Sensitivity to Threshold Voltage Variations of Exact and Incomplete Prefix Addition Trees

Kleanthis Papachatzopoulos and Vassilis Paliouras
Electrical and Computer Engineering Department
University of Patras, Patras, Greece
Email: papachatz@ece.upatras.gr, paliuras@ece.upatras.gr

2022 IEEE International Symposium on Circuits and Systems (ISCAS) | 978-1-6654-8485-5/22/$31.00 ©2022 IEEE | DOI: 10.1109/ISCAS48785.2022.9937629

***Abstract*—Process variations have emerged as a severe performance bottleneck for advanced technology nodes. This paper investigates the delay behavior of a range of exact and incomplete parallel-prefix addition trees under variations, including speculative/approximate trees and trees with duplicated prefix nodes. The performance of a range of incomplete adder variants is investigated comparatively with conventional exact architectures at a 16-nm technology node. In order to capture variations generated from a range of process-dependent sources and their impact on delay characteristics, the analysis considers threshold voltage variations employing Spice-level simulations. Under nominal voltage and in the presence of threshold variations, incomplete architectures are found to still offer a smaller worst-case delay than their exact counterparts; similarly for a low-voltage scenario, where the supply voltage is reduced to 0.8 V. However, focusing on normalized delay variation, it is here found that incomplete-tree adders are more susceptible to variations as they show a wider delay spread than exact architectures around their respective mean values. As a remedy, it is shown that the number of stages in an incomplete adder can be used as a design parameter to investigate accuracy *vs.* variability-tolerance trade-offs. Furthermore, by duplicating delay-critical paths of prefix trees, worst-case delay and standard deviation reduce compared to the exact architectures.**

***Index Terms*—parallel-prefix adders, speculative adders, arithmetic circuits, delay variations, threshold voltage variations**

I. INTRODUCTION

Binary adders and derived arithmetic circuits are required in almost every digital processing system. Along with technology scaling and improved circuit design techniques, novel architectural modifications further reduce adder power dissipation and latency [1], [2]. Several adder architectures have been proposed in the literature targeting a range of applications with diverse power dissipation and performance requirements. In high-speed applications, parallel-prefix adders are employed, such as Kogge-Stone [3], Knowles [4], Sklansky [5], Brent-Kung [6], Han-Carlson [7], and Ladner-Fischer [8] adders, as their latency is $O(\log_2(n))$, where n is the operand bit-width.

Variability in arithmetic circuits is a field of active research as it drives the design decisions for more involved structures as well. A preliminary investigation of the impact of variability sources on adder architectures was conducted by Bernstein *et al.* in [9]. Toledo *et al.* exploits Schmitt Trigger inverters for delay and power variability reduction in full adders [10], accompanied by considerable power and delay overheads. Reis *et al.* investigate the insertion of decoupling capacitor cells at the output of full adders designed in a 7-nm FINFET technology, incurring 20% delay variability reduction at the cost of a moderate 10% delay overhead [11]. Kukner *et al.* investigate the performance degradation of specific parallel-prefix adders under aging effects [12]. Recently, Gaussian distributions have been used to built delay models for parallel-prefix adders under variations that provide sufficient accuracy compared to Monte-Carlo simulations [13].

For beyond 40-nm nodes, mainstream approaches rely on *Liberty Variation Format* and *On-Chip Variation* techniques, requiring a lesser computational effort than *Statistical Static Timing Analysis* (SSTA) algorithms and providing acceptable accuracy [14]. SSTA algorithms model cell delays as probability density functions. However, the difficulty to handle non-Gaussian distributions render them unattractive. Alternatively, Monte-Carlo simulations model cell delays as random variables (RVs) and perform independent timing analysis runs for each set of variables. Handling accurately non-Gaussian distributions makes them attractive at the cost of great computational effort. Correlated RVs are typically handled by approximation methods [15]. As corner-based analysis cannot ensure timing yield in nodes of a few nanometers, Spice-level Monte-Carlo simulations are employed as a statistical timing analysis technique [16].

