interconnection correlations, is partly taken into account when using (19). Instead, (18) is used without loss of accuracy. In either case, variations are expected to be primarily local in advanced technology nodes, as global variations are typically compensated during manufacturing stages [4]. In the examined Spice-level scenario, threshold voltage variations are considered, sourcing from (uncorrelated) random dopant fluctuations [3], [5], [22]. Therefore, device delays are also uncorrelated, and the computation of covariance matrix $\boldsymbol{\Sigma}_{D_k}$ according to (19) does not miss any correlation between cell delays. Moreover, (15) that relates cell and path delays, exactly captures the structural path correlation of prefix network and does not rely on assumptions for the cell delay correlation.

The proposed derivation of maximum-delay CDFs is summarized in the following three steps:

- computation of $\boldsymbol{\mu}_{D_k}$, given by (17);
- computation of $\boldsymbol{\Sigma}_{D_k}$, given either by (18) or (19); and
- evaluation of $\mathrm{cdf}_{\boldsymbol{D}_k}(D_{\max})$, given by (14).

The multivariate Gaussian CDF is conveniently available in computing frameworks as a function, such as function mvncdf in MATLAB [41].

The derived matrix-based transformations, presented in the remainder of the section, are not limited to the Gaussian case and can accommodate any type of multivariate distribution. In case of low supply voltage operation, a multivariate log-normal distribution, can be adopted instead of (14), and, when combined with the derived transformations, it enhances the fitting accuracy to Spice-level maximum-delay CDFs [9]. In the following, we derive expressions for $\boldsymbol{D}_{\text{prefix}}$ per adder topology.

3.3 Maximum-Delay CDF for a Kogge-Stone Adder

To simplify derivation, we initially focus on the case of a 16-bit Kogge-Stone adder and the paths that end at the nth bit position. The MSB position is the end point of $\frac{n}{2} = 8$ paths, depicted in Fig. 1(h). We derive $\boldsymbol{D}_{\text{prefix,KS}}$ that

expresses each prefix path-delay RV as the sum of the delays $D_{i:j}$ of the prefix nodes located along the path, *i.e.*,

$$\boldsymbol{D}_{\text{prefix,KS}} = \left[\, D_{\text{KS},1}\ D_{\text{KS},2}\ \cdots\ D_{\text{KS},\frac{n}{2}} \,\right]^T, \tag{20}$$

where

$$D_{\text{KS},1} = D_{15:14} + D_{15:12} + D_{15:8} + D_{15:0} \tag{21}$$
$$D_{\text{KS},2} = D_{13:12} + D_{15:12} + D_{15:8} + D_{15:0} \tag{22}$$
$$D_{\text{KS},3} = D_{11:10} + D_{11:8} + D_{15:8} + D_{15:0} \tag{23}$$
$$D_{\text{KS},4} = D_{9:8} + D_{11:8} + D_{15:8} + D_{15:0} \tag{24}$$
$$D_{\text{KS},5} = D_{7:6} + D_{7:4} + D_{7:0} + D_{15:0} \tag{25}$$
$$D_{\text{KS},6} = D_{5:4} + D_{7:4} + D_{7:0} + D_{15:0} \tag{26}$$
$$D_{\text{KS},7} = D_{3:2} + D_{3:0} + D_{7:0} + D_{15:0} \tag{27}$$
$$D_{\text{KS},8} = D_{1:0} + D_{3:0} + D_{7:0} + D_{15:0}. \tag{28}$$

The vector $\boldsymbol{D}_{\text{prefix,KS}}$ of (20) can be expressed as a matrix-vector product. We define the vector $\boldsymbol{G}_{\text{KS}}$ of $n-1=15$ RVs, which contains prefix node-delay RVs,

$$\boldsymbol{G}_{\text{KS}} = \begin{bmatrix} D_{15:14} \\ D_{13:12} \\ D_{11:10} \\ D_{9:8} \\ D_{7:6} \\ D_{5:4} \\ D_{3:2} \\ D_{1:0} \\ \hdashline D_{15:12} \\ D_{11:8} \\ D_{7:4} \\ D_{3:0} \\ \hdashline D_{15:8} \\ D_{7:0} \\ \hdashline D_{15:0} \end{bmatrix} \begin{matrix} \left.\vphantom{\begin{matrix} a \\ a \\ a \\ a \\ a \\ a \\ a \\ a \end{matrix}}\right\} \text{level 1, leaf nodes} \\ \left.\vphantom{\begin{matrix} a \\ a \\ a \\ a \end{matrix}}\right\} \text{level 2} \\ \left.\vphantom{\begin{matrix} a \\ a \end{matrix}}\right\} \text{level 3} \\ \left.\vphantom{a}\right\} \text{level 4, root node.} \end{matrix} \tag{29}$$

Vector $\boldsymbol{G}_{\text{KS}}$ is constructed by breadth-first traversing the binary tree, depicted in Fig. 1(h), that describes the computation. Then, the derivation of $D_{\text{KS},i}$ for $i=1,2,\ldots,8$ of (21) to (28) can be described by the product of $\mathbf{G}_{\text{KS}}$ with the 8×15 matrix $\mathbf{A}_{16}$,

$$\boldsymbol{A}_{16} = \begin{bmatrix} 1&0&0&0&0&0&0&0&1&0&0&0&1&0&1 \\ 0&1&0&0&0&0&0&0&1&0&0&0&1&0&1 \\ 0&0&1&0&0&0&0&0&1&0&0&1&0&1&1 \\ 0&0&0&1&0&0&0&0&0&1&0&0&1&0&1 \\ 0&0&0&0&1&0&0&0&0&0&1&0&0&1&1 \\ 0&0&0&0&0&1&0&0&0&0&1&0&0&1&1 \\ 0&0&0&0&0&0&1&0&0&0&0&1&0&1&1 \\ 0&0&0&0&0&0&0&1&0&0&0&1&0&1&1 \end{bmatrix} \tag{30}$$

$$= \left[\boldsymbol{I}_8 \otimes \mathbf{1}_{1\times 1}\ \ \boldsymbol{I}_4 \otimes \mathbf{1}_{2\times 1}\ \ \boldsymbol{I}_2 \otimes \mathbf{1}_{4\times 1}\ \ \boldsymbol{I}_1 \otimes \mathbf{1}_{8\times 1}\right]. \tag{31}$$

Due the binary tree structure of the computation, for the general case of the n-bit adder, where n is a power of two, we write

$$\boldsymbol{A}_n = \left[\boldsymbol{I}_{\frac{n}{2}} \otimes \mathbf{1}_{1\times 1} \quad \boldsymbol{I}_{\frac{n}{4}} \otimes \mathbf{1}_{2\times 1} \quad \cdots \quad \boldsymbol{I}_1 \otimes \mathbf{1}_{\frac{n}{2}\times 1}\right], \tag{32}$$

and

$$\boldsymbol{D}_{\text{prefix,KS}} = \boldsymbol{A}_n \boldsymbol{G}_{\text{KS}}. \tag{33}$$

$\boldsymbol{A}_n$ is an $\frac{n}{2}\times(n-1)$ matrix. Then, from (29) and (32), (15) is re-expressed for the case of Kogge-Stone adder as

$$\boldsymbol{D}_k = \boldsymbol{F} + (\boldsymbol{I}_{\frac{n}{2}} \otimes \mathbf{1}_{3\times 1})\boldsymbol{A}_n \boldsymbol{G}_{\text{KS}} + \mathbf{1}_{\frac{3n}{2}\times 1} f_{\text{XOR}}, \tag{34}$$

where $\boldsymbol{F}$ is given by (16) for $k=n$ and $m_k = \frac{3n}{2}$. Accordingly, the mean value vector $\boldsymbol{\mu}_{D_k}$ for a Kogge-Stone adder is

$$\boldsymbol{\mu}_{D_k} = \boldsymbol{\mu}_F + (\boldsymbol{I}_{\frac{n}{2}} \otimes \mathbf{1}_{3\times 1})\boldsymbol{A}_n \boldsymbol{\mu}_{G_{\text{KS}}} + \mathbf{1}_{\frac{3n}{2}\times 1}\mu_{f_{\text{XOR}}}. \tag{35}$$

Assuming uncorrelated RVs $\boldsymbol{F}, \boldsymbol{G}_{\text{KS}}$, and f_{XOR}, (19) gives

$$\boldsymbol{\Sigma}_{D_k} = \boldsymbol{\Sigma}_F + (\boldsymbol{I}_{\frac{n}{2}} \otimes \mathbf{1}_{3\times 1})(\boldsymbol{A}_n \boldsymbol{\Sigma}_{G_{\text{KS}}} \boldsymbol{A}_n^T)(\boldsymbol{I}_{\frac{n}{2}} \otimes \mathbf{1}_{3\times 1})^T + \mathbf{1}_{\frac{3n}{2}\times 1}\sigma^2_{f_{\text{XOR}}}, \tag{36}$$

using the mean value $\boldsymbol{\mu}_{\boldsymbol{G}_{\text{KS}}}$ and covariance matrices $\boldsymbol{\Sigma}_{\boldsymbol{G}_{\text{KS}}}$ referring to vector $\boldsymbol{G}_{\text{KS}}$. The CDF of maximum delay is evaluated by combining (14), (34), (35) and (36).

3.4 Maximum-Delay CDF for a Sklansky Adder

The nominal maximum-delay critical path in an n-bit Sklansky adder starts from the LSB position of the input, and traverses $\log_2(n)$ prefix stages up to the nth bit position of the result. The nth bit position is the end point for $\frac{n}{2}$ prefix paths. The path-delay RVs can be expressed as

$$\boldsymbol{D}_k = \boldsymbol{F} + (\boldsymbol{I}_{\frac{m_k}{3}} \otimes \mathbf{1}_{3\times 1})\boldsymbol{D}_{\text{prefix,SKL}} + \mathbf{1}_{m_k\times 1} f_{\text{XOR}}, \tag{37}$$

where

$$\boldsymbol{D}_{\text{prefix,SKL}} = \boldsymbol{A}_n \boldsymbol{G}_{\text{SKL}}, \tag{38}$$

$$\boldsymbol{G}_{\text{SKL}} = \begin{bmatrix} D_{15:14} \\ D_{13:12} \\ D_{11:10} \\ D_{9:8} \\ D_{7:6} \\ D_{5:4} \\ D_{3:2} \\ D_{1:0}^{\times 3} \\ \hdashline D_{15:12} \\ D_{11:8} \\ D_{7:4} \\ D_{3:0}^{\times 5} \\ \hdashline D_{15:8} \\ D_{7:0}^{\times 9} \\ \hdashline D_{15:0} \end{bmatrix} \begin{matrix} \left.\vphantom{\begin{matrix} a \\ a \\ a \\ a \\ a \\ a \\ a \\ a \end{matrix}}\right\} \text{level 1, leaf nodes} \\ \left.\vphantom{\begin{matrix} a \\ a \\ a \\ a \end{matrix}}\right\} \text{level 2} \\ \left.\vphantom{\begin{matrix} a \\ a \end{matrix}}\right\} \text{level 3} \\ \left.\vphantom{a}\right\} \text{level 4, root node} \end{matrix}, \tag{39}$$

$\boldsymbol{F}$ is given by (16) for $k=n, m_k=\frac{3n}{2}$, and $\boldsymbol{A}_n$ by (32). A Sklansky adder presents the same binary tree structure for paths that end to the nth bit position as a Kogge-Stone adder, both visiting prefix nodes with the same indexes. The two structures have prefix nodes of different fanout; all prefix nodes of a Kogge-Stone adder have a fanout of two, while a prefix node at the jth prefix stage and (2^j-1)th bit position of a Sklansky adder has a fanout of 2^j+1. Fanout is defined as the maximum logical branching (cf. [23]). Specifically, for the case of $n=16$, prefix nodes $D_{1:0}^{\times 3}, D_{3:0}^{\times 5}$ and $D_{7:0}^{\times 9}$ have a fanout of $3, 5$ and 9, respectively, denoted as a superscript of the corresponding nodes. Superscript of a prefix delay RV with a fanout of two is omitted, *i.e.*, $D_{i:j}^{\times 2}$ is conveniently denoted as $D_{i:j}$.

