Reduction of delay variations in arithmetic circuits using a redundant representation

Kleanthis Papachatzopoulos and Vassilis Paliouras
Electrical and Computer Engineering Dept.
University of Patras
Patras, Greece

***Abstract*—This paper investigates the impact of inter- and intra-die variations on binary and high-radix adders that adopt the borrow-save encoding. High-radix adders have been employed for the recoding of multipliers and for determining quotient and root digits in iterative division and square-root algorithms. These adders have been found to outperform conventional ripple-carry and carry-skip adders in certain applications, as they implement a carry-free addition algorithm. Carry-free addition using 4-to-2 compressors permits the accumulation of terms in constant time, independent of the word length of the operands. In order to estimate the delay distribution of the aforementioned adders, both Static and Dynamic Monte-Carlo Analysis are employed. Our analysis quantitatively proves that high-radix adders can achieve substantially more narrow delay distributions, i.e., they reduce the delay variance by 98.95% and 99.68% in comparison with 32-bit ripple-carry and carry-skip adders respectively as estimated by DMCA; therefore they are suitable for the implementation of variation-tolerant systems.**

***Index Terms*—High-radix, borrow-save, Monte-Carlo, timing analysis, variation-tolerant design.**

I. Introduction

As VLSI technology scales down in the nanometer dimensions domain, new design challenges emerge due to the impact of process variations on operational characteristics of hardware. The manufacturing-induced fluctuations have rendered ICs prone to timing and power violations, because of the inability of contemporary design methodologies and tools to efficiently model and take into account these variations.

Some of the technology parameters of transistors, most affected from process variations, are gate length, L_{gate}, gate width, W_{gate}, and oxide thickness, t_{ox}. The particular parameters severely impact device threshold voltage, and as a result they affect in an exponential manner the timing and leakage of the device [1]. In addition, environmental conditions and aging can further worsen the timing/power behaviour of the system. The manufacturing defects can be categorised as completely *random* or *systematic*. On the one hand, random fluctuations cannot be controlled. On the other hand, systematic fluctuations can be compensated by the methods of *Design for Manufacturability* [1]. Process variations are statistically modeled as *inter-* and *intra-die* variations. *Inter-die* variations are manifested in the same manner across the chip, while *intra-die* variations have a different value for each device parameter.

Parameter fluctuations introduce great challenges in the design flow. Deterministic verification techniques become obsolete due to fluctuations, because in principle every path is possible to become maximum-delay critical. Recently, novel statistical timing verification techniques have been suggested, the most common of which are statistical static and dynamic timing analysis [2]. In these methods, the delay of each logic element is modeled as a probability density function (PDF). Specifically, static timing analysis considers the worst-case timing behaviour of the design, while dynamic analysis evaluates the timing behaviour of the circuit excited by specific input patterns. Dynamic analysis inherently ignores false paths and therefore provides more realistic delay distributions at the cost of increased simulation time. This is particularly important for circuits that contain false paths, such as carry-skip adders.

In this paper, both Static and Dynamic Monte-Carlo simulations are employed to compare the circuits under study [3]. These methods are used for the investigation of borrow-save encoding over the digit set $\{-1, 0, 1\}$ as a data representation suitable for variation-tolerant circuits. Specifically, we comparatively examine the use of high-radix adders that adopt borrow-save encoding against conventional ripple-carry and carry-skip adders in the presence of inter- and intra-die variations, modeled as Gaussian random variables. Similar analysis applied to different arithmetic circuits has been discussed in [4], [5], where RNS is evaluated as an alternative to binary representation for the design of variation-tolerant arithmetic units. The circuits, on which we focus in this paper, are very important in a variety of practical applications. For example, they can be used as building blocks for multi-operand adders, multipliers, digital filters and discrete transforms. The impact of addition techniques used in these algorithms is of great importance, because they define the critical path delay in many Digital Signal Processing systems.

The remainder of this paper is organised as follows. Section II describes the borrow-save encoding and the corresponding hardware architecture. Section III describes the employed delay model. Section IV introduces and evaluates the results of Static and Dynamic Monte-Carlo Analysis. Finally, Section V concludes the paper.