In this paper, we estimate the delay robustness of specific approximate and speculative (SPEC) parallel-prefix adders under threshold voltage variations. This paper shows that SPEC parallel-prefix adders perform better than exact parallel-prefix adders in terms of a statistical worst-case delay, $\tau_w = \tau_\mu + 3\tau_\sigma$, where τ_μ is the mean maximum delay and τ_σ is the standard deviation of the maximum delay in the presence of variations. However, under variations, specific SPEC adders deliver less acceleration than expected, when τ_w is used as a metric, compared to τ_μ. We also evaluate the normalized delay variation of various adder topologies. While still faster, SPEC adders are found to be more susceptible to variance; and the number of prefix stages is here shown to be a design parameter that can control this sensitivity, yet with an impact on the accuracy of the addition. We also propose the duplication of

This research has been co-financed by the European Union and Greek national funds through the Operational Program Competitiveness, Entrepreneurship and Innovation, under the call RESEARCH-CREATE-INNOVATE (project code: T1EDK-02551).

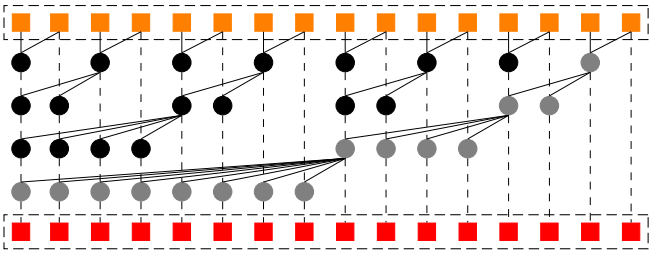

Fig. 1. Propagate-generate network of a 16-bit Sklansky adder. Orange nodes refer to first-stage propagate-generate cells, black to group propagate-generate cells, gray to group generate cells, and red to XOR cells.

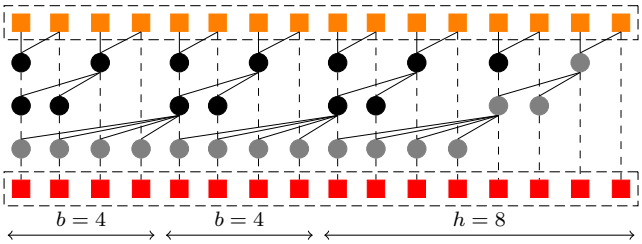

Fig. 2. 16-bit speculative Sklansky adder with three prefix stages [17].

prefix nodes for delay-critical paths to reduce worst-case delay.

The remainder of this paper is structured as follows: Section II presents the main sources of variability considered here and the modeling framework. Section III revisits the basic exact and SPEC parallel-prefix adders. Section IV discusses the delay behavior of a representative set of exact and SPEC parallel-prefix adders. Finally, Section V concludes this paper.

II. Variability Model

As process fluctuations affect cell delays statistically, avoidance of process variations in timing analysis significantly overestimates timing yield. In this work, we abstract process variations at the circuit level, mapping variability to threshold voltage variations. Threshold voltage variations prevail as the main source of variability for addition structures [9], emerging from atomic-scale fluctuations of dopant atoms, known as random dopant fluctuations [18]. Also, other less significant variations, such as line-edge roughness and oxide thickness variations, can be mapped to threshold voltage deviations. Here, we assume that variations are independent and identically distributed Gaussian RVs, with $3\sigma_{V_{\text{th,nom}}} = 0.1V_{\text{th,nom}}$, commonly referred as intra-die variations, where $V_{\text{th,nom}}$ is the nominal threshold voltage value and $\sigma_{V_{\text{th,nom}}}$ its corresponding standard deviation. Inter-die variation components sourcing from process-systematic variations are omitted as they can be compensated during manufacturing stages [16].