3.5 Maximum-Delay CDF for Knowles (1,1,1,2) and Knowles (1,2,2,2) Adders

The nominal maximum-delay critical path in an n-bit Knowles adder starts for the LSB position, traverses $\log_2(n)$ prefix stages and ends to the nth bit position. Furthermore, paths to the nth bit position of a Knowles adder present the same binary tree structure as that of a Kogge-Stone adder. Thus, they can be expressed as a matrix-vector multiplication with the matrix $\boldsymbol{A}_n$. The difference between the two is the fanout of the prefix nodes; the fanout of the prefix nodes in a Knowles adder is determined by each specific architecture [30].

The path delay RVs can be expressed as

$$\boldsymbol{D}_k = \boldsymbol{F} + (\boldsymbol{I}_{\frac{m_k}{3}} \otimes \mathbf{1}_{3\times 1})\boldsymbol{D}_{\text{prefix,KN}} + \mathbf{1}_{m_k\times 1} f_{\text{XOR}} \tag{40}$$

where

$$\boldsymbol{D}_{\text{prefix,KN}} = \boldsymbol{A}_n \boldsymbol{G}_{\text{KN}}, \tag{41}$$

$$\boldsymbol{G}_{\text{KN(1,2,2,2)}} = \begin{bmatrix} D_{15:14} \\ D_{13:12} \\ D_{11:10} \\ D_{9:8} \\ D_{7:6} \\ D_{5:4} \\ D_{3:2} \\ D_{1:0}^{\times 3} \\ \hline D_{15:12} \\ D_{11:8} \\ D_{7:4} \\ D_{3:0}^{\times 3} \\ \hline D_{15:8} \\ D_{7:0}^{\times 3} \\ \hline D_{15:0} \end{bmatrix} \begin{matrix} \left.\begin{matrix} \\ \\ \\ \\ \\ \\ \\ \\ \end{matrix}\right\} \text{level 1, leaf nodes} \\ \left.\begin{matrix} \\ \\ \\ \\ \end{matrix}\right\} \text{level 2} \\ \left.\begin{matrix} \\ \\ \end{matrix}\right\} \text{level 3} \\ \left.\begin{matrix} \end{matrix}\right\} \text{level 4, root node} \end{matrix}, \tag{42}$$

$$\boldsymbol{G}_{\text{KN(1,1,1,2)}} = \begin{bmatrix} D_{15:14} \\ D_{13:12} \\ D_{11:10} \\ D_{9:8} \\ D_{7:6} \\ D_{5:4} \\ D_{3:2} \\ D_{1:0} \\ \hline D_{15:12} \\ D_{11:8} \\ D_{7:4} \\ D_{3:0} \\ \hline D_{15:8} \\ D_{7:0}^{\times 3} \\ \hline D_{15:0} \end{bmatrix} \begin{matrix} \left.\begin{matrix} \\ \\ \\ \\ \\ \\ \\ \\ \end{matrix}\right\} \text{level 1, leaf nodes} \\ \left.\begin{matrix} \\ \\ \\ \\ \end{matrix}\right\} \text{level 2} \\ \left.\begin{matrix} \\ \\ \end{matrix}\right\} \text{level 3} \\ \left.\begin{matrix} \end{matrix}\right\} \text{level 4, root node} \end{matrix}, \tag{43}$$

and $\mathbf{G}_{\text{KN(1,2,2,2)}}$ refers to a Knowles (1,2,2,2) adder, $\mathbf{G}_{\text{KN(1,1,1,2)}}$ to a Knowles (1,1,1,2) adder, $\boldsymbol{F}$ is given by (16) for $k = n, m_k = \frac{3n}{2}$, and $\boldsymbol{A}_n$ by (32).

3.6 Maximum-Delay CDF for a Han-Carlson Adder

For the case of an n-bit Han-Carlson adder, we consider path delays that end at the $(n-1)$th bit position. Considering the case of $n = 16$ and only prefix network paths, the path delay vector $\boldsymbol{D}_{\text{prefix,HC}}$ is given by

$$\boldsymbol{D}_{\text{prefix,HC}} = \left[\, D_{\text{HC},1}\ D_{\text{HC},2}\ \cdots\ D_{\text{HC},\frac{n}{2}} \,\right]^T, \tag{44}$$

with

$$D_{\text{HC},1} = D_{13:12} + D_{13:10} + D_{13:6} + D_{13:0} + D_{14:0} \tag{45}$$
$$D_{\text{HC},2} = D_{11:10} + D_{13:10} + D_{13:6} + D_{13:0} + D_{14:0} \tag{46}$$
$$D_{\text{HC},3} = D_{9:8} + D_{9:6} + D_{13:6} + D_{13:0} + D_{14:0} \tag{47}$$
$$D_{\text{HC},4} = D_{7:6} + D_{9:6} + D_{13:6} + D_{13:0} + D_{14:0} \tag{48}$$
$$D_{\text{HC},5} = D_{5:4} + D_{5:2} + D_{5:0} + D_{13:0} + D_{14:0} \tag{49}$$
$$D_{\text{HC},6} = D_{3:2} + D_{5:2} + D_{5:0} + D_{13:0} + D_{14:0} \tag{50}$$
$$D_{\text{HC},7} = D_{1:0} + D_{5:0} + D_{13:0} + D_{14:0} \tag{51}$$
$$D_{\text{HC},8} = D_{14:0}. \tag{52}$$

As $D_{\text{HC},8}$ comprises a single node, the probability of contributing to the maximum-delay path is low, and it is omitted from the computation.

The Han-Carlson prefix network can be conceived as a Kogge-Stone prefix network of nodes placed at odd bit positions only, followed by a row of nodes placed at even bit positions. Therefore, a matrix formulation relying on (32) is exploited for the paths described from (45) to (51). The prefix nodes placed at the odd bit positions refer to a binary tree structure. However, the related tree is not balanced, as a node in the second stage and second bit position is missing. To simplify derivation and construct a fully balanced prefix tree, a dummy prefix node D_{dummy} is introduced in the second stage and second bit position of Han-Carlson adder, which refers to a zero-mean and variance Gaussian delay RV. Specifically, (45) to (51) can be re-expressed as

$$D_{\text{HC},1} = D_{13:12} + D_{\text{tree},1} \tag{53}$$
$$D_{\text{HC},2} = D_{11:10} + D_{\text{tree},1} \tag{54}$$
$$D_{\text{HC},3} = D_{9:8} + D_{\text{tree},2} \tag{55}$$
$$D_{\text{HC},4} = D_{7:6} + D_{\text{tree},2} \tag{56}$$
$$D_{\text{HC},5} = D_{5:4} + D_{\text{tree},3} \tag{57}$$
$$D_{\text{HC},6} = D_{3:2} + D_{\text{tree},3} \tag{58}$$
$$D_{\text{HC},7} = D_{1:0} + D_{\text{tree},4} \tag{59}$$

and

$$D_{\text{tree},1} = D_{13:10} + D_{13:6} + D_{13:0} + D_{14:0} \tag{60}$$
$$D_{\text{tree},2} = D_{9:6} + D_{13:6} + D_{13:0} + D_{14:0} \tag{61}$$
$$D_{\text{tree},3} = D_{5:2} + D_{5:0} + D_{13:0} + D_{14:0} \tag{62}$$
$$D_{\text{tree},4} = D_{\text{dummy}} + D_{5:0} + D_{13:0} + D_{14:0}. \tag{63}$$

The partial path delays $D_{\text{tree},i}$ for $i = 1, 2, 3, 4$ of (60) to (63) refer to a binary tree structure and, hence, are described in a matrix form as

$$\boldsymbol{D}_{\text{tree}} = \boldsymbol{A}_{\frac{n}{2}} \boldsymbol{G}_{\text{HC}}, \tag{64}$$

where

$$\boldsymbol{G}_{\text{HC}} = \begin{bmatrix} D_{13:10} \\ D_{9:6} \\ D_{5:2} \\ D_{\text{dummy}} \end{bmatrix} \Big\} \text{ level 2} \\ \begin{bmatrix} D_{13:6} \\ D_{5:0} \end{bmatrix} \Big\} \text{ level 3} \\ \begin{bmatrix} D_{13:0} + D_{14:0} \end{bmatrix} \Big\} \text{ level 4,5} \tag{65}$$

and $\boldsymbol{A}_{\frac{n}{2}}$ is given by (32). The path delays of (45) to (51) are

$$\boldsymbol{D}_{\text{prefix,HC}} = \boldsymbol{D}_{\text{prefix},1} + \boldsymbol{A}_{\text{HC}}\boldsymbol{D}_{\text{tree}} \tag{66}$$
$$= \boldsymbol{D}_{\text{prefix},1} + \boldsymbol{A}_{\text{HC}}(\mathbf{A}_{\frac{n}{2}}\boldsymbol{G}_{\text{HC}}), \tag{67}$$

where $\boldsymbol{D}_{\text{prefix},1}$ includes nodes of the first prefix stage only,

$$\boldsymbol{D}_{\text{prefix},1} = \begin{bmatrix} D_{13:12} & D_{11:10} & D_{9:8} & D_{7:6} & D_{5:4} & D_{3:2} & D_{1:0} \end{bmatrix}^T, \tag{68}$$

and

$$\boldsymbol{A}_{\text{HC}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{69}$$

