

Regular paper

Low-power and variation-aware approximate arithmetic units for Image Processing Applications

Mohammad Mirzaei^a, Siamak Mohammadi^{a,b,*}

^a Dependable System Design Lab., School of Electrical and Computer Engineering, University of Tehran, Tehran, Iran

^b School of Computer Science, Institute of Fundamental Sciences (IPM), Tehran, Iran

ARTICLE INFO

Keywords:

Imprecision-tolerant applications
Approximate computing
Approximate full adder
Variation-aware method
Image processing applications

ABSTRACT

In applications such as image processing and machine learning, imprecision can be tolerated because of the nature of the application itself or the limitation of human senses. By using the approximate computation in parts of imprecision-tolerant applications, where the output quality can be slightly degraded, significant power, delay, or area reductions can be achieved. In this paper, three approximate full adders with reasonable accuracy, low power, and low delay are proposed. The effects of die-to-die (D2D) process variation on the threshold voltage of approximate full adders have been evaluated, and a method has been proposed to reduce the effects of variability. For evaluating the accuracy and the variability, these approximate full adders have been used and analyzed in the ripple carry adder structure and image Sharpening algorithm. In terms of power-delay-product (PDP), accuracy, and area for uniformly distributed inputs, one of the presented approximate full adders exhibits the best performance, and another one shows the best peak-signal-to-noise ratio (PSNR) for real images.

1. Introduction

Although recent semiconductor designs use low-power methods to optimize power consumption, portable devices must be high energy-efficient for computer-intensive multimedia processing due to limited capacity of batteries. Applications such as image and signal processing, multimedia, and computer vision are less sensitive to output quality because they either perform most of the computation on image, audio, and video or deal with inaccurate human senses. These applications are called imprecision-tolerant. In these applications, significant area and power reduction and performance improvement can be achieved by using approximate computing via slight output quality reduction [1]. For example, the k-means clustering algorithm can reduce energy by about 50X, while reducing the classification accuracy by 5% [2]. In recent years, various approaches using approximate computing at different design levels have been proposed. Some of these approaches include approximate accelerators [3], approximate computing units [4], approximate Instruction Set Architecture (ISA) [5] and tunable kernels [6].

In the binary computing system, the adder is an essential computing unit used to perform operations such as addition, subtraction, multiplication, and division. Also, most of the imprecision-tolerant

applications use computational units such as approximate adders and multipliers. The main component of the structure of approximate adders and multipliers is the approximate full adder. Therefore, providing approximate full adders with high energy efficiency and appropriate accuracy is a basic need to achieve high energy efficiency and output quality in imprecision-tolerant applications. For this reason, in this paper, approximate full adders have been considered [7].

As technology progresses, during the semiconductor manufacturing process, factors such as lithography and lens defects cause changes in transistor and interconnections parameters. These physical changes will modify the electrical characteristic of the transistor, such as current and threshold voltage. Using such transistor in a circuit causes changes in the power and delay of the circuit. In this case, variability becomes one of the challenges of designers, so that in technologies below 45 nm, the crucial factor for unreliability is variability [8,9]. Variability is assessed in different fields and for many applications such as in network-on-chips [10], many cores [11], neuromorphic computing [12], smart phones [13], analog design [14], high speed communications [15,16], and biomedical applications [17]. Thus, the importance of variability is undeniable and must be considered in all electronic devices and integrated circuits as a critical task. Imprecision-tolerant applications use approximate adders to reduce power and delay, but variability affects

* Corresponding author at: Dependable System Design Lab., School of Electrical and Computer Engineering, University of Tehran, Tehran, Iran.

E-mail addresses: mo.mirzaei@ut.ac.ir (M. Mirzaei), smohammadi@ut.ac.ir (S. Mohammadi).

the power and delay of these circuits. Therefore, the effects of variability on the approximate adders should be investigated and diminished.

As a result, in modern technologies, variability is significant, but most of the works done in the field of approximate computing have not evaluated the effects of variability on the proposed methods. In this paper, the effects of the process variation on approximate full adders will be evaluated. Considering the results of [9,18] and the importance of threshold voltage (V_{th}) in variability, in this paper, the effect of Die-to-Die (D2D) V_{th} variation on the performance of approximate full adders will be investigated.

Our significant contributions in this paper are as follows:

- (1) Three new approximate full adders called AFA4, AFA5, and AFA6 are proposed. AFA5 is better than the existing approximate full adders in terms of delay, area, and PDP. AFA4 and AFA6, compared to their existing counterparts, exhibit fewer errors.
- (2) For approximate full adders, the effects of variability on power, delay, and PDP are evaluated. For this purpose, the impact of D2D variability on the transistor threshold voltage in 32 nm technology has been assessed using Monte Carlo simulation in the HSPICE environment.
- (3) Also, a method is proposed to reduce the variability of approximate full adders, which also reduces the power consumption.

The rest of the paper is as follows: Section 2 describes previous work on approximate adders and full adders. The following section describes the proposed approximate full adders. Section 5 evaluates the performance, error, and variability effects of approximate adders and offers a method to reduce the impact of variability. In the next section, the simulation results on image processing applications will be presented. Finally, we conclude the paper.

2. Previous work

Paper [19] presented a precision adjustable approximate adder. A full adder and a half adder with maskable carry are proposed and used in the structure of a ripple carry adder (RCA). For setting the accuracy, an error correction circuit is designed, which is activated if needed. In [20], an approximate adder called LOA is presented which uses the Or2 gate for the approximate part to compute output Sum, and uses an And2 gate for the most significant approximate bit to calculate the carry output injected to the exact part. If we only use this adder structure for an approximate full adder, output Sum will be produced by Or2 and output Cout by And2.

A method for finding an energy-efficient hybrid approximate adder for image and video processing applications is presented in [21]. The purpose is to perform multiplications without using traditional multipliers, and thus the primary operations in this work are done by shift-and-add by using parallel prefix adders. In [22], the probabilistic error analysis of approximate adders is considered, and in [23] an efficient method is presented for calculating statistical errors of the block-based approximate adder. A configurable approximate carry look-ahead adder is proposed in [24]. This adder operates in two precise and approximate modes and has been evaluated in 15 nm FinFet technology. An approximate discrete cosine transform (DCT) for image compression using approximate full adders is introduced in [25]. It eliminates the floating-point multipliers and uses the integer addition and shift instead.

In [26] a method to extract parametrized error models for various approximate adders has been proposed. Based on this model, signal processing applications are optimized in terms of accuracy and efficiency. In [27] an approach for computing error parameters in linear time has been presented, as well as a model to extract low power approximate adders. In [28] an approximate full adder based on static CMOS technology has been proposed. Two 5:2 and 4:2 approximate compressors using truth table simplification have been proposed in [29] which can be used in approximate multipliers. Their implementation is

done in 7 nm FinFET technology. An approximate multiplication algorithm based on Karatsuba multiplication method has been proposed in [30]. It has shown better results in terms of accuracy and delay compared to approximate Wallace Tree multiplier.

By eliminating some transistors of a mirror full adder in [31,32], 3 and 4 approximate full adders are presented, respectively. Due to the reduced switching capacitances, these approximate full adders have lower power and delay compared to the exact mirror full adder. Also, these approximate full adders have lower areas; however, they produce incorrect outputs in several cases. Three approximate XOR and XNOR based full adders are introduced in [33] using pass-transistors to implement the gates. Three inaccurate full adders using basic gates are proposed in [34], but used pass-transistors to implement basic gates. In the previous two works, in addition to inaccurate outputs for some inputs, the problem of threshold drops due to pass-transistors is seen [35].

In [36], two approximate full adders made of XOR and MUX based on transmission gates logic are presented that do not have the threshold drop problem of pass-transistors but compared to other approximate full adders have more power consumption. In [37], an approximate half adder, an approximate full adder, and an approximate 4:2 compressor for the array multiplier are presented. In the approximate full adder, to create output Sum, an OR gate has been employed instead of one of the XOR gates. For the carry, AND and OR gates have been used. In [38], an approximate half adder and an approximate full adder using NAND2 gates are presented. Based on the paper's results, this approximate full adder is acceptable in terms of energy consumption; however, it ranks among worse designs in terms of error. In paper [39], authors introduce three approximate full adders and evaluate the impacts of variability on the approximate full adders but do not provide a solution to reduce the effects of variability.

All approximate full adders presented in the articles mentioned above are different in terms of power consumption, delay, accuracy and area, and none of them considered the impacts of variability except the paper [39]. Our goal is to evaluate approximate full adders in terms of performance (power, delay and PDP), error and variability effects, and also propose a method to reduce the effects of variability.

3. Proposed approximate full adders

In an approximate full adder with three binary inputs, eight output combinations are possible. With the increase in the number of incorrect output states, the complexity of the design and power consumption decreases, and with the decrease in the number of incorrect output states, power, accuracy, and complexity of the design increases. Given the existing approximate full adders and the evaluations performed in this paper, there will be a good trade-off between accuracy, design complexity, and power consumption, when two of the eight possible outputs are incorrect. In this paper, all cases where an approximate full adder could have two incorrect output states out of 8 possible ones have been evaluated. To do this, an 8-bit RCA assuming $Cin_0 = 0$ is considered and approximate full adders are used in its structure.