This analysis employs Spice-level Monte-Carlo simulations for a precise estimation of maximum delay in the presence of the investigated variations. In a variation-prone framework, deviation of maximum-delay critical path of a circuit from its nominal value becomes possible, as variations greatly affect cell delays. Thus, sensitization of nominal maximum-delay critical path under variations does not guarantee that the actual maximum delay is captured. Considering this, a pre-processing step is necessary to identify paths that are likely to emerge as critical in a variation-prone environment [16]. Our analysis involves input patterns, able to sensitize paths as long as 80% of the nominal maximum-delay critical path, identified under nominal conditions and sensitized under variations.

III. Parallel-Prefix Addition

A. Review of Parallel-Prefix Adders

To avoid serial implementations, carry computation can be formulated as a parallel-prefix problem [19], separating addition in the pre-processing, prefix-processing, and post-processing stages. Assume a pair of n-bit adder operands, $a = a_{n-1}a_{n-2}\dots a_0$ and $b = b_{n-1}b_{n-2}\dots b_0$. The pre-processing stage computes the local generate, $g_i = a_i b_i$, and propagate, $p_i = a_i \oplus b_i$, signals, using bit-wise AND and XOR operations, respectively. Prefix processing relies on the association of group propagate and generate bit pairs, by leveraging the associative operator $\circ$, defined as $(G, P) \circ (G', P') = (G + PG', PP')$. Specifically, denoting the group generate and propagate bits as $(G_{i:k}, P_{i:k})$ referring to bit positions $i, i-1, \dots, k$, consecutive association of bit pairs as $(G_{i:k}, P_{i:k}) = (G_i, P_i) \circ (G_{i-1}, P_{i-1}) \circ \dots \circ (G_k, P_k)$, leads to the computation of carry signal $c_i = G_{i:0}$. Finally, the post-processing stage derives the sum bits at each position as $s_i = p_i \oplus c_{i-1}$.

Exploiting the associativity and idempotence of prefix operator $\circ$ [4], [19], parallel-prefix adder topologies are created that span a wide range of wire track, fanout and prefix stages combinations. Thus, parallelizing addition and implementations of logarithmic logic depth become possible.

B. Incomplete Addition Architectures

Speculative addition relies on the assumption that the sum output can be correct even if the summation function is performed incompletely. This leads to great hardware simplifications and to a much faster addition process, beyond the bound of $O(\log_2(n))$, without compromising quality of results most of the time. For unsigned operands and uniformly distributed numbers at the adder inputs, carry propagation spans only a small fraction of carry chain, since the sum bit in a position depends only on a small number of previous k input bit positions, where $k < n$ [17]. Thus, SPEC adders compute block generate and propagate signals for only a fraction of bit-groups. Specifically, SPEC prefix networks are derived by deleting a number of prefix rows of exact adders. The SPEC architectures employed here can also be utilized for approximate addition [20]–[22] when not supported by error detection and correction [17], [23]. Hereafter, the term SPEC architecture refers to an adder with an incomplete truncated prefix network, a common feature of both speculative and approximate architectures. Indicatively, Figs. 1 and 2 depict the prefix stages of 16-bit exact and SPEC Sklansky adders, respectively. SPEC adders are characterized by the parameters h and b that denote the number of exact sum outputs and the group of outputs with a common prefix node, respectively [17].

IV. Experimental Results and Discussion

A. Synthesis results at a 28-nm FDSOI technology

Since only the SPEC parallel-prefix network determines the maximum delay critical path, the error detection and correction

TABLE I
HARDWARE COMPLEXITY, PERFORMANCE, AND POWER ESTIMATION AT 28-nm FDSOI TECHNOLOGY

Architecture	Speculative			Delay (ps)		Area (μm^2)		Logic cells		Total Power** (nW)		Energy per addition** (fJ)	
	PS*	h	b	Exact	SPEC	Exact	SPEC	Exact	SPEC	Exact	SPEC	Exact	SPEC
Kogge-Stone	3	8	1	241	230	114.893	125.555	253	242	54992.390	59103.658	109.985	118.207
Sklansky	3	8	4	245	218	86.931	101.837	159	188	45588.333	55194.163	91.177	110.388
Ladner-Fischer	4	9	4	262	241	83.341	88.998	153	160	45171.415	48084.218	90.343	96.168
Han-Carlson	4	9	2	258	239	88.672	101.075	181	181	46155.911	54680.445	92.312	109.361
Carry Increment	3	7	4	260	221	84.211	96.179	141	169	45503.436	49088.030	91.007	98.176

*Prefix Stages. **Power and energy estimations refer to a 500 MHz operation frequency and 1 V for architectures synthesized for maximum performance.