The matrix $\boldsymbol{A}_{\text{HC}}$ of (69) is generalized for an n-bit Han-Carlson adder as

$$\boldsymbol{A}_{\text{HC}} = \begin{bmatrix} \boldsymbol{I}_{(\frac{n}{4}-1)} \otimes \mathbf{1}_{2:1} & \mathbf{0}_{(2(\frac{n}{4}-1))\times 1} \\ \mathbf{0}_{1\times(\frac{n}{4}-1)} & \mathbf{1}_{1\times 1} \end{bmatrix}. \tag{70}$$

The complete path delays of an n-bit Han-Carlson adder referring to the nth bit position, including also bit-propagate and generate nodes, are derived as

$$\boldsymbol{D}_k = \boldsymbol{F} + (\boldsymbol{I}_{\frac{m_k}{3}} \otimes \mathbf{1}_{3\times 1})\boldsymbol{D}_{\text{prefix,HC}} + \mathbf{1}_{m_k\times 1} f_{\text{XOR}}, \tag{71}$$

where $m_k = 3(\frac{n}{2} - 1)$ and $\mathbf{F}$ is given by (16) for $k = n-2$,

$$\boldsymbol{F} = \begin{bmatrix} g_{13} & p_{13} & g_{12} & \cdots & p_1 & g_1 & g_0 \end{bmatrix}^T. \tag{72}$$

3.7 Maximum-Delay CDF for a Ladner-Fischer Adder

For the case of an n-bit Ladner-Fischer adder, we consider path delays that end to the $(n-1)$th bit position. For $n = 16$, the path delay vector $\boldsymbol{D}_{\text{prefix,LF}}$,

$$\boldsymbol{D}_{\text{prefix,LF}} = \begin{bmatrix} D_{\text{LF},1} & D_{\text{LF},2} & \cdots & D_{\text{LF},\frac{n}{2}} \end{bmatrix}^T, \tag{73}$$

comprises

$$D_{\text{LF},1} = D_{13:12} + D_{13:8} + D_{13:0} + D_{14:0} \tag{74}$$
$$D_{\text{LF},2} = D_{11:10} + D^{\times 3}_{11:8} + D_{13:8} + D_{13:0} + D_{14:0} \tag{75}$$
$$D_{\text{LF},3} = D_{9:8} + D^{\times 3}_{11:8} + D_{13:8} + D_{13:0} + D_{14:0} \tag{76}$$
$$D_{\text{LF},4} = D_{7:6} + D_{7:4} + D^{\times 5}_{7:0} + D_{13:0} + D_{14:0} \tag{77}$$
$$D_{\text{LF},5} = D_{5:4} + D_{7:4} + D^{\times 5}_{7:0} + D_{13:0} + D_{14:0} \tag{78}$$
$$D_{\text{LF},6} = D_{3:2} + D^{\times 3}_{3:0} + D^{\times 5}_{7:0} + D_{13:0} + D_{14:0} \tag{79}$$
$$D_{\text{LF},7} = D_{1:0} + D^{\times 3}_{3:0} + D^{\times 5}_{7:0} + D_{13:0} + D_{14:0} \tag{80}$$
$$D_{\text{LF},8} = D_{14:0}. \tag{81}$$

We exclude the path delay of (81) from the estimation of maximum-delay CDF. Moreover, a Gaussian delay RV D_{dummy} is introduced in the second row and $(n-3)$th column of the prefix network of a Ladner-Fischer adder. Eqs. (74) to (80) are re-expressed as

$$D_{\text{LF},1} = D_{13:12} + D_{\text{tree},1} \tag{82}$$
$$D_{\text{LF},2} = D_{11:10} + D_{\text{tree},2} \tag{83}$$
$$D_{\text{LF},3} = D_{9:8} + D_{\text{tree},2} \tag{84}$$
$$D_{\text{LF},4} = D_{7:6} + D_{\text{tree},3} \tag{85}$$
$$D_{\text{LF},5} = D_{5:4} + D_{\text{tree},3} \tag{86}$$
$$D_{\text{LF},6} = D_{3:2} + D_{\text{tree},4} \tag{87}$$
$$D_{\text{LF},7} = D_{1:0} + D_{\text{tree},4} \tag{88}$$

with

$$D_{\text{tree},1} = D_{\text{dummy}} + D_{13:8} + D_{13:0} + D_{14:0} \tag{89}$$
$$D_{\text{tree},2} = D^{\times 3}_{11:8} + D_{13:8} + D_{13:0} + D_{14:0} \tag{90}$$
$$D_{\text{tree},3} = D_{7:4} + D^{\times 5}_{7:0} + D_{13:0} + D_{14:0} \tag{91}$$
$$D_{\text{tree},4} = D^{\times 3}_{3:0} + D^{\times 5}_{7:0} + D_{13:0} + D_{14:0}. \tag{92}$$

The partial path delays $D_{\text{tree},i}$ for $i = 1, 2, 3, 4$ of (89) to (92) refer to a binary tree structure and, hence, are described in a matrix form as

$$\boldsymbol{D}_{\text{tree}} = \boldsymbol{A}_{\frac{n}{2}}\boldsymbol{G}_{\text{LF}}, \tag{93}$$

where

$$\boldsymbol{G}_{\text{LF}} = \begin{bmatrix} D_{\text{dummy}} \\ D^{\times 3}_{11:8} \\ D_{7:4} \\ D^{\times 3}_{3:0} \end{bmatrix} \Big\} \text{ level 2} \\ \begin{bmatrix} D_{13:8} \\ D^{\times 5}_{7:0} \end{bmatrix} \Big\} \text{ level 3} \\ \begin{bmatrix} D_{13:0} + D_{14:0} \end{bmatrix} \Big\} \text{ level 4,5}, \tag{94}$$

and $\boldsymbol{A}_{\frac{n}{2}}$ is given by (32). In overall, the paths of (74) to (80) are re-expressed as

$$\boldsymbol{D}_{\text{prefix,LF}} = \boldsymbol{D}_{\text{prefix},1} + \boldsymbol{A}_{\text{LF}}\boldsymbol{D}_{\text{tree}} \tag{95}$$
$$= \boldsymbol{D}_{\text{prefix},1} + \boldsymbol{A}_{\text{LF}}(\boldsymbol{A}_{\frac{n}{2}}\boldsymbol{G}_{\text{LF}}), \tag{96}$$

where $\boldsymbol{D}_{\text{prefix},1}$ is given by (68), and

$$\boldsymbol{A}_{\text{LF}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{97}$$

The matrix $\boldsymbol{A}_{\text{LF}}$ of (97) is generalized for an n-bit Ladner-Fischer adder as

$$\boldsymbol{A}_{\text{LF}} = \begin{bmatrix} \mathbf{1}_{1\times 1} & \mathbf{0}_{1\times(\frac{n}{4}-1)} \\ \mathbf{0}_{(2(\frac{n}{4}-1))\times 1} & \boldsymbol{I}_{(\frac{n}{4}-1)} \otimes \mathbf{1}_{2\times 1} \end{bmatrix}. \tag{98}$$

The complete path delays of an n-bit Ladner-Fischer adder referring to the $(n-1)$th bit position including also bit-wise propagate and generate nodes are derived as

$$\boldsymbol{D}_k = \boldsymbol{F} + (\boldsymbol{I}_{\frac{m_k}{3}} \otimes \mathbf{1}_{3\times 1})\boldsymbol{D}_{\text{prefix,LF}} + \mathbf{1}_{m_k\times 1} f_{\text{XOR}}, \tag{99}$$

where $\mathbf{F}$ is given also by (16) for $k = n-2$ and $m_k = 3(\frac{n}{2} - 1)$.

3.8 Maximum-Delay CDF for a Brent-Kung Adder

An n-bit Brent-Kung adder comprises $2\log_2(n)-1$ prefix stages; we consider path delays that end at the $(n-1)$th bit position. When $n=16$ and considering only prefix network paths referring to path-delay vector $\boldsymbol{D}_{\text{prefix,BK}}$, we have the following path delays

$$D_{\text{BK},1} = D_{14:0} \tag{100}$$
$$D_{\text{BK},2} = D_{13:12} + D_{13:0} + D_{14:0} \tag{101}$$
$$D_{\text{BK},3} = D_{11:10} + D_{11:8} + D_{11:0} + D_{13:0} + D_{14:0} \tag{102}$$
$$D_{\text{BK},4} = D_{9:8} + D_{11:8} + D_{11:0} + D_{13:0} + D_{14:0} \tag{103}$$
$$D_{\text{BK},5} = D_{7:6}+D_{7:4}+D_{7:0}+D_{11:0}+D_{13:0}+D_{14:0} \tag{104}$$
$$D_{\text{BK},6} = D_{5:4}+D_{7:4}+D_{7:0}+D_{11:0}+D_{13:0}+D_{14:0} \tag{105}$$
$$D_{\text{BK},7} = D_{3:2}+D_{3:0}+D_{7:0}+D_{11:0}+D_{13:0}+D_{14:0} \tag{106}$$
$$D_{\text{BK},8} = D_{1:0}+D_{3:0}+D_{7:0}+D_{11:0}+D_{13:0}+D_{14:0}. \tag{107}$$

Given that paths $D_{\text{BK},1}$ and $D_{\text{BK},2}$, given by (100) and (101), respectively, comprise a smaller number of prefix nodes with respect to the remainder of the path delays, *i.e.*, (102) and (107), they can be conveniently neglected from the estimation of maximum-delay CDF.