II. Redundant Arithmetic

A conventional radix-r number system uses digits in the set $[0, r-1]$ to represent numbers, where r is an integer denoting the radix of the representation. In order to speed up the arithmetic operations by restricting the carry propagation, maximally redundant digit sets of the form $[-r+1, \ldots, 0, \ldots, r-1]$ can be used. The necessary condition for arithmetic redundancy is that a radix-r digit set must contain more than r different digits. Redundant digit sets restrict the carry propagation in the operation of addition by concealing it in the redundant representation. Redundant digit sets of the form $[-a, b]$ are called *generalized signed digit (GSD) sets*.

In this paper, we employ the radix-2 number system with digit set $\{-1, 0, 1\}$ alternatively called *binary borrow-save encoding* [6]. Using the borrow-save encoding, each digit value x can be represented as a bit-pair (x^n, x^p) where the left-most bit x^n has negative weight and $x = x^p - x^n \in \{-1, 0, 1\}$. Multi-operand addition of signed digits encoded with the borrow-save representation can be implemented by a tree of carry-save 4-to-2 compressors inverting all the negatively weighted signals [7]. The architecture of a three-digit borrow-save adder is shown in Fig. 1a.

The internal structure of the 4-to-2 compressor can be described by the *carry-free addition algorithm for GSD numbers* and can be divided in three different steps: a) the computation of position sums

978-1-4673-9680-6/16/$31.00 ©2016 IEEE

Authorized licensed use limited to: University of Patras. Downloaded on April 29,2023 at 19:49:34 UTC from IEEE Xplore. Restrictions apply.

$p_i = x_i + y_i$, b) the division of each p_i into a transfer t_{i+1}, and an interim sum $w_i = p_i - rt_{i+1}$, and c) the addition of the incoming transfers to obtain the sum digits $s_i = w_i + t_i$ [8].

The conversion from the redundant to nonredundant representation and inversely involves the costly computation of carry propagation. However, conversions are performed only in the beginning and at the end of a processing sequence and, thus, the delay overhead can be compensated when a sufficient amount of computations are performed in the redundant representation. For this reason, we do not take into account the effect of converters in the delay sensitivity of the circuits under study. Since they are not a part of delay critical computations, their delay variation can be reduced using resizing [9].

III. Timing Analysis and Delay Models

A. Static and Dynamic Monte-Carlo Analysis

To consider on-chip process defects and environmental fluctuations, deterministic verification methods that estimate the critical path of the designed circuits are extended to Statistical Static and Dynamic verification methods in order to capture the timing variability. These methods model timing parameters as statistical variables that follow a specific PDF. In order to take into account the delay variations, two variation-aware timing analysis methods are used, namely Static (SMCA) and Dynamic Monte-Carlo Analysis (DMCA).

In SMCA, each gate delay is modeled as a Gaussian random variable with zero-mean delay and specific variation, with which the impact of inter- and intra-die variations is modeled. This choice for modeling delay variations is common in the literature [1]. Other distributions have also been employed (cf. [10]); the results obtained have been found in agreement with conclusions on the susceptibility of circuits to variations based on the Gaussian assumption. The results reported here refer to simulations of 1500 different circuit instances, each of which is assumed to be implemented on a different die and therefore presents different delay characteristics. SMCA estimates the critical path and provides a upper bound of the delay distribution of the examined circuits, by considering the logic depth of the paths. Notice that false paths may interfere with the estimation, as path sensitization is not considered in SMCA.

In order to circumvent this problem, a dynamic type of analysis is adopted, called DMCA. DMCA captures the impact of the previous state on the delay of a logic cell as a new input pattern is applied on its inputs. Specifically, in the simulation results reported, we consider 100 different circuit instances, each of which is triggered by a uniformly distributed random signal with a variable length of values. Path delays are estimated after propagation of an applied input pattern, taking into account the output values of a previous computation in the overall delay.