By applying all possible input states to the RCA, the best approximate full adders have been explored in terms of error parameters. The result is two approximate full adders namely approximate full adder 4 (AFA4), and approximate full adder 6 (AFA6), which are better than the existing approximate full adders in terms of error parameters. The truth tables of these full adders are presented in Table 1.

Based on Table 1, the approximate full adder AFA4 for two combinations of $ABCin = 011$ and $ABCin = 101$, exhibits errors at the outputs Sum and Cout. Logic equations of AFA4 outputs are shown in (1). Based on its truth table, the approximate full adder AFA6 for two combinations $ABCin = 011$ and $ABCin = 101$ exhibits errors at the output Sum, however, Cout only has errors for $ABCin = 011$. Therefore, for real applications in terms of error, AFA6 outperforms AFA4 (explained in Section 6). Eq. (3) shows AFA6 outputs. As AFA4 and AFA6 are similar to AMA1 and TGA2 in terms of error numbers and conditions, hence, for

Table 1

Truth table and error parameters of precise and approximate full adders.

Inputs			CMA	AMA1	AMA2	AMA3	VAFA	NFAx	TGA2	LOA	AFA1	AFA2	AFA3	AFA4	AFA5	AFA6
A	B	Cin	CS	CS	CS	CS	CS	CS	CS	CS	CS	CS	CS	CS	CS	CS
0	0	0	00	00	01*	01*	00	01*	00	00	01*	00	00	00	01*	00
0	0	1	01	01	01	01	01	10*	01	00*	01	00*	00*	01	01	01
0	1	0	01	10*	01	10*	01	01	10*	01	01	01	01	01	01	01
0	1	1	10	10	10	10	10	10	10	01*	01*	11*	01*	01*	01*	01*
1	0	0	01	00*	01	01	01	01	10*	01	01	01	01	01	01	01
1	0	1	10	10	10	10	10	10	10	01*	10	11*	11*	01*	01*	11*
1	1	0	10	10	10	10	01*	11*	10	11*	10	11*	11*	10	10	10
1	1	1	11	11	10*	10*	10*	11	11	11	10*	11	11	11	10*	11
ER			-	0.25	0.25	0.375	0.25†	0.375	0.25	0.5	0.375	0.5	0.5	0.25†	0.5	0.25†
MRED			-	0.25	0.167	0.292	0.104†	0.312	0.25	0.312	0.229	0.312	0.312	0.125‡	0.292	0.125‡
NMED			-	0.083	0.083	0.125	0.083†	0.125	0.083	0.125	0.125	0.167	0.167	0.083†	0.167	0.083†

their implementation we will limit power, delay and PDP so that their values are less than the average values of AMA1. Therefore, we will only consider designs with the average power less than $4.24 \mu W$, average delay less than 18.73 ps and average PDP less than 82.21 aj. Finally, among the designs that satisfy the constraints, we select the design which has the least power, delay, and PDP for AFA4 and AFA6. When fewer transistors are connected in series, node capacitors are charged and discharged faster reducing the delay. Thus in this paper, in all proposed approximate full adders transistors in series have been used as less as possible compared to CMA design. In addition, AFA designs have used less transistors compared to CMA causing switching activity and load capacitance to be reduced, which have led to dynamic power (and area) reduction [32].

Proposed Approximate full adders transistor-level schematics are shown in Fig. 1. To implement output Sum in the approximate full adders AFA4 and AFA6, a structure similar to a mirror adder has been used (see Fig. 2-a). In AFA4, AND gates have been employed to produce Cout. This will shorten the carry chain in structures such as an RCA, significantly reducing the delay in an add operation. Thus, with few transistors, low-power approximate full adders AFA4 and AFA6 have been designed at the transistor level. Looking closer at the CMA truth table, in 6 out of 8 possible states for the outputs, Sum equals Cout. Using this fact as well as LOA idea [20], the approximate full adder 5 (AFA5) has been proposed, where its Cout is similar to that of LOA, but uses the Nand2 gate instead of Or2 to produce Sum. LOA uses one And2 and one Or2, whereas only one And2 (Nand2 + Not) has been used in AFA5 resulting in power consumption and delay reduction. AFA5 has the lowest power, delay, PDP, and area among approximate full adders. The approximate full adder AFA6 is the improved version of AFA4 in terms of Cout, in the way that by reducing the existing errors at output Cout, AFA6 outperforms AFA4 when carry chain is long. Also, AFA6 performs better for image processing real input applications. For the hardware implementation of Sum in AFA6, which is similar to AFA4, CMA circuit (Fig. 2-a) has been used by removing some transistors from it. By simplifying the truth table, Cout in AFA6 equals $A(B + Cin)$. For its

implementation CMOS logic has been utilized. The logical equations of these proposed approximate full adders are as follows:

$$AFA4 : Cout = AB, Sum = A\bar{B} + \bar{A}B + Cin \quad (1)$$

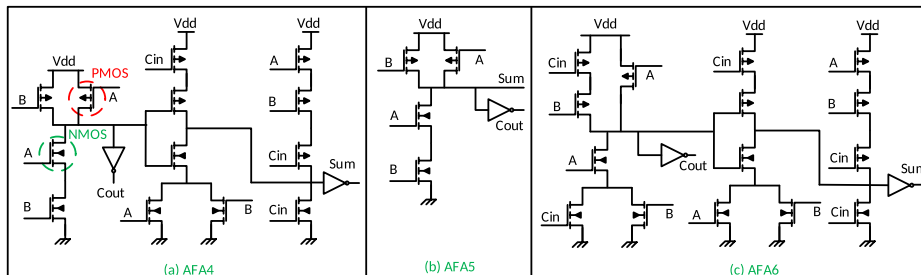
$$AFA5 : Cout = AB, Sum = \bar{A}\bar{B} \quad (2)$$

$$AFA6 : Cout = A(B + Cin), Sum = A\bar{B} + \bar{A}B + Cin \quad (3)$$

As seen in Fig. 2, input capacitance of Cin in CMA includes 6 gate capacitances, whereas for AFA4, AFA5 and AFA6 this amounts to 3, 0, and 5, respectively (Fig. 1). Similarly, each of A and B inputs in CMA includes 8 gate capacitances, whereas for AFA4, AFA5 and AFA6 this amounts to 4, 2, and 4, respectively. Therefore, in all proposed approximate full adders as the load capacitance has been reduced, the delay propagation and dynamic power have decreased. Thereby, for low power designs, one can keep delay constant at the same level as in CMA by reducing the supply voltage to such an extent the delay constraints remain satisfied, and in return achieve a significant reduction in dynamic and static power.

According to Eqs. (1) and (2), AFA4 and AFA5 do not require carry from the previous stage to calculate Cout, and therefore, the critical path delay of the adders using these approximate full adders is lower. Table 1 compares all of the above full adders in terms of precision for different inputs. The column CS is the full adder outputs, where the LSB is the sum (S), and the MSB is the carry (C). Whenever the approximate full adder output is incorrect, the corresponding output in the CS column is shown in red (*). Outputs that are depicted in black are correct.

In this paper, the exact mirror adder presented in [32] is used as the base design, referred to as CMA in the rest of the paper. Ten different approximate full adders; AMA1, AMA2, AMA3 [32], VAFA [37], NFAx [38], TGA2 [36], LOA [20], and AFA1, AFA2, AFA3 [39], will be evaluated. In Fig. 2, precise and approximate full adders at the transistor or gate level evaluated in this paper have been presented. To better understand circuits in this figure, all logical equations of Sum and Cout of