For the derivation of a matrix transformation for $\boldsymbol{D}_{\text{prefix,BK}}$, it is observed that path delays of (102) to (103) and (104) to (107) correspond to two distinct binary trees. Thus, the transformation matrix $\boldsymbol{A}_{\text{BK}}$ is constructed by two sub-matrices, *i.e.*, $\boldsymbol{A}_8$ and $\boldsymbol{A}_4$, that describe the two binary trees, and a third transformation sub-matrix, *i.e.*, $\mathbf{1}_{6\times 1}$, for the common delay components. The path delays referring to (102) to (107) are reformulated as

$$\mathbf{D}_{\text{prefix,BK}} = \boldsymbol{A}_{\text{BK}}\boldsymbol{G}_{\text{BK}}, \tag{108}$$

where

$$\boldsymbol{A}_{\text{BK}} = \begin{bmatrix} 1&0&1&0&0&0&0&0&0&0&1\\ 0&1&1&0&0&0&0&0&0&0&1\\ 0&0&0&1&0&0&0&1&0&1&1\\ 0&0&0&0&1&0&0&1&0&1&1\\ 0&0&0&0&0&1&0&0&1&1&1\\ 0&0&0&0&0&0&1&0&1&1&1 \end{bmatrix} \tag{109}$$
$$= \begin{bmatrix} \boldsymbol{A}_4 & \mathbf{0}_{2\times 7} & \mathbf{1}_{6\times 1} \\ \mathbf{0}_{4\times 3} & \boldsymbol{A}_8 & \end{bmatrix}, \tag{110}$$

$$\boldsymbol{G}_{\text{BK}} = \begin{bmatrix} \boldsymbol{G}_{\text{BK1}} \\ \boldsymbol{G}_{\text{BK2}} \\ D_{11:0}+D_{13:0}+D_{14:0} \end{bmatrix} \begin{matrix} \} \text{ subtree 1} \\ \} \text{ subtree 2} \\ \} \text{ root node} \end{matrix}, \tag{111}$$

$$\boldsymbol{G}_{\text{BK1}} = \begin{bmatrix} D_{11:10} \\ D_{9:8} \\ D_{11:8} \end{bmatrix} \begin{matrix} \} \text{ level 1} \\ \} \text{ level 2} \end{matrix}, \tag{112}$$

$$\boldsymbol{G}_{\text{BK2}} = \begin{bmatrix} D_{7:6} \\ D_{5:4} \\ D_{3:2} \\ D_{1:0} \\ D_{7:4} \\ D_{3:0} \\ D_{7:0} \end{bmatrix} \begin{matrix} \} \\ \} \\ \} \\ \} \\ \} \\ \} \\ \} \end{matrix} \begin{matrix} \text{level 1} \\ \\ \\ \\ \text{level 2} \\ \\ \text{level 3} \end{matrix}. \tag{113}$$

Algorithm 1 Computation of Vector $\boldsymbol{G}$ and Matrix $\boldsymbol{A}$

```
1:  function COMPUTEGA(CircuitGraph, k, t_th)
2:      paths ← DepthFirstSearch(CircuitGraph, k)
3:      paths ← ExcludeShortPaths(paths, t_th)
4:      G ← BreadthFirstSearch(CircuitGraph, k)
5:      m ← Columns(G)        ▷ number of columns   } construction of A
6:      l ← Rows(paths)       ▷ number of rows      }
7:      A ← 0_{l×m}                                 }
8:      for path_i in paths do                      }
9:          for node_j in G do                      }
10:             if node_j belongs to path_i then    }
11:                 A(i, j) = 1                     }
12:     return G, A
```

For the case of the n-bit Brent-Kung adder, a general expression for matrix $\boldsymbol{A}_{\text{BK}}$ is

$$\boldsymbol{A}_{\text{BK}} = \begin{bmatrix} \boldsymbol{A}_{\frac{n}{4}} & \mathbf{0}_{\frac{n}{8}\times(\frac{n}{2}-1)} & \mathbf{1}_{\frac{3n}{8}\times 1} \\ \mathbf{0}_{\frac{n}{4}\times(\frac{n}{4}-1)} & \boldsymbol{A}_{\frac{n}{2}} & \end{bmatrix}. \tag{114}$$

The complete path delays of an n-bit Brent-Kung adder referring to the $(n-1)$ sum output are derived as

$$\boldsymbol{D} = \boldsymbol{F} + (\boldsymbol{I}_{\frac{m_k}{3}} \otimes \mathbf{1}_{3\times 1})\boldsymbol{D}_{\text{prefix,BK}} + \mathbf{1}_{m_k\times 1} f_{\text{XOR}}, \tag{115}$$

where $m_k = 3(\frac{n}{2}-2)$ and $\boldsymbol{F}$ is provided by (16) for $k = n-\log_2(n)$.

3.9 Algorithmic Derivation of G and A

The introduced models formulate path delays of adders, characterized by a set of fanout, number of wiring tracks and prefix stages combinations, as well as covering extreme design choices [35]. Based on design constraints, it is likely that synthesis tools create prefix structures [34], [38], [39], not covered above. To handle the case of a generic prefix structure and derive path delay formulations, the introduced Alg. 1 is leveraged. For the derivation of $\boldsymbol{D}_{\text{prefix}} = \boldsymbol{A}\boldsymbol{G}$, it suffices to derive $\boldsymbol{A}$ and $\boldsymbol{G}$, and, subsequently, compute $\boldsymbol{D}_k$, given by (15).

Alg. 1 comprises three main steps, *i.e.*, the extraction of paths that end to the kth output, the construction of vector $\boldsymbol{G}$ and the construction of matrix $\boldsymbol{A}$. The path extraction is performed through a depth-first-search traversal on the circuit graph based on the kth output (line 2). The index k can be potentially identified at the final synthesis stages, during timing analysis at the nominal design corner. To reduce the dimensions of matrix $\boldsymbol{A}$, paths with less prefix nodes (or delay) than t_{th} can be excluded, in line 3. The vector $\boldsymbol{G}$ is computed in line 4 by a breadth-first-search traversal from the kth output of circuit graph. Then, the matrix $\boldsymbol{A}$ is constructed in lines 5–11. Notably, the matrix $\boldsymbol{A}$ and vector $\boldsymbol{G}$, derived by Alg. 1, take a different form from that described in Sections 3.3–3.8, in certain adder cases. This is due to the fact that vector $\boldsymbol{G}$, given in Sections 3.3–3.8, may not involve prefix nodes in the same order as $\boldsymbol{G}$, derived by Alg. 1 and based on breadth-first-search traversal, *e.g.*, Brent-Kung adder. Moreover, common path delay components are identified and, thus, compact forms are provided by the derived models. Subsequently, $\boldsymbol{A}$s derived by the two methods may differ. Although $\boldsymbol{A}$s may differ, the derived $\boldsymbol{D}_k$s are essentially equivalent (a

row permutation leads to the same matrices), as they refer to the same set of paths. In [42], Python scripts are provided that derive matrix $\boldsymbol{A}$ for the particular adders using Alg. 1.

While the proposed algorithmic method for the path delay formulation is well-suited for SSTA methods and variation-aware sizing tools [2], the model-based method generalizes easily to long bit lengths, it can be exploited to provide design insights for variation-tolerant architectures, and it can be unified with transistor-level models, such as in [18], to further enhance delay estimation accuracy.

4 Model Evaluation and Discussion

This section quantifies the accuracy of the proposed path-based models for the estimation of worst-case delay of the investigated adders compared to MC simulations. In a first scenario in Section 4.1, we evaluate the effectiveness of the introduced models to capture the maximum delay of the MSB positions. The analysis refers to Kogge-Stone adders for a set of bit-lengths employing a unit delay model and delay variations. A technique is also proposed that increases the accuracy of the estimation for the case of Kogge-Stone adder. In a second scenario in Section 4.2, we compare the introduced CDF models with Spice-level MC-based CDFs for 16-bit parallel-prefix adder topologies. An extensive relative comparison among adders and with prior works is presented. In Section 4.3, the tangible benefits of the introduced models in terms of runtime are evaluated. MATLAB scripts used for simulations can be found in [42].

4.1 Unit Gate Delay Model

The introduced analysis initially focuses on a unit gate delay model to demonstrate how accurately the introduced CDF models capture the maximum-delay CDFs at the MSB position, derived by MC simulations. Moreover, the unit delay model is chosen due to its simplicity; while it ignores electrical and technology characteristics, it determines cell delays relying only on the fanout. Indicatively, the case of a Kogge-Stone adder is demonstrated.

4.1.1 General Case

The agreement of the proposed model with MC simulations is examined for certain bit-lengths of a Kogge-Stone adder. We initially employ a unit gate delay model, where the delay of the ith cell is a Gaussian RV with mean value μ_i and standard deviation $\sigma_i = 0.05\mu_i$. The mean value μ_i is

$$\mu_i = 1 + c_{\text{fanout}}(\text{fanout} - 1), \tag{116}$$

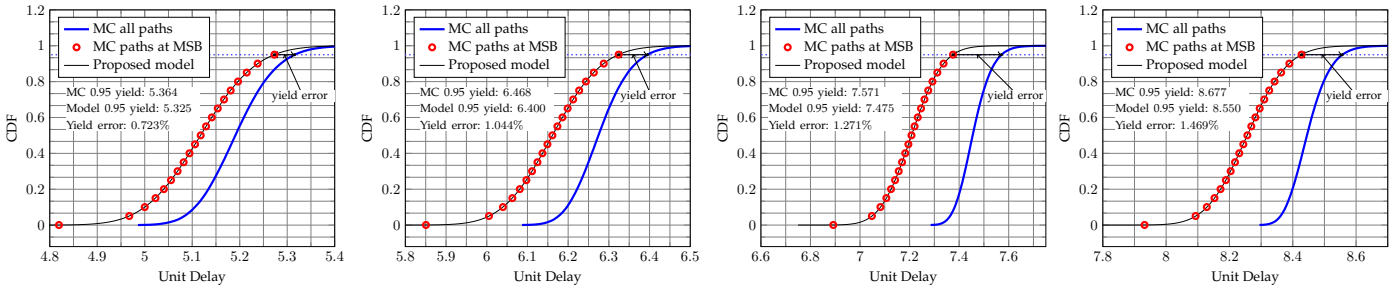
where c_{fanout} is a fanout coefficient that captures the dependency of the delay on the fanout. The fanout coefficient is estimated based on Spice-level simulations for a 16-nm technology node, and equals to $c_{\text{fanout}} = 0.0227$. Delay correlations are assumed zero, denoting an uncorrelated random variation source, such as random dopant fluctuations. The reported data are obtained by simulations of the corresponding circuit graphs, resembling SSTA algorithms [6], [8], and performing 10^6 simulation iterations in MATLAB. Fig. 2 displays the MC-based and the introduced model CDFs for the maximum delay. The MC-based CDF is evaluated for two cases: by measuring the maximum delay of all adder (sum) bit outputs (blue line) and by measuring delays at the MSB of the output only (red marks). The MSB position is captured by the introduced model (black line). Fig. 2 reveals that the proposed path model exactly captures the MC-based CDF of the MSB output for all reported bit-lengths. We also quantify the delay yield at the 0.95 quantile point of the MC-based CDF of all adder outputs and of the introduced model. We define as yield error the difference of 0.95 (0.9987) quantile point of the MC-based CDF (all sum outputs) and that of introduced model. The percentage of yield error between the two is depicted in Fig. 2, with a maximum value of 1.469% for the 64-bit case. As revealed, the percentage yield error increases with an increase of bit-length. This happens because the introduced CDF approximation relies only on paths to the most-significant bit position (or of a neighboring position) only, and misses more paths that contribute to the maximum delay, as the bit-length increases.