B. Employed Delay Models

In this subsection the employed delay models used in SMCA and DMCA are presented. The SMCA model is introduced by Bowman *et al.* [11]. Let D_{ij} be the delay of the i^{th} logic element implemented in the j^{th} die of the test circuit. Delay D_{ij} can be modeled as

$$D_{ij} = D_{\text{nom}}(i) + D_{\text{inter}}(i,j) + D_{\text{intra}}(i,j), \tag{1}$$

where $D_{\text{nom}}(i)$ denotes the nominal delay of i^{th} logic element and $D_{\text{inter}}(i,j)$, $D_{\text{intra}}(i,j)$ represent independent Gaussian variables referred to as inter- and intra-die delay variations respectively. Since all logic elements implemented on the j^{th} die are exposed to identical of inter-die variation, (1) can be simplified to

$$D_{ij} = D_{\text{nom}}(i) + D_{\text{inter}}(j) + D_{\text{intra}}(i,j). \tag{2}$$

As mentioned earlier, we use Static and Dynamic Monte-Carlo Analysis in order to estimate the timing behaviour of the examined adders. In the case of SMCA, the output delay of i^{th} logic component is described by

$$T_i = \max\{T_{i-1}, T_{i-2}, \ldots, T_{i-k}\} + D_{ij}(i), \tag{3}$$

where $T_{i-1}, T_{i-2}, \ldots, T_{i-k}$ denote the output delays of logic elements $i-1, i-2, \ldots, i-k$ respectively that drive the logic element i and D_{ij} is the delay of i^{th} logic element described by (2).

A more elaborate delay model than (3) is used by DMCA, which is a simplified version of the timing model introduced in [12]. The objective of this model is to eliminate false paths by identifying the propagation of a new value through a circuit path and take into account the control value of the logic gates that cause fast output value changes. As an illustrative example, assume a two-input OR gate whose input values are both 0. We assume that both input values are switching from 0 to 1, but not simultaneously. Then the output value becomes 1 and the actual delay of the OR gate is determined by the input that becomes 1 first, since 1 is the control value of OR gate and the switching from 0 to 1 at the second input does not alter the output value. The cases examined are for an OR, NOR, AND, NAND, Full Adder (FA) (for the carry out bit), which has $1, 0, 0, 1, 00/11$ as a control value/s for a $0 \to 1, 1 \to 0, 1 \to 0, 0 \to 1, 1/0 \to 0/1$ output transition respectively.

DMCA describes the total delay T_i at the output of logic cell i as

$$T_i = \begin{cases} \max\{T_{i-1}, T_{i-2}, \ldots, T_{i-k}\} + D_{ij}, & \text{if } \widetilde{g_i} = 1, \widetilde{h_i} = 0 \\ \min\{T_h, T_{h-1}, \ldots, T_{h-k}\} + D_{ij}, & \text{if } \widetilde{g_i} = 1, \widetilde{h_i} = 1, \\ 0, & \text{if } \widetilde{g_i} = 0 \end{cases} \tag{4}$$

where $\widetilde{g_i} = 0$ and 1 denotes the lack or the existence of an output transition for the logic element i respectively, $\widetilde{h_i}$ is 1 when the transition is due to the control value, $i-1, i-2, \ldots, i-k$ is the set of fan-in elements of logic element i and $h, h-1, \ldots, h-k$ is the subset of fan-in elements of logic element i, whose output changes due to a control value.

The nominal delays are derived from actual cell delays of a standard-cell library. A Full Adder is assumed to have a delay for input to carry output $t_{\text{FA},s_{in}\to c_o} = 2t_{\text{gate}}$ and from input to sum output $t_{\text{FA},s_{in}\to s_o} = 2.5t_{\text{gate}}$. Additionally, the nominal delays of AND-OR and XOR gates are $t_{\text{AND-OR,XOR}} = t_{\text{gate}}$, while AND and OR gates have a nominal delay of $t_{\text{AND,OR}} = 0.5t_{\text{gate}}$.