TABLE II
DELAY VARIATION METRICS OF 16-BIT PARALLEL-PREFIX ADDERS AT A 16-nm TECHNOLOGY NODE. NOMINAL SUPPLY VOLTAGE AT 0.9 V.

16-bit Architecture	τ_μ (ns)	τ_σ (ps)	τ_w (ns)	τ_σ/τ_μ
Exact Kogge-Stone	0.238	8.456	0.264	0.035 47
SPEC Kogge-Stone**	0.207	9.517	0.236	0.045 87
SPEC Kogge-Stone-2PS	0.151	6.967	0.171	0.046 28
Exact Sklansky	0.285	11.177	0.318	0.039 28
Exact Sklansky-DUPL*	0.269	10.821	0.301	0.040 26
SPEC Sklansky†	0.215	9.526	0.248	0.044 26
SPEC Sklansky-DUPL*	0.207	9.584	0.236	0.044 53
SPEC Sklansky-2PS	0.154	6.764	0.175	0.043 83
Exact Ladner-Fischer	0.324	12.414	0.361	0.038 32
SPEC Ladner-Fischer†	0.262	10.889	0.295	0.041 49
Exact Han-Carlson	0.280	9.546	0.309	0.034 09
SPEC Han-Carlson†	0.258	10.581	0.290	0.040 93
Exact Carry Increment	0.329	13.193	0.369	0.040 05
Exact Carry Incr.-DUPL*	0.310	12.497	0.348	0.040 25
SPEC Carry Increment†	0.211	9.667	0.240	0.045 79

*Introduced in this paper. **Introduced in [23]. †Introduced in [17].

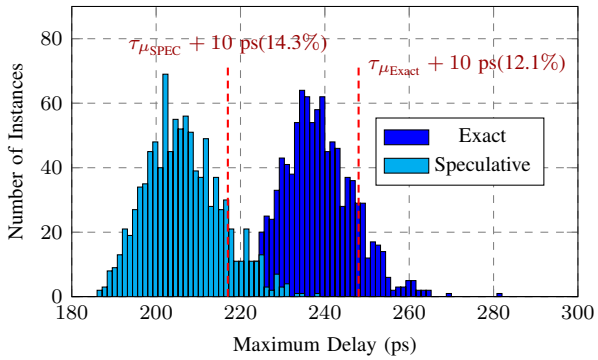

Fig. 3. Maximum delay histograms of 16-bit parallel-prefix Kogge-Stone adder under intra-die threshold voltage variations.

units are excluded from variation analysis. Table I reports the parameters for the design of SPEC parallel-prefix adders.

Table I provides hardware cost and performance synthesis results for exact and SPEC architectures including error detection and correction stages. Results refer to a commercial-grade 28-nm FDSOI technology and maximum performance. Although Kogge-Stone adder is the fastest among exact adders, with 241 ps maximum delay, Sklansky adder is the fastest among SPEC adders, with 218 ps maximum delay. Furthermore, Table I gives the total power and average energy per 16-bit addition. Estimations refer to post-synthesis simulation at 500 MHz and a nominal voltage of 1 V, by back annotated switching activities. Power estimations reveal that SPEC adders consume more power than the respective exact adders, without incurring excessive power overheads, while exact Ladner-Fischer adder is the most energy efficient.