We also examine the accuracy of introduced model for the case of correlated delays for a 16-bit Kogge-Stone. Fig. 3 demonstrates the maximum-delay CDFs, where logic cells in $\boldsymbol{F}$ ($\boldsymbol{D}_{\text{prefix}}$) are correlated with a correlation coefficient ρ, while delays between $\boldsymbol{F}$ and $\boldsymbol{D}_{\text{prefix}}$ are uncorrelated. Even under the correlated case, the proposed model exactly captures the delay CDF at the MSB position. Moreover, it is shown that the delay yield error reduces as correlation increases, as the CDF at the MSB position converges to the maximum-delay CDF of all outputs with an increase in correlation. For the case of $\rho = 1$, that is, inter-die variations are manifested, only a single path suffices to describe the maximum-delay behavior when all outputs are considered, as all delays are subjected to identical variations [3].

4.1.2 Extension to multiple end points for Kogge-Stone

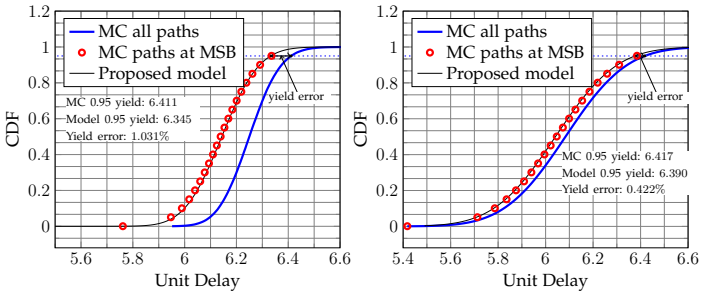
In order to compensate for the yield error of the introduced model for long bit-length adders, we here propose a solution that increases the accuracy of the introduced model to capture the worst-case delay, assessed at high quantiles. The proposed solution relies on the fact that the maximum path delay at the ith output bit position of the Kogge-Stone adder, for $i = \frac{n}{2}, \ldots, (n-1)$, resembles that at the nth bit position. Specifically, under no variations, the mean path delay of ith output bit position, for $i = \frac{n}{2}, \ldots, (n-1)$, reduces as i decreases. This happens as the number of paths to the ith bit position that traverse all prefix stages decrease when i decreases. However, the nominal maximum delay at the bit position i remains the same as that of bit position n. Thus, the proposed matrix-based formulation for the estimation of maximum-delay CDF at the nth sum output can be used as an approximation of the maximum-delay CDF at the ith bit position, for $i = \frac{n}{2}, \ldots, (n-1)$. We define a delay RV vector $\boldsymbol{S}_l = [\boldsymbol{D}_{n-l}, \ldots, \boldsymbol{D}_n]$ that consists of l sets of path delays referring to bit positions $n-l$ to n, with $n-l \leq k \leq n$. We further make the simplifying assumption that paths having as end points different sum outputs do not share common nodes. This assumption significantly reduces the complexity of the matrix-based formulation. For example, a 64-bit Kogge-Stone adder consists of 451 logic paths and 325 prefix nodes. Instead, the proposed model considers only 96 paths and 63 prefix nodes. A transformation that considers all prefix node delays would unnecessarily in-

This article has been accepted for publication in IEEE Transactions on Emerging Topics in Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TETC.2023.3242555



(a) 8-bit (b) 16-bit (c) 32-bit (d) 64-bit

Fig. 2. MC-based (MATLAB) and model maximum-delay CDFs for certain bit-lengths of Kogge-Stone adder. A unit delay model is assumed for the constituent logic cells.



(a) $\rho = 0.1$ (b) $\rho = 0.9$

Fig. 3. MC-based (MATLAB) and model maximum-delay CDFs of 16-bit Kogge-Stone adder. Certain correlation coefficients (ρ) are examined.

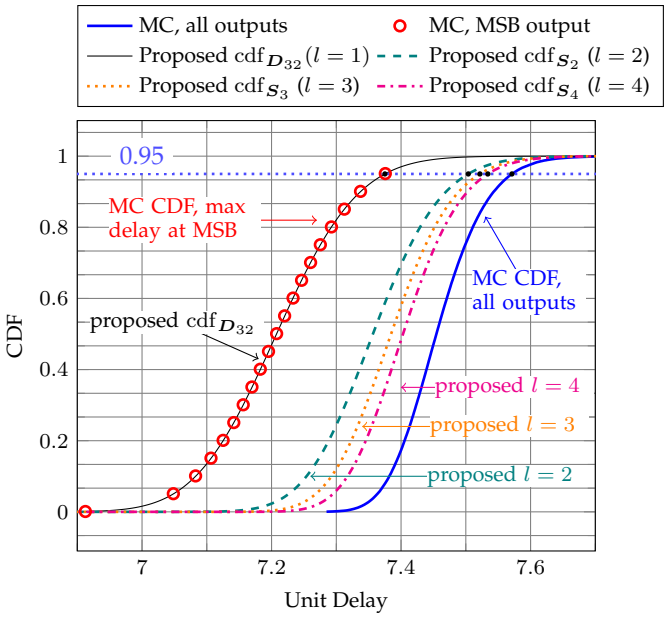


Fig. 4. MC-based (MATLAB) and model CDFs for unit delay model and a 32-bit Kogge-Stone adder.

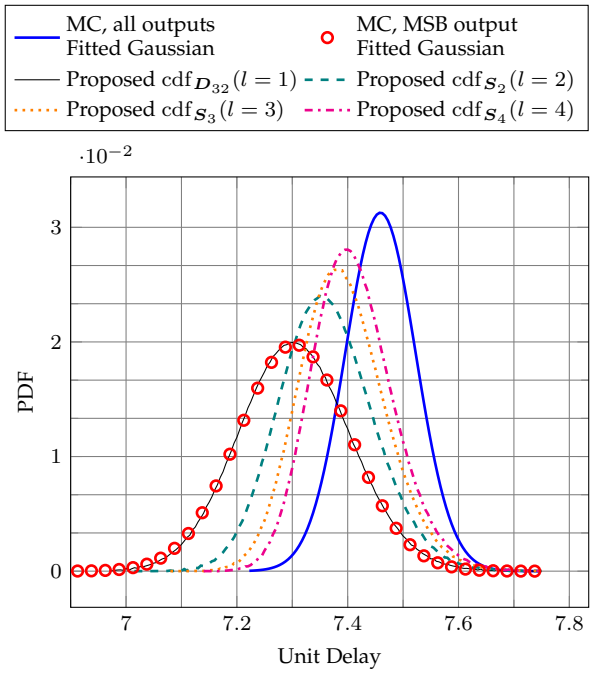


Fig. 5. MC-based (MATLAB) and model PDFs for unit delay model and a 32-bit Kogge-Stone adder.

crease complexity, rendering the technique cumbersome. At the same time, although the extension to multiple end points ignores the structural correlation between paths to different sum outputs, the estimation of worst-case delay by the introduced model extension improves. Specifically, the mean value vector $\boldsymbol{\mu}_{S_l}$ and covariance matrix $\boldsymbol{\Sigma}_{S_l}$ of delay vector $\boldsymbol{S}_l$, required for the evaluation of $\text{cdf}_{\boldsymbol{S}_l}$, are given as

$$\boldsymbol{\mu}_{S_l} = \mathbf{1}_{l\times 1} \otimes \boldsymbol{\mu}_{D_k}, \tag{117}$$

$$\boldsymbol{\Sigma}_{S_l} = \boldsymbol{I}_l \otimes \boldsymbol{\Sigma}_{D_k}, \tag{118}$$

exploiting the mean value vector $\boldsymbol{\mu}_{D_k}$ and covariance matrix $\boldsymbol{\Sigma}_{Dk}$ of $\boldsymbol{D}_k$. Specifically, Fig. 4 and 5 display the maximum-delay CDF and PDF, respectively, for a 32-bit Kogge-Stone adder when considering a certain number of sum outputs, l. It is shown that, as the number of sum bit outputs considered for the construction of the maximum delay CDF increases, the derived CDF more closely approximates the MC-based CDF that refers to all bit outputs, even if the structural correlation of path delays of different outputs

TABLE 1
Percentage of Worst-Case Delay Error at Certain Quantiles Computed by the Proposed Model and the MC Simulations for 32-bit Kogge-Stone

Model	Quantile	
	0.95	0.9987
$\mathrm{cdf}^{*}_{\boldsymbol{D}_{32}}$	1.270820	0.952235
$\mathrm{cdf}_{\boldsymbol{S}_2}$	0.890670	0.600794
$\mathrm{cdf}_{\boldsymbol{S}_3}$	0.606072	0.419540
$\mathrm{cdf}_{\boldsymbol{S}_4}$	0.484102	0.378698

*For $l = 1$, only the $k = 32$nd output is considered and $\mathrm{cdf}_{\boldsymbol{D}_{32}} = \mathrm{cdf}_{\boldsymbol{S}_1}$

is not captured by (118). Furthermore, Table 1 reports the delay yield error, quantified for certain values of l. The error decreases as more outputs are considered for the evaluation of CDF, however, yield error decreases more slowly as the number of output increases. By employing only a small l value, yield error substantially improves. Practically, the actual worst-case delay may not be known, and the error between certain yield points of successive evaluations of $\mathrm{cdf}_{\boldsymbol{S}_l}$ can be used as a cost metric for the convergence to the target worst-case delay, as shown in Fig. 4.

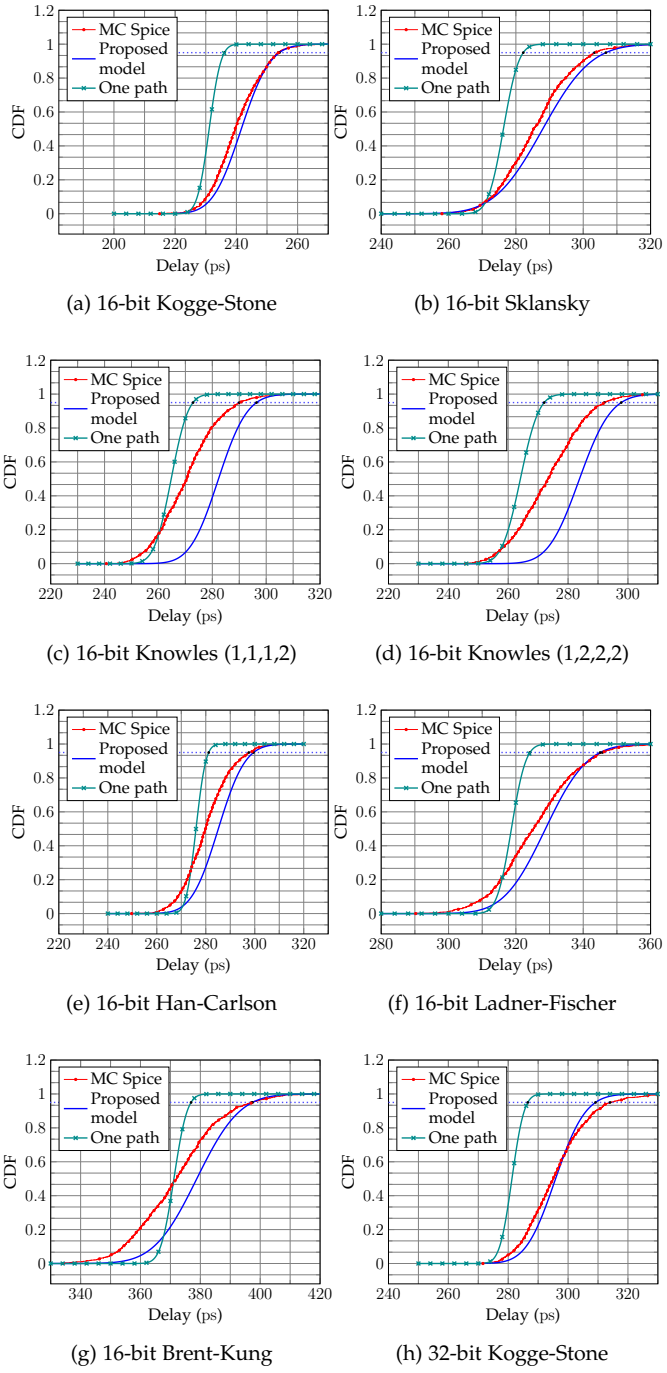
4.2 Comparisons to Spice-level Simulations