IV. Experimental Results and Discussion

This section discusses the impact of inter- and intra-die variations on the delay sensitivity of a high-radix adders with borrow-save encoding and its dynamic-range-equivalent ripple-carry or carry-skip adders evaluated via both SMCA and DMCA. The architectures of a radix-16 adder and a carry-skip adder with a 4-bit propagate group are depicted in Fig. 1. As we can see, the architecture of the radix-16 adder has a constant delay, that of an 4-to-2 compressor, independently of the word length.

Specifically, Tables I and II quantitatively show the delay variance, σ^2, of the two architectures for a wide range of inter- and intra-die variation values. The value of inter- and intra-die variation denotes the percentage of nominal gate delays used as variance for the corresponding Gaussian random components of (2). Tables I and II reveal that the radix-16 adder with borrow-save encoding outperforms the 4-bit binary ripple-carry adder and the maximum difference of their variance is 75.61% for SMCA and 81.69% for DMCA. Furthermore, comparing results in Tables I and II, we can observe that

Authorized licensed use limited to: University of Patras. Downloaded on April 29,2023 at 19:49:34 UTC from IEEE Xplore. Restrictions apply.

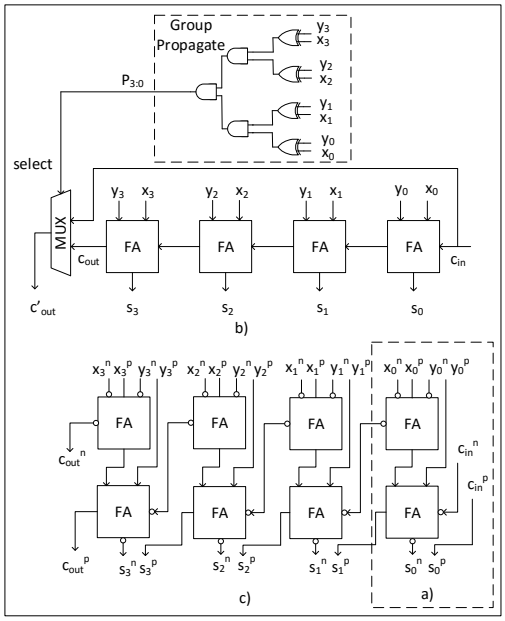

Fig. 1. The architecture of a) a modified 4-to-2 compressor implemented by two Full Adders (FA), b) a part of carry-skip adder with 4-bit groups and c) a radix-16 adder with borrow-save encoding.

TABLE I
SMCA FOR RADIX-16 AND 4-BIT RIPPLE-CARRY ADDER.

inter-die	intra-die	$\sigma^2_{T_{\text{radix-16}}}$	$\sigma^2_{T_{\text{RCA}}}$	Reduction %
0.05	0.05	0.0176	0.0528	66.64
0.05	0.15	0.1246	0.3912	68.15
0.05	0.25	0.2930	1.0786	72.83
0.1	0.05	0.0252	0.0940	73.21
0.1	0.15	0.1353	0.4141	67.33
0.1	0.25	0.3287	1.0479	68.64
0.15	0.05	0.0361	0.1342	73.13
0.15	0.15	0.1471	0.4610	68.10
0.15	0.25	0.3439	1.0207	66.31
0.2	0.05	0.0546	0.1885	71.06
0.2	0.15	0.1667	0.5401	69.14
0.2	0.25	0.3678	1.2313	70.13
0.25	0.05	0.0772	0.3165	75.61
0.25	0.15	0.1827	0.6199	70.53
0.25	0.25	0.3734	1.3273	71.87
0.3	0.05	0.1008	0.3751	73.12
0.3	0.15	0.2207	0.7236	69.49
0.3	0.25	0.3978	1.4949	73.39
0.3	0.3	0.5560	1.6836	66.98

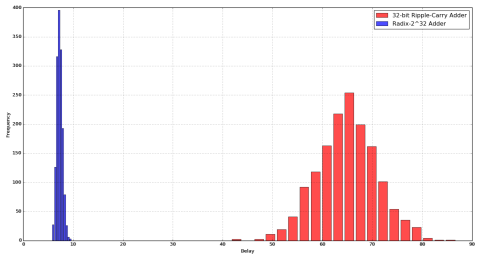

Fig. 2. Delay Distribution of radix-2^{32} adder with borrow-save-encoding and 32-bit ripple-carry adder in case of 0.3 inter- and 0.3 intra-die variations obtained with SMCA.