**Fig. 1.** The proposed AFA4 (a), the proposed AFA5 (b), and the proposed AFA6 (c).

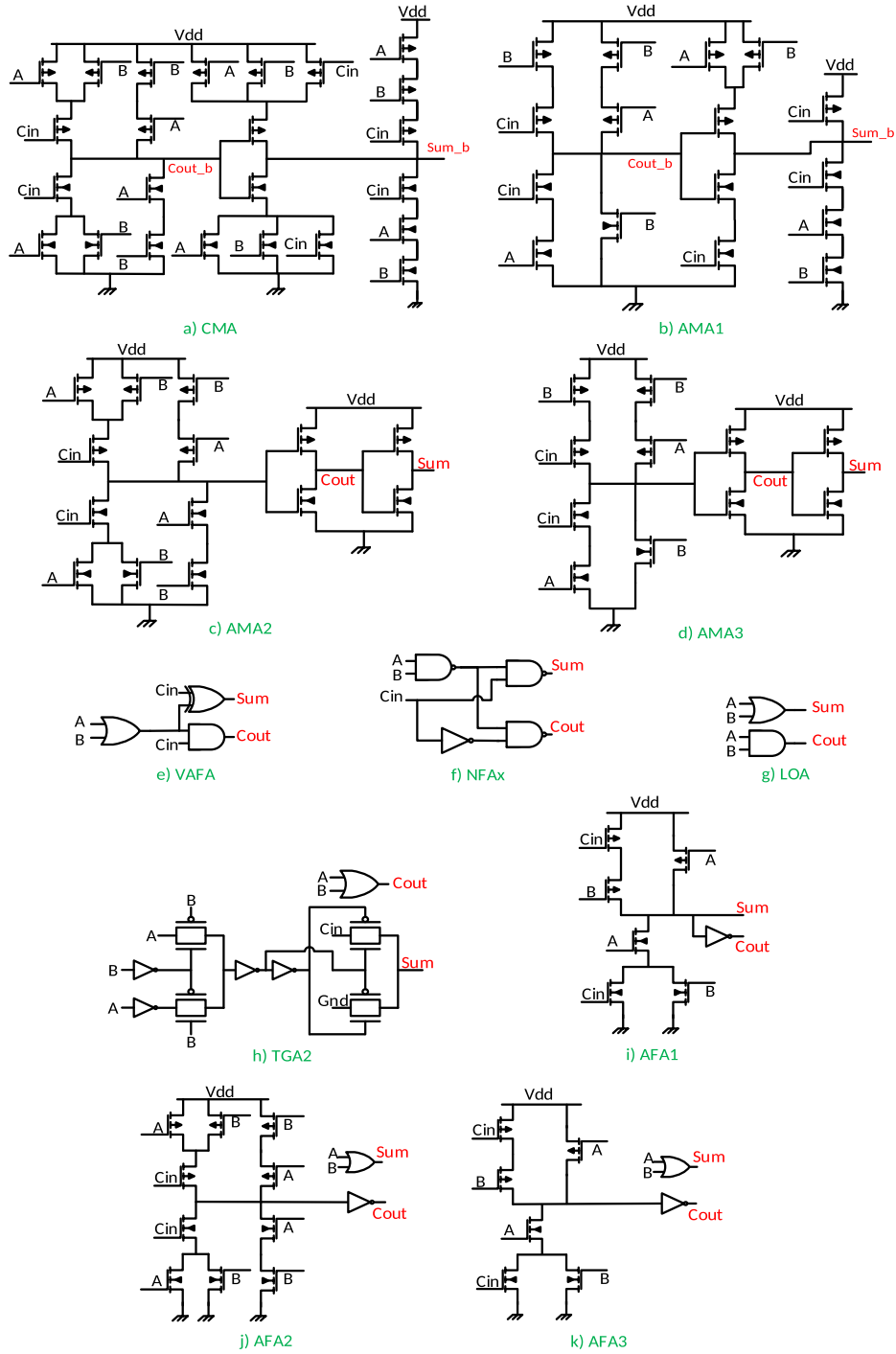


Fig. 2. The existing full adders. a) CMA, b) AMA1, c) AMA2, d) AMA3, e) VAFA, f) NFAx, g) LOA, h) TGA2, i) AFA1, j) AFA2, and k) AFA3.

the approximate full adders are shown in Table 2. According to the results shown in [39], approximate full adders AXAs and InXAs [33,34] have a threshold drop problem, significant error parameters, and high sensitivity to variability. Therefore, this paper ignored these full adders.

In order to make a fair comparison, all approximate full adders in terms of delay, power, PDP, and error have been compared in the presence of variation in three different ways. First, the approximate full adders have been compared for all input combinations. Second, all of them have been used and analyzed in an 8-bit RCA. In the last way, all approximate full adders are evaluated in one image application for five 256*256 images.

For evaluating the error parameters of approximate adders, accord-

ing to [34], the error rate (ER), the normalized mean error distance (NMED), and the mean relative error distance (MRED) are considered. The following equations describe each of these parameters, where n represents the total number of input combinations.

$$ER = \frac{\text{Number of Erroneous Outputs}}{n} \quad (4)$$

$$NMED = \frac{\frac{1}{n} \sum_{i=1}^n |ExactOutput_i - ApproximateOutput_i|}{ExactOutput_{Max}} \quad (5)$$

Table 2

The logical equations of previous approximate full adders.

	Cout	Sum
AMA1	ACin + B	$(A \odot B)Cin$
AMA2	$(A + B) Cin + AB$	$\overline{(A + B)Cin + AB}$
AMA3	ACin + B	$\overline{ACin + B}$
VAFA	$(A + B) Cin$	$(A + B) \oplus Cin$
NFAx	$\overline{AB.Cin}$	$\overline{AB.Cin}$
TGA2	A + B	$(A \odot B)Cin$
LOA	AB	A + B
AFA1	$A(B + Cin)$	$\overline{A(B + Cin)}$
AFA2	$(A + B) Cin + AB$	A + B
AFA3	$A(B + Cin)$	A + B

$$MRED = \frac{1}{n} \sum_{i=1}^n \frac{|ExactOutput_i - ApproximateOutput_i|}{ExactOutput_i} \quad (6)$$

In the last three rows of Table 1, the values of the error parameters for approximate full adders are presented. To calculate ER in Eq. (4), all input combinations are applied and those that result in erroneous outputs are calculated, and finally the total number of errors are divided by n. For example, in Table 1 for AMA1 out of 8 possible input combinations, two of them produce errors at the output and thus $ER = \frac{2}{8} = 0.25$. To calculate NMED in Eq. (5) for every erroneous output, first the sum of absolute values of error distances is calculated, and then divided by the total number of inputs. Finally, the obtained value is divided by the greatest precise value that the circuit produces. For example for AMA1, from 8 possible inputs combinations, two combinations cause errors at the output. The sum of absolute values of error distances equals two which is then divided by 8. Knowing that in a precise full adder the maximum output value is 3, therefore $NMED = \frac{0.25}{3} = 0.083$. MRED in Eq. (6) is calculated similarly. To calculate MRED for AMA2, AMA3, NFAx, AFA1, and AFA5, there are errors for inputs A = B = Cin = 0, and according to Eq. 6, RED is infinite, but in this case, RED is assumed to be 1. According to Table 1, VAFA has the lowest error rate, followed by AFA4 and AFA6, respectively.

3.1. Full adder performance evaluation

In this section, one exact mirror full adder and 13 approximate full adders will be compared in terms of power, delay, PDP, and area, using HSPICE simulator in 32 nm PTM technology [40]. The area is reported by the number of transistors in the full adder. To obtain power, delay,

Table 3

Area and average and maximum power, delay and PDP of approximate full adders.

Full Adders	Power (μ w)		Delay (ps)		PDP (aj)		Area # of Tran
	Avg	Max	Avg	Max	Avg	Max	
CMA	6.04	10.63	24.81	40	151.69	287.72	28
AMA1	4.24	6.8	18.73	33.59	82.21	189.88	20
AMA2	3.77	4.89	21.55	28.37	80.98	118.31	14
AMA3	3.36	4.31	20.76	25.46	69.52	100.62	11
VAFA	6.89 [§]	11.15 [§]	27.58 [§]	43.41 [§]	205.8 [§]	454.06 [§]	24 [§]
NFAx	3.25	4.89	14.66	19.35	47.16	76.45	14
TGA2	4.59	7.65	19.48	39.6	100.1	286.11	22
LOA	2.77	5.02	13.69	17.12 [‡]	38.46	78.74	12
AFA1	2.01 [‡]	2.58 [‡]	12.59 [‡]	19.15	25.66 [‡]	47.76 [‡]	8 [‡]
AFA2	3.78	6.55	15.57	21.52	59.3	117.3	18
AFA3	2.76	4.99	13.75	19.15	38.45	73.13	14
AFA4	3.95	6.18	18.48	30.72	76.26	166.20	17
AFA5	2.14 [‡]	2.8 [‡]	12.05 [‡]	14.82 [‡]	25.74 [‡]	40.31 [‡]	6 [‡]
AFA6	4.15	6.99	18.03	34.73	77.85	210.67	19

and PDP, all possible scenarios are applied to the full adder's inputs, and their mean and maximum values are presented in Table 3. For example, the first scenario is the input change from ABCin = 000 to ABCin = 001; the second scenario is the input switch from ABCin = 000 to ABCin = 010, and so on to the last scenario, where the input changes from ABCin = 111 to ABCin = 110. In this paper the base inverter has the following sizes: Wn = 64 nm, Wp = 128 nm and Lp = Ln = 32 nm. The other gates and more complex designs sizes are an integer factor of the base size. For instance, in the Nand2 gate we have Wn = 128 nm, Wp = 128 nm and Lp = Ln = 32 nm. For the load capacitance 4 inverters at the output of all approximate full adders have been considered, which means each full adder drives 4 base inverters. Also, all full adders' inputs are changed every 250 ps.