4.2.1 Simulation Set-Up

This section presents a quantitative comparison between the introduced path-based models for the approximation of maximum-delay CDF and the MC-based CDF at Spice-level. The comparison is performed between a set of 16-bit parallel-prefix adders. The Spice-level CDF is obtained by performing a set of 1000 iterations of Monte-Carlo simulations using NGSPICE simulator [43].

In the presence of variations, the maximum-delay critical path potentially differs from that of the nominal case. Thus, Spice-level analysis is not restricted only to the sensitization of nominal maximum-delay critical path. Instead, input patterns are also employed that sensitize paths with a high probability to emerge as maximum-delay critical. The corresponding input patterns are applied at each iteration, where a different set of RVs are distributed for the threshold voltage values. The applied input patterns during Spice simulations sensitize also sum outputs different from the kth bit position, *i.e.,* the one where the nominal maximum-delay critical path ends.

The investigated 16-bit parallel-prefix adders are designed at a typical design corner of a 16-nm CMOS technology referring to PTM BSIM-4 transistor models [44]. For the implementation of XOR logic cell at the pre- and post-processing stages of adders, we employ a pass transistor logic design [45]. Furthermore, in order to achieve uniform variations among devices and targeting low area implementations, minimum-size transistors are used for all logic cells. Threshold-voltage variations are applied to CMOS devices to introduce variability in the system, as this is the primary variability contributor in arithmetic circuits [22]. Specifically, threshold voltage values are drawn from a Gaussian distribution with $\mu_{V_{\mathrm{th}}} = V_{\mathrm{th,nominal}}$ and $3\sigma_{V_{\mathrm{th}}} = 0.1V_{\mathrm{th,nominal}}$, where $V_{\mathrm{th,nominal}}$ denotes the nominal threshold voltage value. Corner values of V_{th} range in $[-30\%, +30\%]$ of $V_{\mathrm{th,nominal}}$ [7]; while here a moderate value



(a) 16-bit Kogge-Stone (b) 16-bit Sklansky

(c) 16-bit Knowles (1,1,1,2) (d) 16-bit Knowles (1,2,2,2)

(e) 16-bit Han-Carlson (f) 16-bit Ladner-Fischer

(g) 16-bit Brent-Kung (h) 32-bit Kogge-Stone

Fig. 6. Spice MC-based, proposed model-based, and one-path-based CDFs for certain parallel-prefix adders. Spice-level CDF refers to the maximum delay of all sum outputs. Horizontal lines refer to 0.95 quantile.

of 10% normalized variation is selected during the Spice-level simulations. In the presence of uncorrelated threshold voltage variations, cell delays are uncorrelated and (19) captures this behavior. We focus on uncorrelated delays for two reasons: (a) they lead to the maximum yield error, as the delay distance between the CDF at the MSB position and the maximum-delay CDF of all outputs is maximized (cf. Section 4.1.1); (b) inter-die variations can be compensated for during manufacturing stages [4]. For the employed 16-nm node, the nominal threshold voltage is 620 mV.

The proposed models use worst-case primitive cell de-

This article has been accepted for publication in IEEE Transactions on Emerging Topics in Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TETC.2023.3242555

TABLE 2
Delay at Certain Quantiles for Parallel-Prefix Adders

Bit-length	Adder	0.95				0.9987			
		Spice* (ps)	Model (ps)	Error (%)†	Error (%)††	Spice* (ps)	Model (ps)	Error (%)†	Error (%)††
16-bit	Kogge-Stone	253.6	253.7	−0.006	6.931	268.0	264.6	1.279	10.345
	Sklansky	303.4	306.4	−1.004	6.973	319.9	322.3	−0.747	10.256
	Knowles (1,2,2,2)	292.1	297.8	−1.939	4.974	308.7	310.0	−0.415	8.655
	Knowles (1,1,1,2)	290.2	296.5	−2.163	4.934	306.1	308.8	−0.897	8.415
	Han-Carlson	297.6	299.2	−0.544	5.471	311.4	311.4	−0.014	8.272
	Ladner-Fischer	345.7	344.7	0.296	6.225	364.3	358.5	1.586	9.739
	Brent-Kung	397.2	397.0	0.031	5.067	419.2	412.6	1.583	8.953
	Mean absolute error	–	–	0.856	5.796	–	–	0.932	9.234
32-bit	Kogge-Stone	314.4	309.2	1.648	8.837	339.8	321.0	5.538	14.365

*1000 iterations †Error w.r.t. Spice-level MC-based CDF referring to all sum outputs
††Error of one-path-based measurements ($m_k = 1$) w.r.t. Spice-level MC-based CDF referring to all sum outputs

lays. Specifically, in order to characterize primitive cells in terms of maximum delay mean and standard deviation, MC simulations are performed. The related analysis refers only to the sensitization of nominal maximum-delay critical path in the presence of threshold-voltage variations. Subsequently, the derived statistical metrics are exploited for the delay characterization of cells with the same logic type and fanout. The derivation of $\boldsymbol{\mu}_{D_k}$ and $\boldsymbol{\Sigma}_{D_k}$ based on worst-case cell delays compensates for the fact that only one output is considered for the evaluation of maximum-delay CDF by the introduced models. An alternative policy for the cell delay characterization possibly considers a set of paths and output transitions at the cost of increased runtime. In this work, a policy on only the nominal maximum-delay critical path is selected, minimizing computational runtime. It is important to note that the model accuracy in the Spice-level framework strongly depends on the cell characterization, as logic cells may present different delay, depending on the applied input patterns. In contrast, the model-based analysis for the unit delay model eliminates this dependence.

4.2.2 Simulation Results and Comparative Performance

Fig. 6 depicts the Spice-level MC-based CDF, referring to all sum outputs, and the introduced CDFs ($l = 1$) for certain 16-bit adder topologies. The reported simulation results correspond to the nominal supply voltage, 0.9 V. The worst-case delay of logic circuits in the presence of delay variations is typically assessed at the right tail of the maximum delay distribution as the sum of mean maximum delay ($\mu_{\max}$) and a multiple of standard deviation ($\sigma_{\max}$). At this delay point, the CDF of maximum delay approaches one. As in prior works [6], [11], [23], the worst-case delay is evaluated here in statistical terms and quantified at the 0.95 or 0.9987 quantile of delay CDF. If assumed that the MC-based CDF is Gaussian, delay yield at the 0.9987 quantile corresponds to the $\mu_{\max}+3\sigma_{\max}$ delay. Hence, fitting between the introduced and the simulation-based CDF at high quantile points is of practical interest. Quantitatively, the closest resemblance is achieved for the Kogge-Stone adder, where the delay distance of simulation-based and model CDF is minimized. This case also validates that, indeed, the maximum delay under variations is determined by paths to the MSB position. In some cases, the model CDFs overestimate worst-case delays at high quantiles, as a result of the selection of worst-case delays during the cell delay characterization.

Furthermore, Table 2 reports the model and MC simulation delays at the 0.95 and 0.9987 quantiles. The best fitting for the 0.95 and 0.9987 quantile points is achieved for Kogge-Stone and Han-Carlson adders, presenting an absolute error of 0.006% and 0.014%, respectively. The lowest error for the case of 16-bit Kogge-Stone adder and the 0.95 quantile case is attributed to the fact that the paths ending to the MSB position, considered by the introduced model, have almost identical delay, and the nodes along these paths have the same fanout load. This fact perfectly fits to the selection of the statistical delay metrics for the model evaluation, *i.e.*, only the nominal critical path delay is characterized and the extracted metrics are employed for the model construction. In contrast, the employed cell delay characterization policy does not capture the general delay behavior, under a range of input patterns and sensitized input pins, which leads to worst-case delay errors of almost 2% for the case of Knowles adders. The maximum absolute error in both yield cases does not surpass 2.2% for the investigated 16-bit adders.

Assume the case (cf. [13]) where the maximum delay under variations is evaluated in terms of $\mu_{\text{nom-crit}} + 3\sigma_{\text{nom-crit}}$ (0.9987 quantile) of the nominal maximum-delay critical path. We also consider the case where the maximum adder delay is only determined by the statistical metrics of the nominal maximum-delay critical path under variations. Hereafter, we refer to this set of simulations as one-path-based measurements and it corresponds to $m_k = 1$ in (10). In more detail, Table 2 reports the error between the one-path-based simulations and that of Spice-level data points, referring to all sum outputs. On average, the introduced model reduces delay yield error at the 0.95 (0.9987) quantile by $\times 6.771$ ($\times 9.907$) compared to the one-path-based case. The obtained Gaussian CDF, referring to the one-path-based case, is also depicted in Fig. 6, along with the introduced model-based and simulation-based CDFs. The one-path-based simulations show that they fail to capture the maximum-delay CDF, in contrast to the proposed models.