TABLE II
DMCA FOR RADIX-16 AND 4-BIT RIPPLE-CARRY ADDER.

Signal length		100			1000	10000
inter-die	intra-die	$\sigma^2_{T_{\text{RCA}}}$	$\sigma^2_{T_{\text{radix-16}}}$	Red. %	Red. %	Red. %
0.05	0.05	3.0648	0.6221	79.70	80.86	81.69
0.05	0.15	3.1583	0.8223	73.96	75.29	76.02
0.05	0.25	3.2937	1.2558	61.87	68.20	66.62
0.1	0.05	3.1338	0.6130	80.44	81.21	81.35
0.1	0.15	3.2763	0.8653	73.59	76.27	75.77
0.1	0.25	3.4519	1.2433	63.98	64.35	67.29
0.15	0.05	3.0874	0.6305	79.58	81.29	81.05
0.15	0.15	3.1877	0.8968	71.87	73.91	72.75
0.15	0.25	3.6064	1.1816	67.24	66.77	67.61
0.2	0.05	3.0772	0.6629	78.46	80.96	80.44
0.2	0.15	3.2737	0.8823	73.05	74.99	75.09
0.2	0.25	3.5255	1.2309	65.09	63.65	64.22
0.25	0.05	3.1219	0.6421	79.43	80.41	80.11
0.25	0.15	3.2234	0.8462	73.75	74.28	73.58
0.25	0.25	3.3061	1.2730	61.49	64.35	66.47
0.3	0.05	3.1568	0.7378	76.63	79.17	78.69
0.3	0.15	3.2293	0.8899	72.44	73.47	74.32
0.3	0.25	3.5816	1.3186	63.19	64.82	62.26
0.3	0.3	3.4389	1.4519	57.78	61.28	62.63

TABLE III
DMCA FOR HIGH-RADIX AND RIPPLE-CARRY ADDERS FOR A RANGE OF BIT-LENGTHS.

Signal length			100		
bit-length	inter-die	intra-die	$\sigma^2_{T_{\text{RCA}}}$	$\sigma^2_{T_{\text{high-radix}}}$	Reduction %
8	0.1	0.1	4.5496	0.1807	96.03
8	0.2	0.2	4.8515	0.4223	91.30
8	0.3	0.3	4.6971	0.7962	83.05
16	0.1	0.1	5.3091	0.0778	98.54
16	0.2	0.2	5.8752	0.2535	95.69
16	0.3	0.3	6.2345	0.5318	91.47
20	0.1	0.1	5.6702	0.0642	98.87
20	0.2	0.2	5.5139	0.2546	95.38
20	0.3	0.3	6.2332	0.4573	92.66
24	0.1	0.1	5.8490	0.0741	98.73
24	0.2	0.2	6.0093	0.2484	95.87
24	0.3	0.3	6.4410	0.5167	91.98
28	0.1	0.1	5.9204	0.0629	98.94
28	0.2	0.2	6.2929	0.2089	96.68
28	0.3	0.3	7.2173	0.4187	94.20
32	0.1	0.1	6.0941	0.0638	98.95
32	0.2	0.2	6.3681	0.2161	96.61
32	0.3	0.3	6.7578	0.4413	93.47

Static Analysis always underestimates the value of σ^2. The advantage of high-radix adders is also maintained for wider bit-lengths as shown in Table III, where high-radix adders have equivalent bit-length to ripple-carry adders. The improvements become greater as the bit-length becomes wider, due to the constant delay of the high-radix adders.

Table IV also reports the number of instances of 4-bit RCA and Radix-16 adder the delay of which are above $\mu_{\text{RCA}} + a\sigma_{\text{RCA}}$ delay bound for $a = 1.5, 2$ in case of DMCA. We can conclude that using the borrow-save representation combined with circuits implemented with 4-to-2 compressors we can achieve variation tolerance and avoid costly overdesigning as implied, for example, by resizing [9].