To report the delay, the largest delay between the Sum and Cout is considered. In Table 3, the values in green (or [‡]) indicate the best value, the values in blue (or [§]) show the second-best amount, and the values in red (or [§]) indicate the worst amount (the same applies to all subsequent tables). According to Table 3, the lowest average and maximum power belongs to AFA1, the lowest average and maximum delay belongs to AFA5. For the lowest average PDP, AFA1 is the best, and AFA5 is the second-best, however the values are very close. The lowest maximum of PDP is for AFA5. In terms of area, AFA5 with six transistors has the least area. The highest amounts of power, delay, PDP, and area belong to VAFA as it uses the XOR2 gate in CMOS logic. According to Table 3, AFA4 reduces the average power, average delay, average PDP, and area compared to CMA by 34.59%, 25.50%, 49.72%, and 39.29%, respectively. AFA5 also reduces the average power, average delay, average PDP, and area compared to CMA by 64.64%, 51.43%, 83.03%, and 78.57%, respectively.

According to Table 1, the approximate full adders VAFA, AFA4, and AFA6 are similar for ER and NMED, but for MRED, VAFA is better than the others. Nevertheless, according to Table 3, VAFA is the worst performer in terms of power, delay, PDP, and area, and may not be the right choice for imprecision-tolerant applications. Based on Tables 1 and 3, AFA4 is the best choice in terms of power, delay, PDP, and accuracy. In later sections, these approximate full adders are used in larger structures such as RCA or image processing algorithms to make the above conclusions trustworthy.

3.2. Evaluation of variability effects on approximate full adders

In this section, we will evaluate the approximate full adders in terms of variability. According to [18], Vth is the most critical parameter in the variability for new technologies, and the impacts of D2D variability are approximately 2 to 3 times greater than WID variability [39]. For this reason, in this paper, the effects of Vth D2D variability on approximate full adders have been evaluated. All simulations are performed using 32 nm PTM technology using HSPICE, and 1024-point Monte Carlo simulations are used to assess the variability effects.

As a matter of fact, based on HSPICE manual the relation between the relative error and the number of Monte Carlo iterations is as follows [41]:

$$RelativeError = \frac{1}{\sqrt{NumberOfMonteCarloIterations}} \quad (7)$$

Based on Eq. 7 and NumOfMC = 1024, the relative error is about 3% which is relatively small. According to this manual, if the circuit operates correctly for all 1024 iterations, there is a 99% probability that over 96% of all possible component values also operate correctly. Based on simulations, when 10000 Monte Carlo iterations are performed, Mean, Standard Deviation, and Cv values have between 10^{-4} to 10^{-3} difference with 1024 Monte Carlo iterations. Hence, the accuracy of results is almost the same in both cases; however, the simulation time of 10000 Monte Carlo iterations is 10X longer.

Variability of Vth relative to its nominal value is considered 20%

based on a Gaussian distribution [18]. For each of the scenarios, a 1024-point Monte Carlo simulation is executed, and the amount of variation is calculated (we will explain how we measure variability later in the same section). The average variation impacts of these scenarios will be considered. To assess variation impacts, we use some mathematical equations [9]. For this purpose, by using Monte Carlo simulation for the Vth parameter, power, delay, and PDP are measured. Based on Eqs. (9)–(11), the Mean $\mu(x)$, the variance $\text{Var}(x)$, and the Standard Deviation $\sigma(x)$ of each of power, delay and PDP, based on simulation results are determined, and then using Eq. 8, variation coefficient Cv is obtained. First 1024 Monte Carlo iterations are done for input scenarios. For example, for scenario one, 1024 values are achieved for each of the power, delay, and PDP. To calculate the power variation in this scenario, first the Mean value (Eq. 9), and the Standard Deviation (Eq. 11) are calculated, then using Eq. 8, Cv is obtained. The same procedure is used for delay and PDP. Once this is done for all input scenarios, the average of power Cv, delay Cv, and PDP Cv are calculated. For two designs with identical Mean values, a smaller variation coefficient shows a lower variation impact on the full adder, meaning that it is robust against variability.

$$C_v = \frac{\sigma(x)}{\mu(x)} \quad (8)$$

$$\mu(x) = \frac{\sum_{i=1}^n x_i}{n} \quad (9)$$

$$\text{Var}(x) = \frac{\sum_{i=1}^n (x_i - \mu(x))^2}{n - 1} \quad (10)$$

$$\sigma(x) = \sqrt{\text{Var}(x)} \quad (11)$$

Table 4 presents the evaluation results of the Vth parameter D2D variation on the power, delay, and PDP of the approximate full adders. As seen in this table, the variability increases the average power, delay, and PDP of the full adders. The least effects of power, delay, and PDP variability are 27.14%, 18.72%, and 28.24% for AMA2, AFA5, and AFA4, respectively. The highest power and PDP variability impacts belong to TGA2 with 56.67% and 47.90%, respectively, whereas the most delay variability effects belong to VAFA with 24.87%. One of the reasons TGA2 is more vulnerable to variability is related to its transistor-level design, as transmission gates are used in its structure.

Table 4

Average Vth D2D variation impacts on full adders' power, delay and PDP.

Full Adders	Power (μW)			Delay (ps)			PDP (aj)		
	Mean	S.D	C.V(%)	Mean	S.D	C.V(%)	Mean	S.D	C.V(%)
CMA	6.56	1.94	29.56	27.5	7.15	26	177.94	47.64	26.77
AMA1	4.68	1.64	35.01	20.43	4.68	22.89	96.15	27.43	28.53
AMA2	4.07	1.11	27.14 [†]	23.72	5.83	24.59	94.66	28.04	29.62
AMA3	3.64	1.01	27.88 [‡]	22.63	5.18	22.89	80.66	23.37	28.98
VAFA	7.53	2.56	33.96	30.65	7.62	24.87 [§]	243.5	99.41	40.83
NFAx	3.62	1.36	37.52	15.73	3.05	19.36 [‡]	55.09	16.96	30.78
TGA2	5.3	3	56.67 [§]	21.23	4.99	23.48	120.34	57.64	47.9 [§]
LOA	3.1	1.25	40.33	14.79	3.05	20.59	45.34	14.78	32.59
AFA1	2.24	0.79	35.46	13.54	2.7	19.96	30.18	9.84	32.61
AFA2	4.17	1.5	35.89	17.01	3.89	22.88	69.78	22.1	31.67
AFA3	3.13	1.37	43.66	14.9	3.14	21.07	45.93	16.48	35.89
AFA4	4.36	1.46	33.54	20.30	4.77	23.55	89.50	25.30	28.24 [†]
AFA5	2.35	0.78	33.02	12.86	2.41	18.72 [†]	29.69	8.53	28.72
AFA6	4.58	1.54	33.64	19.80	4.74	23.91	91.90	26.10	28.41 [‡]

4. Evaluation of approximate RCA

In this section, we intend to use the approximate full adders in the RCA structure, discussed in the preceding section. The performance, error, and variability effects are evaluated, and a method to reduce the variability effects will be proposed. For this purpose, an 8-bit RCA has been designed and approximate full adders in the first to fourth bits have been used. Four scenarios for implementing the RCA have been considered, where NAB is defined as the number of approximate bits. In the first one, namely NAB1, an approximate full adder is employed only for the least significant bit. In the second one, namely NAB2, approximate full adders are employed for the two least significant bits. Similarly, in the fourth scenario, namely NAB4, approximate full adders are used for the four least significant bits.

4.1. Approximate RCA performance evaluation

To evaluate the performance (power, delay, and PDP) of an approximate 8-bit RCA, the HSPICE simulator and 32 nm technology are used. For example, in the approximate RCA with NAB1, all 131072 input scenarios, where two unsigned 8-bit numbers are added, have been simulated and the average and maximum values of the power, delay, and PDP parameters are obtained. The results for NAB4 are presented in Table 5. In terms of performance (power, delay, PDP, and area) in the approximate 8-bit RCA with NAB4, AFA5 is always the best, and LOA is

Table 5

Area and average and maximum power, delay and PDP of approximate adders for NAB4.

Adders	Power (μW)		Delay (ps)		PDP (aj)		Area # of Tran
	Avg	Max	Avg	Max	Avg	Max	
CMA	20.95	24.8	127.33	296.9	2.65	5.23	224
AMA1	15.39	20.31	94.57	169.6	1.48	2.87	192
AMA2	15.66	18.34	110.39 [§]	192.6	1.71	2.73	168
AMA3	13.47	17.72	93.36	170.7	1.27	2.3	156
VAFA	17.32 [§]	25.01 [§]	97.86	246.6 [§]	1.77 [§]	4.67 [§]	208 [§]
NFAx	12.66	17.72	87.49	173.7	1.17	2.56	168
TGA2	13.27	17.56	87.36	163.4	1.19	2.28	200
LOA	11.41 [†]	15.69 [‡]	86.59 [‡]	161.2 [‡]	1.01 [†]	1.89 [‡]	142 [‡]
AFA1	12.83	16.33	93.45	179.4	1.21	2.21	144
AFA2	16.21	19.46	106.75	185.9	1.72	2.78	184
AFA3	14.27	17.7	93.35	179.5	1.34	2.45	168
AFA4	13.35	18.57	87.45	162.50	1.19	2.60	180
AFA5	10.51 [†]	14.22 [†]	86.56 [†]	161.2 [†]	0.93 [†]	1.71 [†]	130 [†]
AFA6	16.39	20.50	96.49	192.30	1.60	3.15	188

the second-best. The worst performance belongs to VAFA, except for the average delay, which is the worst for AMA2. The same trend is true for NAB1 to NAB3. In terms of delay, AFA4 for different NABs usually occupies the first to third position due to the use of the And2 gate for computing the next stage carry, reducing thereby the carry chain length.