Furthermore, Fig. 6(h) depicts the model and Spice-level MC-based CDFs for a 32-bit Kogge-Stone, while Table 2 shows that, even for larger than 16 bit-lengths, the delay error is no larger than 5.538%. Research efforts, such as

This article has been accepted for publication in IEEE Transactions on Emerging Topics in Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TETC.2023.3242555

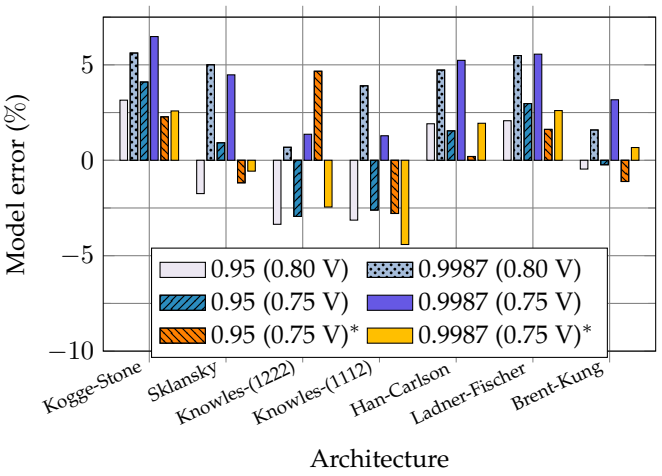
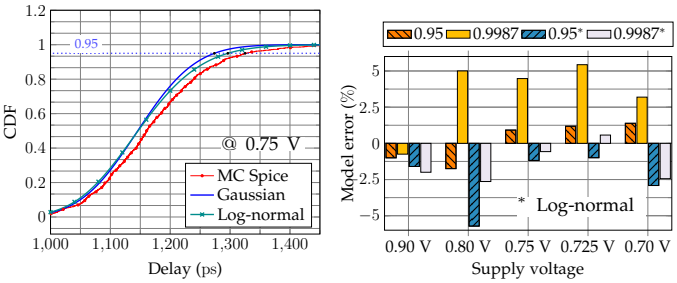


Fig. 7. Error of introduced model for Spice-level simulations at sub-nominal supply voltages. * refers to the multivariate log-normal model.



(a) 16-bit Kogge-Stone CDFs (b) Model error for 16-bit Sklansky

Fig. 8. Gaussian and log-normal model accuracy at low supply voltages.

generic transistor-level variation-aware models in [18] and SSTA algorithms in [6], present an error of 11% and 2.3%, respectively. The method in [25] evaluates the nominal critical-path delay of Knowles adders with a maximum error of 14% with respect to Spice simulations. Hence, the accuracy achieved by the introduced model improves prior art.

Table 2 reveals that the introduced models provide acceptable accuracy for a first-order evaluation of maximum delay for parallel-prefix adders. It can be observed that the introduced models order the investigated parallel-prefix adders in terms of maximum delay as Spice-level MC simulations. The best performance, quantified at the 0.95 quantile, is achieved by Kogge-Stone adder, with a delay margin of 36.6 ps compared to the second fastest Knowles (1,1,1,2) adder. Providing a good compromise between circuit complexity and delay, the delay differences between Knowles (1,1,1,2), Knowles (1,2,2,2), and Han-Carlson are below 10 ps, despite the fact that Han-Carlson comprises an additional prefix stage. Brent-Kung adder presents the worst performance, as comprising the greatest number of prefix stages, and presents a delay margin of 51.5 ps from the second slowest Ladner-Fischer adder. The worst relative performance degradation, given by the ratio $\frac{3\sigma_{\max}}{\mu_{\max}}$, is observed by Knowles architectures. In absolute terms, *i.e.*, $\sigma_{\max}$, the worst performance degradation is observed by Brent-Kung, as it comprises the longest logic paths. Kogge-Stone adder achieves the smallest performance degradation in both absolute and relative terms. This is due to a combination of shortest length logic paths, *i.e.*, four prefix stages, and a small number of common prefix nodes along these paths that average out variability. The simulation results regarding delay spread are consistent with these in [23], which find Kogge-Stone and Brent-Kung at the two extremes. The reported performance metrics indicate that variation mechanisms narrow the delay margins between parallel-prefix adders, compared to the nominal performance, and highlight the need for efficient variation-aware modeling. Notably, compared to the work in [24], the performance ordering of parallel-prefix adders at 16 nm, under threshold voltage variations, is the same as that offered by the introduced models in this work. Furthermore, the performance evaluation in [24] tightly relies on MC simulations; *i.e.*, a set of transient simulations for the nominal maximum-delay combinational path are performed for each architecture, under variations, and the results are utilized for fitting on a Gaussian delay PDF. In contrast, the estimation of maximum delay, proposed here, is based on detailed models that exploit characterized cell delays in the presence of threshold voltage variations. The computational time for the proposed model construction (fourth column of Table 3) is significantly less than that of [24], which needs a mean computational time of 4.5 h for MC simulations.

The accuracy of the introduced model is also evaluated at sub-nominal supply voltages. Fig. 7 depicts the model error with respect to the Spice-level MC simulations and 10% threshold voltage variations. For the case of 0.8 V, the maximum absolute error is 3.355% for the 0.95 quantile and 5.619% for the 0.9987 quantile. At 0.75 V and for the 0.95 quantile, the maximum absolute error is 4.102%, lying in the same range as that at 0.8 V in the general case. The maximum absolute error at 0.75 V and for the 0.9987 quantile slightly increases compared to that at 0.8 V, to reach 6.482%; at this point, the introduced model underestimates the worst-case delay for all adders. For lower supply voltages, the obtained simulation-based maximum-delay distribution demonstrates increasing skewness. To remedy the increased error, especially at the tails of the maximum-delay distribution, a multivariate log-normal distribution is leveraged instead of a Gaussian one in (14), as it has been reported to be suitable for delay modeling at low voltages [9], [11]. The use of log-normal multivariate distribution reduces the errors, compared to the Gaussian, shown in Fig. 7. However, its use at the nominal supply voltage does not offer substantial improvements (see Fig. 8(b) at 0.9 V). The maximum error is observed for Knowles-(1,2,2,2) adder (4.664%), and the minimum for Han-Carlson adder (0.199%) for the two quantiles. Fig. 8(a) depicts the two alternative distributions, where a closer fit of the log-normal distribution compared to the Gaussian one on the simulation measurements is observed at high quantiles. Fig. 8(b) displays the model error for a 16-bit Sklansky adder for several supply voltages, demonstrating a maximum value of 5.435% at 0.725 V.

4.3 Computational Time

Table 3 provides indicative runtime measurements for the estimation of worst-case delay by MC simulations and the introduced model for the 16-bit adder case and the analysis

TABLE 3
Computational Time for the Estimation of Worst-Case Delay by MC Simulations and The Introduced Models

16-bit Architecture	MC Sim.*	Charact.†	Model Constr.
Kogge-Stone	63.06 h	5.18 h	125.50 s
Sklansky	45.17 h	2.21 h	225.10 s
Knowles (1,2,2,2)	54.28 h	2.50 h	146.99 s
Knowles (1,1,1,2)	66.05 h	3.39 h	178.19 s
Han-Carlson	46.26 h	2.26 h	98.08 s
Ladner-Fischer	42.10 h	2.09 h	174.60 s
Brent-Kung	41.24 h	2.07 h	184.29 s
Mean time	51.16 h	2.58 h	161.82 s

* 1000 iterations † Only nominal critical path sensitization

at Spice level. The estimation of worst-case delay by the introduced model is comprised by two processes. The first process is the cell delay characterization. Mean delay and standard deviation of the delay in the presence of threshold voltage variations are derived for each cell, with Spice MC simulations. The cell delay characterization relies on the sensitization of only the nominal maximum-delay critical path, as explained previously. This process leads to the computation of $\boldsymbol{\mu}_{D_k}$ and $\boldsymbol{\Sigma}_{D_k}$. The second process is the model construction, where the worst-case delay is evaluated based on the $\mathrm{cdf}_{\boldsymbol{D}_k}$ and the model description of Section 3. The related time measurements of model construction refer to the runtime of MATLAB scripts. While MC simulations can be accelerated by parallelism, for a fair comparison, reported runtimes refer to the case where no parallelism is applied. The MC simulations are performed on a linux server with a 12-core Intel XEON E-2176G CPU running at 3.7 GHz and 16 GB RAM. The time for the estimation of MC-based $\mathrm{cdf}_{d_{\max}}$ and worst-case delay computation from MC simulation data is negligible compared to the runtime of simulations and not reported in Table 3. The cell delay characterization is performed once per technology generation, and thus, it is not required every time the worst-case delay is evaluated. In a typical EDA flow that exploits commercial-grade standard-cell libraries, statistical metrics of cell delays under variations may be given in the form of Liberty Variation Format (LVF) libraries [12], and, thus, delay characterization does not computationally burden the proposed modeling process. Furthermore, the cell delay characterization is not required for the unit-delay model as described in Section 4.1. Even if the characterization process is not performed every time the worst-case delay is evaluated, a $\times 17.2$ runtime reduction on average is provided compared to the MC simulations according to Table 3.

5 Conclusions

This paper introduces matrix-based formulations that describe delays of paths in parallel-prefix adders with significant probability of becoming maximum-delay critical. The introduced formulations are exploited for the estimation of maximum-delay CDF, based on a multivariate Gaussian CDF, and for the assessment of delay yield at high quantile points. The accuracy of the derived CDF is evaluated at high quantiles compared with a unit delay model and Spice-level MC simulations at a 16-nm node. The analysis compared to unit delay models shows that the model and MC-based CDF at the MSB position perfectly match. Furthermore, a technique is investigated that reduces the error of delay yield, by deriving maximum-delay CDFs when considering multiple adder outputs. Finally, Spice-level MC simulations are performed for 16-bit parallel-prefix adders in the presence of threshold-voltage variations. The quantitative comparison of the MC-based CDF, extracted from Spice-level simulations, and the introduced one shows that the introduced models evaluate the maximum adder delay at the 0.95 quantile with a mean maximum error of almost 1.0% for the 16-bit adders. The introduced modeling framework reduces computational runtime by $\times 17.2$, even if the variation-aware cell characterization is considered as a process of the framework. At the moment, EDA tools rely on delay statistics of the nominal maximum-delay critical path. The comparison of the proposed models with the one-path-based simulation results shows that the proposed models both achieve a closer approximation of the worst-case delay and a greater fitting to maximum-delay MC-based CDFs of all sum outputs. The analytical models introduced could be employed to provide insight to the behavior of general parallel-prefix networks under variability and could be exploited in the formulation of the corresponding design optimization problems, as in [2], taking variability into account. Furthermore, the introduced algorithmic derivation of the model parameters may allow the incorporation of the proposed approach into synthesis frameworks.

References

[1] A. Srivastava, D. Sylvester, and D. Blaauw, *Statistical Analysis and Optimization for VLSI: Timing and Power*. New York, NY, USA: Springer, 2006.

[2] S. M. Ebrahimipour, B. Ghavami, and M. Raji, "A Statistical Gate Sizing Method for Timing Yield and Lifetime Reliability Optimization of Integrated Circuits," *IEEE Transactions on Emerging Topics in Computing*, vol. 9, no. 2, pp. 759–773, 2020.

[3] S. S. Sapatnekar, "Overcoming Variations in Nanometer-Scale Technologies," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 1, no. 1, pp. 5–18, 2011.