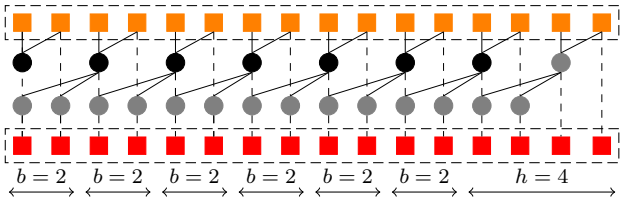

Fig. 4. 16-bit speculative Sklansky adder with two prefix stages.

TABLE III
DELAY VARIATION METRICS OF 16-BIT PARALLEL-PREFIX ADDERS AT A 16-nm TECHNOLOGY NODE. SUPPLY VOLTAGE IS REDUCED TO 0.8 V.

16-bit Architecture	τ_μ (ns)	τ_σ (ps)	τ_w (ns)	τ_σ/τ_μ
Exact Kogge-Stone	0.574	33.765	0.675	0.058 84
SPEC Kogge-Stone	0.485	33.476	0.585	0.069 08
SPEC Kogge-Stone-2PS	0.351	27.898	0.435	0.079 49
Exact Sklansky	0.666	44.377	0.799	0.066 58
SPEC Sklansky	0.502	38.484	0.618	0.076 60
SPEC Sklansky-2PS	0.361	27.218	0.442	0.075 46
Exact Ladner-Fischer	0.761	50.607	0.912	0.066 54
SPEC Ladner-Fischer	0.616	44.723	0.750	0.072 59
Exact Han-Carlson	0.668	39.151	0.786	0.058 57
SPEC Han-Carlson	0.607	40.587	0.729	0.066 86
Exact Carry Increment	0.766	53.490	0.927	0.069 81
SPEC Carry Increment	0.494	38.835	0.610	0.078 66

B. *Variability at nominal-voltage scenario*

Adder architectures are further designed with a 16-nm technology and minimum-sized transistors targeting low-area implementations for 16-bit architectures. Spice simulations are based on BSIM-4 CMOS transistor models [24] and a relatively high threshold voltage library is used to minimize leakage. As a simulation engine, NGSPICE is employed [25]. For the implementation of XOR cells used in the pre- and post-processing stages of adders, a pass-transistor logic design is utilized [26]. Reported statistical delay metrics correspond to 1000 iterations of Monte-Carlo simulations, each one evaluating the maximum delay for a selected set of input patterns.

Table II refers to the nominal voltage of this technology, that is, 0.9 V. Values reported in Table II show that, indeed, SPEC architectures present better worst-case delay performance than exact non-SPEC adders and a Kogge-Stone adder is the fastest. Furthermore, SPEC Kogge-Stone and Han-Carlson architectures present a standard deviation of maximum delay that is greater than that of the corresponding exact architectures. This

essentially means that the delay benefits achieved by SPEC adders over exact counterparts diminish in this case, in terms of the worst-case delay τ_w than the mean maximum delay τ_μ.

In several practical cases, a timing constraint T is a fixed value denoting a time interval for an operation to be completed. In this context, assume a constraint around τ_μ for exact and SPEC architectures. The percentage of instances that violate the constraint around τ_μ is greater for SPEC Kogge-Stone and Han-Carlson architectures than the corresponding exact ones, due to greater τ_σ of the SPEC adders. Thus, it is concluded that SPEC Kogge-Stone and Han-Carlson architectures are more sensitive to threshold voltage variations than the respective exact adders. This is depicted in the two maximum-delay histograms of Fig. 3, for a Kogge-Stone adder. It is shown that the percentage of adder instances that surpass a delay constraint of $T = \tau_\mu + 10$ ps, is greater for a SPEC adder: that is 12.1% and 14.3% for exact and SPEC architectures, respectively. The same conclusion is reached when the ratio τ_σ/τ_μ of Table II is used to measure delay sensitivity to variations. The τ_σ/τ_μ metric shows that SPEC adders experience a greater increase of standard deviation than the corresponding decrease of τ_μ with respect to exact architectures, rendering them more susceptible to delay variations.