Table V shows the improvement of variance of a radix-2^{32} and a 32-bit carry-skip adder with 4-bit propagation groups under SMCA and DMCA. The improvement is above 93% for both Static and Dynamic Analysis. A carry-skip adder includes by construction false

Authorized licensed use limited to: University of Patras. Downloaded on April 29,2023 at 19:49:34 UTC from IEEE Xplore. Restrictions apply.

TABLE IV
NUMBER OF INSTANCES OF 4-BIT RCA AND RADIX-16 ADDER THE DELAY OF WHICH ARE ABOVE $\mu_{RCA} + a\sigma_{RCA}$ DELAY BOUND FOR $a = 1.5, 2$ IN CASE OF DMCA.

a		1.5			2		
inter-die	intra-die	4-bit RCA	Radix-16	Red. %	4-bit RCA	Radix-16	Red. %
0.05	0.25	8059	2006	75.11	3601	0	100
0.1	0.15	8440	0	100	3804	0	100
0.1	0.25	8445	2417	71.38	3754	257	93.15
0.15	0.15	8158	0	100	4030	0	100
0.25	0.15	8345	269	96.78	4008	0	100
0.25	0.25	8673	1730	80.05	3995	264	93.39
0.3	0.3	7741	2809	63.71	4274	253	94.08

TABLE V
SMCA AND DMCA FOR RADIX-2^{32} AND 32-BIT CARRY-SKIP ADDER.

MC Analysis		Static			Dynamic
Signal length		-			100
inter-die	intra-die	$\sigma^2_{T_{\text{h-radix}}}$	$\sigma^2_{T_{\text{carry-skip}}}$	Red. %	Red. %
0.05	0.05	0.0103	1.6665	99.38	99.68
0.05	0.15	0.0725	4.5351	98.40	97.92
0.05	0.25	0.1740	9.9345	98.25	95.58
0.1	0.05	0.0168	5.9342	99.72	99.52
0.1	0.15	0.0698	8.6307	99.19	98.05
0.1	0.25	0.1835	13.7470	98.66	95.06
0.15	0.05	0.0282	13.7604	99.79	99.24
0.15	0.15	0.0901	14.7442	99.39	97.65
0.15	0.25	0.1898	21.5106	99.12	95.21
0.2	0.05	0.0481	23.4301	99.79	99.16
0.2	0.15	0.1046	27.1787	99.62	97.46
0.2	0.25	0.2005	30.7392	99.35	94.78
0.25	0.05	0.0698	38.5254	99.82	98.91
0.25	0.15	0.1270	35.1549	99.64	97.23
0.25	0.25	0.2246	41.2540	99.46	94.02
0.3	0.05	0.1034	49.0930	99.79	98.51
0.3	0.15	0.1595	56.3197	99.72	96.79
0.3	0.3	0.3029	61.1951	99.51	93.64

paths as the carry chains are interrupted by multiplexers, the operation of which confuses the static technique. As we can see from Figs. 2 and 3, SMCA overestimates the maximum delay of carry-skip in comparison with ripple-carry adder. However, the DMCA delay distributions in Fig. 4 provides more realistic results, i.e., ripple-carry adder exhibits longer delay paths.

V. CONCLUSION

In this paper, we investigate the impact of process and environmental fluctuations on the timing behaviour of certain adder architectures. We employ delay models that use Gaussian random variables for

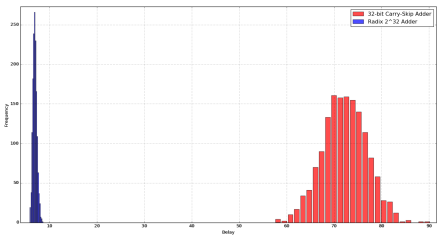

Fig. 3. Delay Distribution of radix-2^{32} adder with borrow-save-encoding and 32-bit carry-skip adder in case of 0.15 inter- and 0.25 intra-die variations obtained with SMCA.