[4] M. Orshansky, S. Nassif, and D. Boning, *Design for Manufacturability and Statistical Design: A Constructive Approach*. New York, NY, USA: Springer, 2007.

[5] M. H. Abu-Rahma and M. Anis, "A Statistical Design-Oriented Delay Variation Model Accounting for Within-Die Variations," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 27, no. 11, pp. 1983–1995, 2008.

[6] J. Singh and S. S. Sapatnekar, "A Scalable Statistical Static Timing Analyzer Incorporating Correlated Non-Gaussian and Gaussian Parameter Variations," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 27, no. 1, pp. 160–173, 2007.

[7] T. Liu, C.-C. Chen, and L. Milor, "Comprehensive Reliability-Aware Statistical Timing Analysis Using a Unified Gate-Delay Model for Microprocessors," *IEEE Transactions on Emerging Topics in Computing*, vol. 6, no. 2, pp. 219–232, 2016.

[8] D. Mishagli, E. Koskin, and E. Blokhina, "Path-Based Statistical Static Timing Analysis for Large Integrated Circuits in a Weak Correlation Approximation," in *2019 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, 2019, pp. 1–5.

[9] H. A. Balef, M. Kamal, A. Afzali-Kusha, and M. Pedram, "All-Region Statistical Model for Delay Variation Based on Log-Skew-Normal Distribution," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 35, no. 9, pp. 1503–1508, 2015.

[10] R. R. Rao, A. Devgan, D. Blaauw, and D. Sylvester, "Parametric Yield Estimation Considering Leakage Variability," in *Proceedings of the 41st Annual Design Automation Conference*, 2004, pp. 442–447.

[11] M. Alioto, G. Scotti, and A. Trifiletti, "A Novel Framework to Estimate the Path Delay Variability On the Back of an Envelope via the Fan-Out-of-4 Metric," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 64, no. 8, pp. 2073–2085, 2017.

[12] A. B. Kahng, "New Game, New Goal Posts: A Recent History of Timing Closure," in *2015 52nd ACM/EDAC/IEEE Design Automation Conference (DAC)*. IEEE, 2015, pp. 1–6.

[13] B. Bautz and S. Lokanadham, "A Slew/Load-Dependent Approach to Single-variable Statistical Delay Modeling," in *Proc. Tau Workshop*, 2014, pp. 1–18. [Online]. Available: http://www.tauworkshop.com/2014/Slides/Bautz_SOCV_TAU_2014.pdf

[14] K. Agarwal, M. Agarwal, D. Sylvester, and D. Blaauw, "Statistical Interconnect Metrics for Physical-Design Optimization," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 25, no. 7, pp. 1273–1288, 2006.

[15] T. Huynh-Bao, J. Ryckaert, Z. Tőkei, A. Mercha, D. Verkest, A. V.-Y. Thean, and P. Wambacq, "Statistical Timing Analysis Considering Device and Interconnect Variability for BEOL Requirements in the 5-nm Node and Beyond," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 25, no. 5, pp. 1669–1680, 2017.

[16] E. A. Foreman, P. A. Habitz, M.-C. Cheng, and C. Tamon, "Inclusion of Chemical-Mechanical Polishing Variation in Statistical Static Timing Analysis," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 30, no. 11, pp. 1758–1762, 2011.

[17] T.-B. Chan, S. Dobre, and A. B. Kahng, "Improved Signoff Methodology with Tightened BEOL Corners," in *2014 IEEE 32nd International Conference on Computer Design (ICCD)*. IEEE, 2014, pp. 311–316.

[18] F. Frustaci, P. Corsonello, and S. Perri, "Analytical Delay Model Considering Variability Effects in Subthreshold Domain," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 59, no. 3, pp. 168–172, 2012.

[19] M. Alioto and G. Palumbo, "Impact of Supply Voltage Variations on Full Adder Delay: Analysis and Comparison," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 14, no. 12, pp. 1322–1335, 2006.

[20] K. Papachatzopoulos and V. Paliouras, "Static Delay Variation Models for Ripple-Carry and Borrow-Save Adders," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 66, no. 7, pp. 2546–2559, 2019.

[21] ——, "Low-Power Addition with Borrow-Save Adders Under Threshold Voltage Variability," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 65, no. 5, pp. 572–576, 2018.

[22] K. Bernstein, D. J. Frank, A. E. Gattiker, W. Haensch, B. L. Ji, S. R. Nassif, E. J. Nowak, D. J. Pearson, and N. J. Rohrer, "High-performance CMOS variability in the 65-nm regime and beyond," *IBM Journal of Research and Development*, vol. 50, no. 4.5, pp. 433–449, 2006.

[23] D. Patil, O. Azizi, M. Horowitz, R. Ho, and R. Ananthraman, "Robust Energy-Efficient Adder Topologies," in *2007 IEEE 18th Symposium on Computer Arithmetic (ARITH)*. IEEE, 2007, pp. 16–28.

[24] K. Papachatzopoulos and V. Paliouras, "Maximum Delay Models for Parallel-Prefix Adders in the Presence of Threshold Voltage Variations," in *2020 IEEE 27th Symposium on Computer Arithmetic (ARITH)*. IEEE Computer Society, 2020, pp. 88–95.

[25] Y. Choi and E. E. Swartzlander, "Parallel Prefix Adder Design with Matrix Representation," in *2005 IEEE 17th Symposium on Computer Arithmetic (ARITH)*. IEEE, 2005, pp. 90–98.

[26] G. Dimitrakopoulos and D. Nikolos, "High-Speed Parallel-Prefix VLSI Ling Adders," *IEEE Transactions on Computers*, vol. 54, no. 2, pp. 225–231, 2005.

[27] G. Dimitrakopoulos, K. Papachatzopoulos, and V. Paliouras, "Sum Propagate Adders," *IEEE Transactions on Emerging Topics in Computing*, 2021.

[28] P. M. Kogge and H. S. Stone, "A Parallel Algorithm for the Efficient Solution of a General Class of Recurrence Equations," *IEEE Transactions on Computers*, vol. 100, no. 8, pp. 786–793, 1973.

[29] J. Sklansky, "Conditional-Sum Addition logic," *IRE Transactions on Electronic computers*, no. 2, pp. 226–231, 1960.

[30] S. Knowles, "A family of Adders," in *1999 IEEE 14th Symposium on Computer Arithmetic (ARITH)*. IEEE, 1999, pp. 30–34.

[31] T. Han and D. A. Carlson, "Fast Area-Efficient VLSI Adders," in *1987 IEEE 8th Symposium on Computer Arithmetic (ARITH)*. IEEE, 1987, pp. 49–56.

[32] R. E. Ladner and M. J. Fischer, "Parallel Prefix Computation," *Journal of the ACM (JACM)*, vol. 27, no. 4, pp. 831–838, 1980.

[33] R. P. Brent and H. T. Kung, "A Regular Layout for Parallel Adders," *IEEE Transactions on Computers*, no. 3, pp. 260–264, 1982.

[34] R. Zimmermann, "Non-Heuristic Optimization and Synthesis of Parallel-Prefix Adders," in *proc. of IFIP workshop*. Citeseer, 1996.

[35] D. Harris, "A Taxonomy of Parallel Prefix Networks," in *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*, vol. 2. IEEE, 2003, pp. 2213–2217.

[36] B. D. Lee and V. G. Oklobdzija, "Improved CLA Scheme with Optimized Delay," *Journal of VLSI Signal Processing Systems for Signal, Image and Video Technology*, vol. 3, no. 4, pp. 265–274, 1991.

[37] S. Roy, M. Choudhury, R. Puri, and D. Z. Pan, "Towards Optimal Performance-Area Trade-Off in Adders by Synthesis of Parallel Prefix Structures," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 33, no. 10, pp. 1517–1530, 2014.

[38] R. Roy, J. Raiman, N. Kant, I. Elkin, R. Kirby, M. Siu, S. Oberman, S. Godil, and B. Catanzaro, "PrefixRL: Optimization of Parallel Prefix Circuits using Deep Reinforcement Learning," in *2021 58th ACM/IEEE Design Automation Conference (DAC)*. IEEE, 2021, pp. 853–858.

[39] T.-D. Ene and J. E. Stine, "Point-Targeted Sparseness and Ling Transforms on Parallel Prefix Adder Trees," in *2022 IEEE 29th Symposium on Computer Arithmetic (ARITH)*. IEEE, 2022. [Online]. Available: https://arith2022.arithsymposium.org/program.html

[40] V. G. Oklobdzija, B. R. Zeydel, H. Dao, S. Mathew, and R. Krishnamurthy, "Energy-Delay Estimation Technique for High-Performance Microprocessor VLSI Adders," in *2003 IEEE 16th Symposium on Computer Arithmetic (ARITH)*. IEEE, 2003, pp. 272–279.

[41] *Multivariate Normal Cumulative Distribution Function*, accessed: Jan, 2022. [Online]. Available: https://www.mathworks.com/help/stats/mvncdf.html

[42] *Path-Based Delay Variation Models for Parallel-Prefix Adders*, accessed: July, 2022. [Online]. Available: https://github.com/papachatz/ppa_models

[43] "NGSPICE: Open Source Spice Simulator," http://ngspice.sourceforge.net/.

[44] "Predictive Technology Model," http://ptm.asu.edu/.

[45] H. Naseri and S. Timarchi, "Low-Power and Fast Full Adder by Exploring New XOR and XNOR Gates," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 26, no. 8, pp. 1481–1493, 2018.

Kleanthis Papachatzopoulos (S'16) received the Diploma degree in Electrical and Computer Engineering, and the M.Sc. degree in Integrated Hardware-Software Systems from the University of Patras, Greece, in 2016 and 2018, respectively.

Currently, he is pursuing a PhD degree and he is working as a Research Assistant with the VLSI Design Laboratory, ECE Dept., University of Patras, Patras, Greece. His current research interests include VLSI architectures for signal processing and computer arithmetic.

Vassilis Paliouras (Member, IEEE) is a Full Professor with the Electrical and Computer Engineering Department, University of Patras, Greece. His research interests are in the areas of VLSI architectures for machine learning, signal processing and communications, low-power systems and computer arithmetic. He is advisor to five Ph.D. students, and has supervised four Ph.D., 36 masters', and 40 diploma theses. Prof. Paliouras has received the IEEE CASS Guillemin—Cauer Best-Paper Award for the year 2000. He has served as the General Co-Chair for International Workshop on Power and Timing Modeling, Optimization and Simulation (PATMOS) 2004. He has also served as a Technical Program Chair of PATMOS 2005, the IEEE Workshop on Signal Processing Systems Implementation (SiPS) 2005, and Technical Program Co-Chair of the IEEE International Conference on Electronics Circuits and Systems (ICECS) 2010 and a European liaison for the IEEE ISCAS 2012, South Korea.