C. Variability at low-voltage scenario

A low supply voltage scenario is investigated; the voltage is reduced from 0.9 V to 0.8 V. The statistical metrics in Table III reveal that only the SPEC Han-Carlson architecture presents a marginally greater τ_σ compared to its exact counterpart. Furthermore, in the low-voltage scenario, the difference between the τ_σ/τ_μ ratios increase compared to the nominal supply voltage among SPEC and conventional adders, denoting increased delay variability in low supply voltage. Among the investigated adders, SPEC Ladner-Fischer exhibits the smallest τ_σ/τ_μ difference compared to the exact adders.

While Han-Carlson adder prevails in terms of performance over Sklansky at 0.9 V, as shown in Table II, this situation changes at the low voltage scenario; the effect of the number of fanout nodes of Sklansky adder on the delay is more pronounced at 0.9 V. In parallel, the increased number of prefix stages of a Han-Carlson adder compared to the Sklansky architecture has a greater impact on the delay at low voltage.

D. Architectures with reduced number of prefix stages

Two additional architectures are examined, obtained by limiting the number of prefix stages to only two (2PS), referring to Kogge-Stone and Sklansky structures. Indicatively, Fig. 4 illustrates the investigated Sklansky architecture with two prefix stages. The two-stage architectures are found to present a significantly smaller τ_w than their conventional counterparts, however, at the cost of increased τ_σ/τ_μ ratio. It is noted that SPEC architectures with less prefix stages include paths more likely to become maximum-delay critical than exact architectures. This happens as SPEC architectures contain more paths of equal delays than architectures with a greater number of prefix stages. These paths include fewer common prefix nodes than paths in exact architectures, thus, path delays are substantially less correlated.

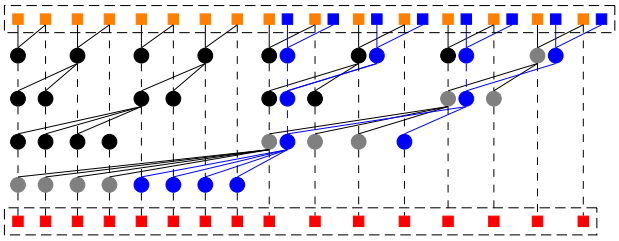

Fig. 5. Sklansky-DUPL: Modified and new nodes are colored blue.

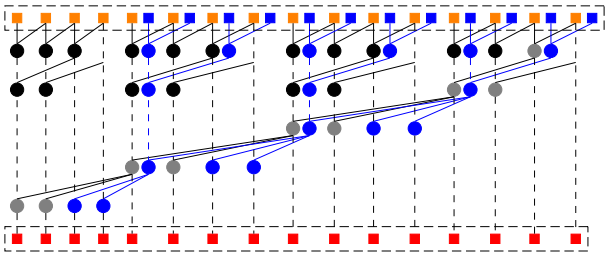

Fig. 6. Carry Increment-DUPL: Modified and new nodes are colored blue.

E. Proposed architectures with duplicated delay-critical paths

In order to control variability and worst-case delay, we modify addition architectures: We duplicate the paths that compute prefix terms of exact adders with a significant number of driving nodes. Specifically, we reduce the number of driving nodes of the last two and three prefix stages for Sklansky and Carry Increment adders, respectively, de-correlating also paths. The new adders, named as DUPL, are shown in Figs. 5 and 6. We assume a minimum-size strategy for CMOS devices to keep variations among devices uniform. We also avoid significant modifications of the prefix structure to maintain regularity. These prefix trees reduce nominal maximum delay as the number of nodes to be driven is reduced. Furthermore, after path duplication, the standard deviation reduces, offering better performance, *i.e.*, 301 ps, *vs.* 318 ps for the exact Sklansky adder, and 348 ps *vs.* 369 ps for the exact Carry Increment adder. SPEC Sklansky-DUPL adder also achieves a reduced τ_w compared to SPEC Sklansky adder, showing that path duplication can be used as a countermeasure to variation sensitivity and reduce τ_w and delay variation. For the 16-bit case, DUPL-based modifications cause 16.3%, 10.5%, and 3.8% area overheads for Sklansky, Carry Increment, and SPEC Sklansky adders at a 28-nm node, respectively.