According to the average results in Table 5, in comparison to CMA, the AFA family has reduced power by 21% to 49%, delay by 16% to 32%, and PDP by 35% to 65%. The AMA family has reduced power by 25% to 35%, delay by 13% to 26%, and PDP by 35% to 51%. For example, AFA4 has reduced power, delay, and PDP by 36.28%, 31.32%, and 55.01%, and TGA2 has reduced power, delay, and PDP by 36.66%, 31.39%, and 55.18%, respectively.

4.2. Approximate RCA error evaluation

To evaluate the approximate RCA errors (ER, NMED, and MRED), in the MATLAB environment, all possible input states for the addition of two unsigned 8-bit numbers have been considered. Figs. 3–5 show the error parameters. According to Table 1, AFA4 and AFA6 are accurate when $C_{in} = 0$. Also, for the addition of two unsigned numbers, C_{in0} is equal to zero, and in NAB1, only the least significant bit of the adder is approximate, and the other 7 bits are accurate. As a result, in AFA4 and AFA6 for NAB1, all 8 bits of the adder perform precisely, and therefore in Figs. 3–5, the error parameters in NAB1 for these adders are equal to zero.

According to Fig. 3, AFA4 and AFA6 are quite similar and have the lowest error rates for different NABs. The highest error rate also belongs to NFAX. As shown in Fig. 4 (Fig. 5), the lowest NMED (MRED) error for NAB = 1,2 is jointly owned by AFA4 and AFA6, and for NAB = 3,4, the lowest NMED (MRED) belongs to AFA4, and AFA6 is in the second position. The highest NMED belongs to NFAX. According to Fig. 5, the least MRED for NAB = 1,2 jointly belongs to AFA4 and AFA6, whereas for NAB = 3,4 it belongs to AFA4, and AFA6 comes in second position. The highest MRED for NAB = 1 belongs to AMA3, and for NAB = 2,3,4 it belongs to NFAX. Although AFA5 performs better than LOA in terms of power, delay, PDP, and area, LOA is far better in terms of error parameters.

In order to better compare the approximate adders, the performance and the error parameters are considered together. For this purpose, we define two criteria $PDP \times Area \times NMED(PAN)$ and $PDP \times Area \times MRED(PAM)$ for approximate adders. The PAN (PAM) is the product of three parameters of average PDP, area, and NMED (MRED), respectively. In designing approximate circuits, we always attempt to reduce

PDP, area, and error. Therefore, the smaller PAN and PAM represent better design in terms of performance and error. Table 6 shows the values of the PAN and PAM for different approximation adders for different NABs.

According to Table 6, AFA4 is always the best for the PAN criterion, and the second place for NAB = 1,2,3 belongs to AFA6, and for NAB = 4 it belongs to LOA. The worst PAN for NAB = 1 is for AMA1, for NAB = 2 is for NFAX, and for NAB = 3,4 is for AFA2. AFA4 has reduced PAN by 13.15% and 24.23% compared to AFA6 for NAB = 2,3 and decreased by 9.25% for NAB = 4 compared to LOA, respectively. Given that AFA4 and AFA6 are accurate for the addition of two unsigned numbers for NAB1, the PAN and PAM values corresponding to these states are zero. As for the PAM criterion, the above conclusions are almost the same.

4.3. Approximate RCA variation evaluation

To evaluate the D2D process variation of the V_{th} parameter of an approximate 8-bit RCA, the HSPICE simulator with 32 nm technology has been used and 1024-point Monte Carlo simulation has been run. To assess the effects of the V_{th} parameter variation on different adders, all possible input scenarios are applied, and their impacts on power, delay, and PDP using Eqs. (8)–(11) are calculated. Finally, the average variation of all input scenarios is considered as the final value. On the left-hand side of Figs. 6–8 (Nominal section), C_v values of power, delay, and PDP are presented for different adders.

In Figs. 6–8, the results are divided into Nominal and PV Aware. Nominal variability implies the effects of process variation without the presence of any method of reducing the impacts of variability. PV Aware shows the effects of variability after applying methods to reduce the impacts of variability, explained below.

According to Figs. 6–8 in the Nominal mode, with increasing NAB, the power, delay, and PDP variations of the approximate adders increase. Based on Fig. 6, the lowest power variations belong to AMA2, AFA2, and AFA1 by about 60.1%, 61.4% and 64%, respectively. The highest power variations belong to TGA2 and NFAX, by 80.5% and 71.7%, respectively. According to Fig. 7 in the Nominal mode, the delay variations of the various approximate adders are very close, with the least delay variation being for VAFA at 25.95% and the highest being for AFA6 at 27.10%. According to Fig. 8 in this mode, the least PDP variations belong to AMA2, AFA2 and AFA1, by about 45.46%, 46.45% and 48.75%, respectively. The most PDP variations belong to TGA2 and NFAX by 61.3% and 54.55%, respectively. The percentages reported here are the average variations of NAB1 to NAB4.

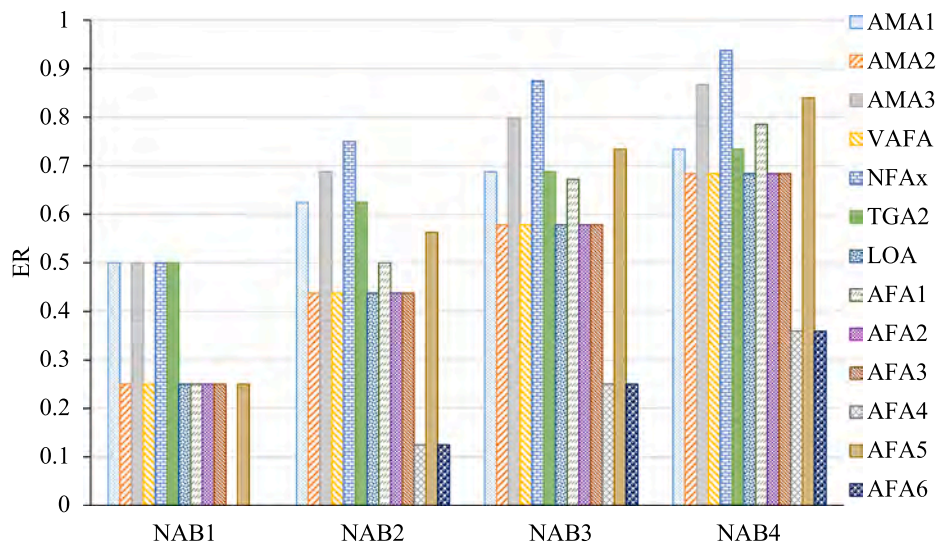


Fig. 3. ER of different 8-bit approximate adders for various NAB values.

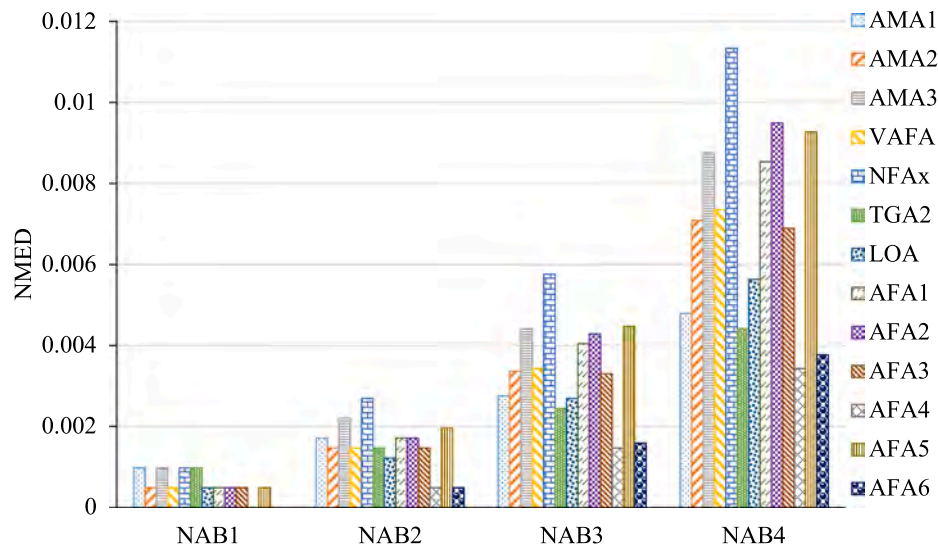


Fig. 4. NMED of different 8-bit approximate adders for various NAB values.