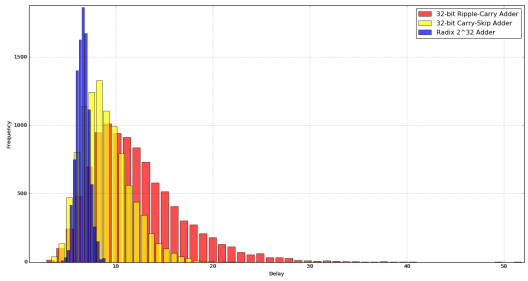

Fig. 4. Delay Distribution of 32-bit carry-skip adder, 32-bit ripple-carry adder and radix-2^{32} adder with borrow-save-encoding in case of 0.3 inter- and 0.3 intra-die variations obtained with DMCA.

modeling variations and examine their effect on the delay sensitivity of the high-radix adder with borrow-save encoding against the conventional binary ripple-carry and carry-skip adders. For the estimation of delay distributions, two types of timing analysis were performed, namely SMCA and DMCA. The corresponding simulations have shown that by adopting the borrow-save encoding, which belongs to the class of *GSD* representations, high-radix adders achieve an improvement of 98.95% and 99.68% in the delay variation compared to 32-bit ripple-carry and carry-skip adder respectively as estimated by DMCA. Thus, the borrow-save encoding is here proposed as a number representation that enhances the delay variation-tolerance of arithmetic circuits for addition.

REFERENCES

[1] M. Orshansky, S. Nassif, and D. Boning, *Design for manufacturability and statistical design: a constructive approach*. Springer Science & Business Media, 2007.
[2] J. Jung and T. Kim, "Variation-aware false path analysis based on statistical dynamic timing analysis," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 31, no. 11, pp. 1684–1697, 2012.
[3] T. McConaghy, P. Drennan, K. Breen, J. Dyck, and A. Gupta, *Variation-Aware Design of Custom Integrated Circuits: A Hands-on Field Guide*. Springer Science & Business Media, 2012.
[4] K. Papachatzopoulos, I. Kouretas, and V. Paliouras, "Dynamic delay variation behaviour of RNS multiply-add architectures," in *IEEE International Symposium on Circuits and Systems (ISCAS), 2016*. IEEE, 2016.
[5] I. Kouretas and V. Paliouras, "Delay-variation-tolerant FIR filter architectures based on the Residue Number System," in *IEEE International Symposium on Circuits and Systems (ISCAS), 2013*. IEEE, 2013, pp. 2223–2226.
[6] P. Kornerup, "Reviewing High-Radix Signed-Digit Adders," *IEEE Transactions on Computers*, vol. 64, no. 5, pp. 1502–1505, 2015.
[7] ——, "Reviewing 4-to-2 adders for multi-operand addition," *Journal of VLSI Signal Processing Systems for Signal, Image and Video technology*, vol. 40, no. 1, pp. 143–152, 2005.
[8] B. Parhami, *Computer arithmetic: algorithms and hardware designs*. Oxford University Press, Inc., 2009.
[9] S. H. Choi, B. C. Paul, and K. Roy, "Novel sizing algorithm for yield improvement under process variation in nanometer technology," in *Proceedings of the 41st annual Design Automation Conference*. ACM, 2004, pp. 454–459.
[10] I. Kouretas and V. Paliouras, "Residue arithmetic for designing multiply-add units in the presence of non-gaussian variation," in *IEEE International Symposium on Circuits and Systems (ISCAS), 2012*. IEEE, 2012, pp. 1231–1234.
[11] K. Bowman, S. G. Duvall, and J. D. Meindl, "Impact of die-to-die and within-die parameter fluctuations on the maximum clock frequency distribution for gigascale integration," *IEEE Journal of Solid-State Circuits*, vol. 37, no. 2, pp. 183–190, 2002.
[12] R. Garg and S. P. Khatri, "Sensitizable Statistical Timing Analysis," in *Analysis and Design of Resilient VLSI Circuits*. Springer, 2010, pp. 131–151.

Authorized licensed use limited to: University of Patras. Downloaded on April 29,2023 at 19:49:34 UTC from IEEE Xplore. Restrictions apply.