V. Conclusions

We present a quantitative comparison of SPEC and exact parallel-prefix adders in statistical terms under threshold voltage variations. Specifically, Spice-level Monte-Carlo simulation are performed and the statistical delay metrics referring to the maximum delay are estimated at a 16-nm node. The analysis reveals that specific 16-bit SPEC adders exhibit a greater τ_σ than the respective exact ones. This essentially renders them more susceptible to variations around τ_μ, although having a smaller τ_w. A trade-off between the prefix stages and the normalized delay variation metric is also revealed. Finally, the proposed path duplication achieves to reduce τ_w.

REFERENCES

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

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

[3] 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.

[4] S. Knowles, "A Family of Adders," in *Proceedings 14th IEEE Symposium on Computer Arithmetic (Cat. No. 99CB36336)*. IEEE, 1999, pp. 30–34.

[5] J. Sklansky, "Conditional-Sum Addition Logic," *IRE Transactions on Electronic Computers*, no. 2, pp. 226–231, 1960.

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

[7] 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.

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

[9] 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.

[10] S. P. Toledo, A. L. Zimpeck, R. Reis, and C. Meinhardt, "Pros and Cons of Schmitt Trigger Inverters to Mitigate PVT Variability on Full Adders," in *2018 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, 2018, pp. 1–5.

[11] F. G. R. G. da Silva, C. Meinhardt, and R. Reis, "Mitigation Effects of Decoupling Cells on Full Adders Process Variability," in *2020 IEEE 63rd International Midwest Symposium on Circuits and Systems (MWSCAS)*. IEEE, 2020, pp. 317–320.

[12] H. Kukner, P. Weckx, S. Morrison, P. Raghavan, B. Kaczer, F. Catthoor, L. Van der Perre, R. Lauwereins, and G. Groeseneken, "NBTI aging on 32-bit adders in the downscaling planar FET technology nodes," in *2014 17th Euromicro Conference on Digital System Design*. IEEE, 2014, pp. 98–107.

[13] K. Papachatzopoulos and V. Paliouras, "Maximum Delay Models for Parallel-Prefix Adders in the Presence of Threshold Voltage Variations," in *ARITH*, 2020, pp. 88–95.

[14] 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.

[15] 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.

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

[17] D. Esposito, D. De Caro, and A. G. M. Strollo, "Variable Latency Speculative Parallel Prefix Adders for Unsigned and Signed Operands," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 63, no. 8, pp. 1200–1209, 2016.

[18] Y. Ye, F. Liu, M. Chen, S. Nassif, and Y. Cao, "Statistical Modeling and Simulation of Threshold Variation Under Random Dopant Fluctuations and Line-Edge Roughness," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 19, no. 6, pp. 987–996, 2010.

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

[20] M. Macedo, L. Soares, B. Silveira, C. M. Diniz, and E. A. da Costa, "Exploring the Use of Parallel Prefix Adder Topologies into Approximate Adder Circuits," in *2017 24th IEEE International Conference on Electronics, Circuits and Systems (ICECS)*. IEEE, 2017, pp. 298–301.

[21] F. Ebrahimi-Azandaryani, O. Akbari, M. Kamal, A. Afzali-Kusha, and M. Pedram, "Block-Based Carry Speculative Approximate Adder for Energy-Efficient Applications," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 67, no. 1, pp. 137–141, 2019.

[22] D. Esposito, G. Castellano, D. De Caro, E. Napoli, N. Petra, and A. G. Strollo, "Approximate Adder with Output Correction for Error Tolerant Applications and Gaussian Distributed Inputs," in *2016 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, 2016, pp. 1970–1973.

[23] A. K. Verma, P. Brisk, and P. Ienne, "Variable Latency Speculative Addition: A New Paradigm for Arithmetic Circuit Design," in *Proceedings of the conference on Design, automation and test in Europe*, 2008, pp. 1250–1255.

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

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

[26] 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.