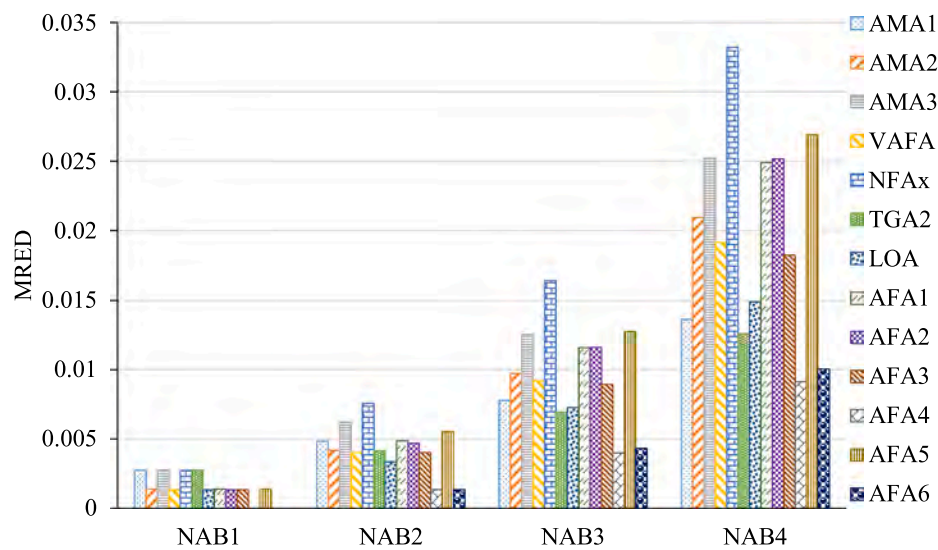


Fig. 5. MRED of different 8-bit approximate adders for various NAB values.

Table 6

PAN and PAM criteria of different 8-bit adders for various NAB values.

Adders	PAN (10^{-16})				PAM (10^{-16})			
	NAB1	NAB2	NAB3	NAB4	NAB1	NAB2	NAB3	NAB4
AMA1	4.98 [§]	7.28	9.81	13.61	13.99 [§]	20.51	27.74	38.66
AMA2	2.58	6.47	12.15	20.42	7.3	18.44	35.1	60.31
AMA3	4.67	8.1	12.3	17.41	13.16	22.87	34.96	50.1
VAFA	2.67	6.96	13.55	27.07	7.36	18.96	36.27	70.41
NFAx	4.79	10.09 [§]	15.34	22.37	13.37	28.4 [§]	43.78	65.54
TGA2	4.67	5.77	7.86	10.46	13.11	16.27	22.3	29.84
LOA	2.21	4.03	6.28	8.11 [‡]	6.08	11.02	16.94	21.42 [‡]
AFA1	2.29	5.98	10.27	14.81	6.48	16.96	29.41	43.2
AFA2	2.63	7.85	16.51 [§]	29.98 [§]	7.25	21.5	44.69 [§]	79.45 [§]
AFA3	2.41	5.7	10.05	15.54	6.63	15.6	27.15	41.06
AFA4	0 [‡]	1.85 [‡]	4.41 [‡]	7.36 [‡]	0 [‡]	5.07 [‡]	11.96 [‡]	19.59 [‡]
AFA5	2.1	5.96	9.23	11.25	5.93	16.82	26.32	32.66
AFA6	0 [‡]	2.13 [‡]	5.82 [‡]	11.31	0 [‡]	5.84 [‡]	15.81 [‡]	30.15

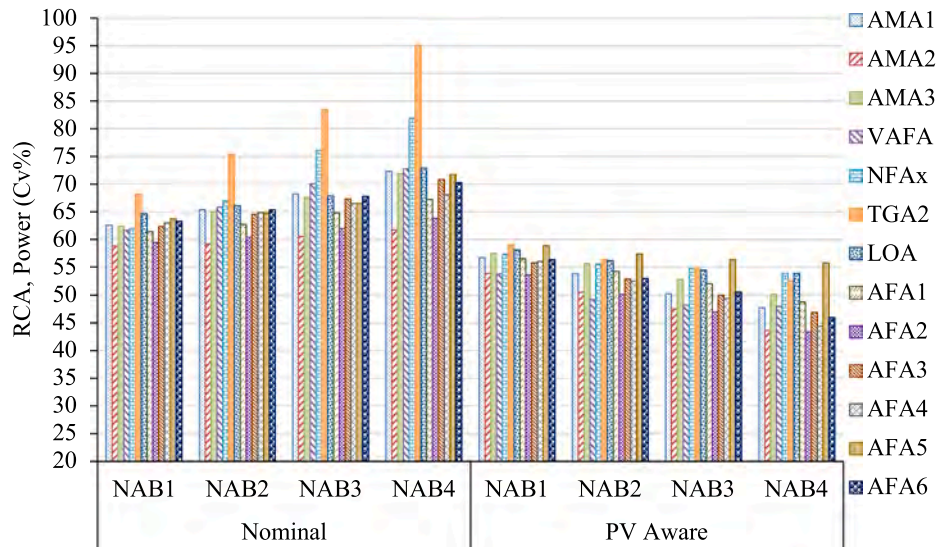


Fig. 6. D2D power variation impacts on Vth for approximate adders for various NAB values (Nominal and PV Aware).

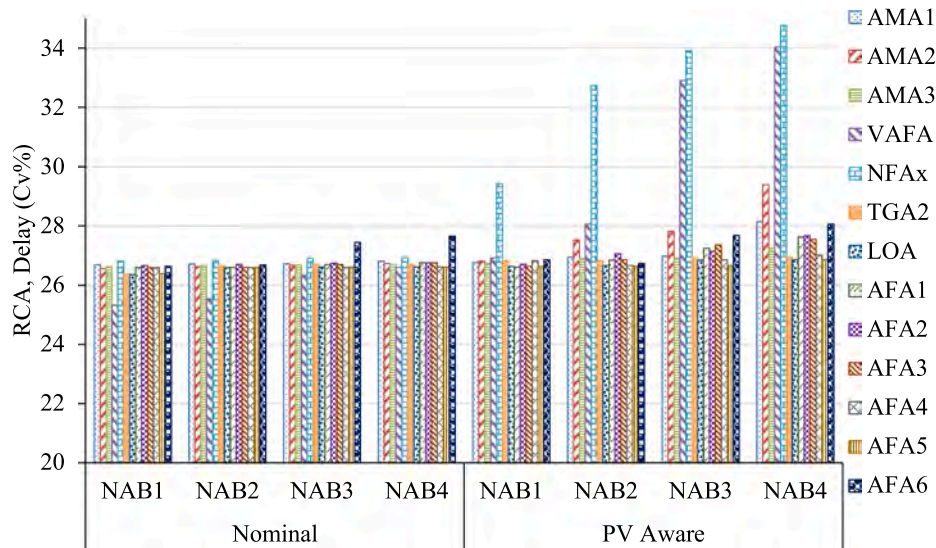


Fig. 7. D2D delay variation impacts on Vth for approximate adders for various NAB values (Nominal and PV Aware).

4.4. Reducing the effects of approximate adder variability with PV Aware method

In [9], two separate methods for reducing the effects of variability in 32 nm technology have been presented. The first one is called Variation-aware dual-Vth (VADVT), and the second one is named Variation-aware dual-Vdd (VADVD). Based on the simulations performed in HSPICE, when approximate full adders with a low threshold voltage (V_{thL}) are used in the imprecision parts (NAB) of approximate adders, a small increase in power, a decrease in delay, and a slight decrease in PDP are resulted. V_{thL} state reduces the power and PDP variations significantly and increases the delay variation slightly. When using the approximate full adders with a low supply voltage (V_{ddL}) in the imprecision parts (NAB) of the approximate adders, it significantly reduces power and PDP and increases delay a little. V_{ddL} state reduces power and PDP variations while increasing delay variation. The variation increase or decrease in the V_{ddL} state in the approximate adder is much less than in the V_{thL} state, however the power, delay, and PDP increase or decrease is much higher.

The proposed PV Aware method in this paper is a combination of the

above two methods to reduce power and its variation in the approximate adders. In terms of delay, we are also looking to keep it constant or increase the delay slightly. For this purpose, in the approximate part of an approximate adder, approximate full adders are used whose threshold voltage and supply voltage are lower than those of the exact full adders.

In the dual-Vdd method, we have two voltage domains. A gate in the V_{ddH} domain can directly drive a gate in the V_{ddL} domain, in which case the latter gate can switch faster. However, a gate in the V_{ddL} domain can drive a gate in the V_{ddH} domain only when the difference $V_{ddH}-V_{ddL}$ is less than the transistors threshold voltage. Otherwise, there will be a contention current between PMOS and NMOS transistors. For solving this problem, there are two methods: (1) using a high- V_{th} PMOS transistor, (2) using a level converter [9]. To alleviate this problem in our proposed approach, a constraint is added, which chooses among the extracted supply and threshold voltages, those that have the difference $V_{ddH}-V_{ddL}$ less than the threshold voltage. In this way, neither a level converter is needed to be used nor high- V_{th} PMOS transistors.

To determine V_{thL} and V_{ddL} values, the Pareto diagram between the PDP and the PDP variation values has been used. The step length for

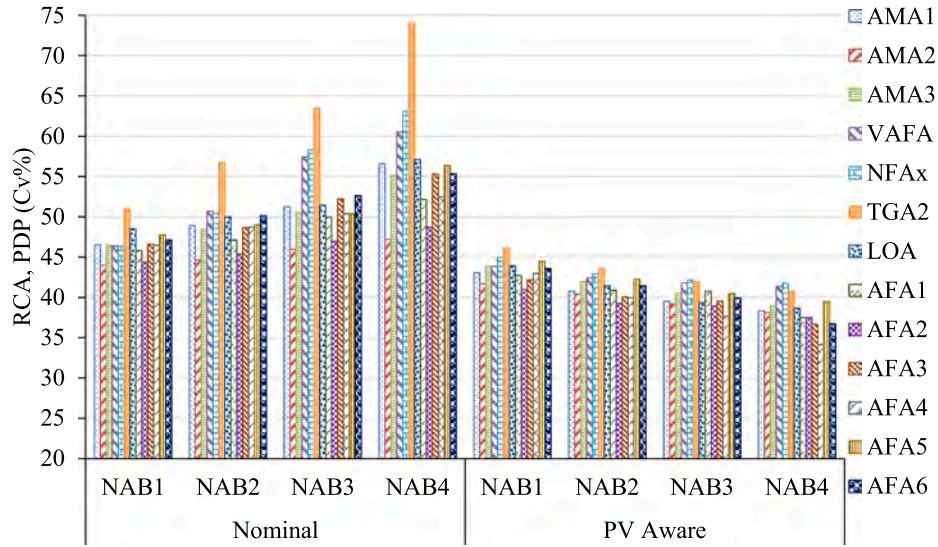


Fig. 8. D2D PDP variation impacts on Vth for approximate adders for various NAB values (Nominal and PV Aware).

VthL is 0.001v and for VddL 0.01v. Also, the limitation of not using a level converter is being considered. The values of VddH and VthH are 0.9v and 0.16v, respectively (typical values for 32 nm technology). Table 7 shows the extracted values of VthL and VddL for each adder for NAB4, which are the best trade-off between performance and variability. Having too many supply voltage levels and technologies with many threshold voltages is not practical, and therefore in our proposed method only two supply and two threshold voltages have been considered. According to results we have obtained in Table 7, these two levels are: VddH = 0.9v, VddL = 0.8v, VthH = 0.16v and VthL = 0.15v. In Table 7, the relative performance of 8-bit approximate adders with NAB4 for PV Aware compared to the Nominal method is presented. According to this table, the PV Aware method reduces power and PDP and increases delay. For example, the use of PV Aware in AFA4 reduces power and PDP by 5% and 4%, respectively, and increases delay by 1% compared to the Nominal method.

According to Figs. 6–8 in the PV Aware method, with increasing NAB, the amount of power and PDP variations of the approximate adders decrease, whereas the delay variation increases. The delay variation is negligible compared to reduced power and PDP variations. According to Fig. 6 in the PV Aware method, the lowest power variations belong to AFA2, AMA2 and VAFA by 48.54%, 48.9% and 49.81%, respectively. The highest power variations are in AFA5 and TGA2 by 57.08% and 55.65%, respectively. According to Fig. 7 in the PV Aware method, the delay variations of various approximate adders except VAFA and NFAx

are very close, with the least delay variations being for AFA5, LOA, AFA4 and AMA3 with 26.7%, 26.75%, 26.84% and 26.95%, respectively. The highest delay variation is for NFAx with 32.71%. According to Fig. 8 in the PV Aware method, the least variations of PDP belong to AFA4, AFA2, AFA3 and AMA2 by about 38.68%, 39.17%, 39.61% and 39.86%, respectively. The highest variations of PDP belong to TGA2 and NFAx by about 43.1% and 42.95%, respectively. The percentages reported here are the average variation of NAB1 to NAB4. According to the results of Table 7 and Figs. 6–8, the PV Aware method has reduced power, PDP, power variation, and PDP variation compared to the Nominal method and increased delay and delay variation very slightly.

5. Imprecision-tolerant real application simulation results

Image Sharpening application has been used to evaluate the approximate full adders. To produce a sharp image, Eq. (12) in Image Sharpening application has been employed. In this equation I and O are input and output images, respectively and Mask is a 5*5 matrix as in Eq. (13). Image Sharpening algorithm consists of multipliers, adders, dividers, and subtractors, where most of the runtime belongs to multiplication and addition. To produce an output pixel in this algorithm, 26 multiplications, 25 additions, 1 division and 1 subtraction are needed [29,30]. Therefore, as there is a great number of multiplications and additions in this algorithm, approximate full adders have been used in the structure of multipliers and adders. As the maximum values for an image pixel and in the Mask Matrix are 255 and 41, respectively, an 8*6 approximate array multiplier as well as a 14-bit RCA approximate adder have been used to implement this application. All approximate full adders discussed in this paper have been employed to implement multiple approximate multipliers and adders, which are used in Image Sharpening application. Five 256*256 images are applied as input to this application. These images include Lena, Cameraman, Baboon, House and Rice shown in Fig. 9.

$$O(x, y) = 2I(x, y) - \frac{1}{273} \sum_{i=-2}^2 \sum_{j=-2}^2 I(x+i, y+j) \text{Mask}(i+3, j+3) \quad (12)$$

$$\text{Mask} = \begin{bmatrix} 1 & 4 & 7 & 4 & 1 \\ 4 & 16 & 26 & 16 & 4 \\ 7 & 26 & 41 & 26 & 7 \\ 4 & 16 & 26 & 16 & 4 \\ 1 & 4 & 7 & 4 & 1 \end{bmatrix} \quad (13)$$

Table 7

Average power, delay, and PDP of approximate adders in PV Aware method relative to Nominal method for NAB4.

Adders	Voltage (V)		Relative		
	VddL	VthL	Power	Delay	PDP
AMA1	0.80	0.149	0.93	1.03	0.96
AMA2	0.81	0.150	0.94	1.05	0.99
AMA3	0.80	0.149	0.95	1.02	0.97
VAFA	0.78	0.152	0.87	1.09	0.96
NFAx	0.80	0.151	0.95	1.04	0.99
TGA2	0.81	0.149	0.97	1.01	0.98
LOA	0.80	0.149	0.97	1.01	0.98
AFA1	0.81	0.149	0.96	1.02	0.98
AFA2	0.80	0.150	0.94	1.06	0.99
AFA3	0.80	0.149	0.94	1.02	0.96
AFA4	0.80	0.150	0.95	1.01	0.96
AFA5	0.82	0.147	0.99	1.01	0.99
AFA6	0.80	0.149	0.94	1.03	0.96

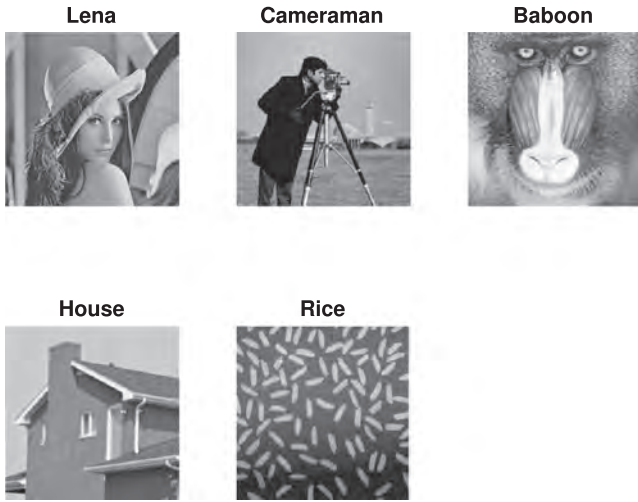


Fig. 9. Input images for Sharpening application.

5.1. Adders performance and variability evaluation with image processing application

First the Sharpening image processing application has been implemented in HSPICE and to evaluate the performance and variability of this algorithm, a special configuration (Config1) has been considered. In this configuration for the multiplier structure, except for P0 output which is produced by a two-input AND gate, P1 to P6 outputs are produced by approximate full adders, and P7 to P13 outputs by exact full adders. For the adder, a 14-bit approximate adder has been used, where the 7-bit LSBs are approximate. Five 256*256 images have been applied as the application input. Also a python-based tool has been developed which takes input images and Mask coefficients matrix, and transforms them to correct input HSPICE pulses. For every output pixel, input pixels and Vth variations are applied to HSPICE by considering Monte Carlo simulations. Finally, based on the results the tool extracts power, delay, PDP and variation effects for the Nominal and PV Aware modes for each image. At the end for an entire image the average power, the maximum delay and average PDP are obtained. In Table 8 for the configuration 1 and 5 different images, we have normalized the values of the average power, maximum delay and average PDP, related to CMA. In this table, the values of performance were presented first with the Nominal mode and second with the PV Aware mode.

Table 8

Average Power and PDP and Maximum Delay of five different images in Sharpening application for Config1.

Sharpening	Relative to CMA					
	Nominal			PV Aware		
	Power	Delay	PDP	Power	Delay	PDP
AMA1	0.845	0.849	0.823	0.822	0.892	0.794 [§]
AMA2	0.771	0.904 [§]	0.756	0.718	0.989 [§]	0.714
AMA3	0.741	0.831	0.726	0.685	0.906	0.715
VAFA	0.864 [§]	0.836	0.843 [§]	0.762	0.953	0.742
NFAx	0.748	0.877	0.697	0.719	0.931	0.676
TGA2	0.836	0.813	0.827	0.761	0.843	0.783
LOA	0.676 [†]	0.711 [†]	0.662 [†]	0.629 [†]	0.720	0.627 [†]
AFA1	0.694	0.804	0.676	0.652	0.867	0.647
AFA2	0.777	0.903	0.732	0.737	0.976	0.701
AFA3	0.760	0.870	0.696	0.746	0.935	0.680
AFA4	0.830	0.715	0.793	0.803	0.717 [‡]	0.764
AFA5	0.682 [‡]	0.679 [†]	0.636 [†]	0.640 [‡]	0.689 [†]	0.607 [†]
AFA6	0.850	0.808	0.816	0.825 [§]	0.860	0.782

Based on Table 8, the highest power reduction is obtained first for LOA and then for AFA5. In terms of PDP and delay, the highest reduction belongs first to AFA5 and then to LOA, although with PV Aware mode AFA4 comes second and LOA third in delay reduction. The reason in Table 8 the best performance belongs to AFA5 and LOA is in these structure the carry out does not depend on carry in. The carry out is generated by an AND gate of inputs A and B, which do not have any dependency on Cin, and thus in array multipliers and RCA adds the carry chain is significantly reduced. On the other hand, to produce Sum, OR2 and NAND2 gates have been simply used which compared to other approximate full adders consume less power.

To evaluate the effects of D2D Vth process variation on this application, 1024-point Monte Carlo simulation in 32 nm technology has been performed. The variation of Vth relative to its nominal value is 20% based on a Gaussian distribution. The average variation of all input scenarios is considered as the final variation. Fig. 10 illustrates the impacts of variability on power, delay and PDP for Sharpening application for the Nominal and PV Aware modes. To obtain results for Fig. 10, using the python code, image pixels are given to HSPICE simulator and a Monte Carlo simulation is performed, and based on Eqs. (8)–(11) sensitivity to variations are calculated.

According to Fig. 10 for Sharpening, the lowest power Cv in the Nominal mode belongs to AMA1 and AMA2 by 48.89% and 49.98%, respectively, and the highest is for VAFA with 58.78%. The lowest power Cv in the PV Aware mode relates to AFA6 and AMA2 by 41.16% and 41.38%, respectively. The lowest delay Cv in the Nominal mode belongs to AMA2 and AFA5 by 26.06% and 26.10%, respectively, and the highest is for TGA2 with 29.73%. The lowest delay Cv in the PV Aware mode relates to AMA2 and AFA6 by 27.07% and 27.53%, respectively, and the highest is for TGA2 with 31.82%. The lowest PDP Cv in the Nominal mode belongs to AFA5 and AFA1 by 28.80% and 31.76%, respectively, and the highest is for VAFA with 43.76%. The lowest delay Cv in the PV Aware mode relates to AFA4 and AMA1 by 23.64% and 24.00%, respectively, and the highest is for VAFA with 33.99%. Based on Fig. 10, PV Aware method causes the delay variation to increase by 1.7%, and the PDP variation to decrease by 6.4% on average.

5.2. Approximate adder error evaluation with image processing application

In image processing algorithms, such as Sharpening, some of the most important parameters used to compare different methods are PSNR, MSE and MSSIM [42], are explained as follows:

Mean Square Error (MSE):

$$MSE = \frac{1}{n \times m} \sum_{i=1}^m \sum_{j=1}^n (P_{ij} - \widehat{P}_{ij})^2 \quad (14)$$

Peak Signal to Noise Ratio (PSNR):

$$PSNR = 10 \log \frac{255^2}{MSE} \quad (15)$$

Mean Structural Similarity Index (MSSIM):

$$MSSIM = \frac{1}{n \times m} \sum_{i=1}^m \sum_{j=1}^n SSIM(P_{ij}, \widehat{P}_{ij}) \quad (16)$$

In the above equations, P_{ij} expresses the exact pixel value in the i^{th} row and j^{th} column of the produced exact image; whereas \widehat{P}_{ij} expresses the approximate pixel value in the i^{th} row and j^{th} column of the produced approximate image. m and n are the size of row and column of the image, respectively. Detailed description of the above parameters (Eqs. (14)–(16)) has been given in [42].

In order to compare approximate full adders' error with each other, Sharpening application is implemented in MATLAB and to evaluate the

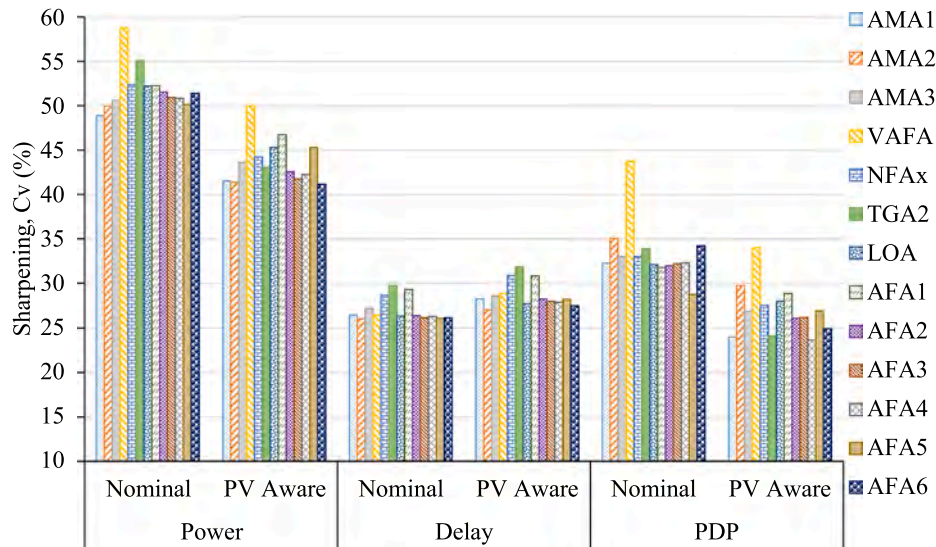


Fig. 10. Sharpening Power, Delay and PDP variations in the Nominal, PV Aware modes for Config1.

precision of this algorithm, two different configurations have been considered. In the first configuration (Config1) for the multiplier structure, except for P0 output which is produced by a two-input AND gate, P1 to P6 outputs are produced by approximate full adders, and P7 to P13 outputs by exact full adders. For the adder, we have used a 14-bit approximate adder, where the 7-bit LSBs are approximate. In the second configuration (Config2) for the multiplier structure, except for P0 which is produced by a two-input AND gate, P1 to P5 are produced by approximate full adders, and P6 to P13 by exact full adders. For the adder, we have used a 14-bit approximate adder, where the 6-bit LSBs are approximate. For input, we also give five images of 256*256 for this application, and we present the average value of PSNR and MSSIM for config1 and config2 in Table 9.

According to results shown in Table 9, for both configurations 1 and 2, AFA6 has the largest PSNR and MSSIM values, where AMA1 comes next. Also, the least PSNR belongs to AMA3. In terms of MSSIM, in the first and the second configurations NFAx and AMA3 are the worst, respectively. Based on Table 9, the approximate full adders LOA and AFA5 occupy the third and fourth places in terms of PSNR and MSSIM parameters. These full adders have simple structures with lower delay and power compared to other full adders and have shown better efficiency in the above application.

6. Conclusion

In this paper, three new approximate full adders (AFA4, AFA5, AFA6) have been proposed, and the effects of D2D Vth process variation on all approximate full adders have been evaluated. A variation-aware approach has been proposed which allows to best fit an approximate full adder by slightly increasing the delay, and reducing power and sensitivity to variations. According to the simulation results, when using approximate full adders in the RCA structure, AFA4 has the best performance in terms of PAN, whereas AFA6 is ranked second. For different NABs, AFA4 reduces PAN by about 73% to 82% compared to the worst approximate full adder. The lowest power, delay, and PDP in the approximate adders belong to AFA5 and for NAB4 compared to the worst approximate adder it reduces power, delay, and PDP by 49.27%, 21.27%, and 57.55%, respectively. In Sharpening application for five different images, the average PSNR for different configurations is the highest for AFA6, and the lowest for AMA3. In an approximate full adder design, which is obtained by introducing some errors to its truth table, when MRED parameter is important for the designer, there must not be any error at the output for small inputs such as ABCin = 000,001;

Table 9

Average PSNR and MSSIM of five different images in Sharpening application for Config1 and Config2.

Sharpening	Config1		Config2	
	PSNR(dB)	MSSIM	PSNR(dB)	MSSIM
AMA1	31.84 [‡]	0.9573 [‡]	36.34 [‡]	0.9842 [‡]
AMA2	25.59	0.8525	28.03	0.9200
AMA3	21.97 [§]	0.7400	24.34 [§]	0.8366 [§]
VAFA	23.42	0.8289	25.40	0.8822
NFAx	22.09	0.7355 [§]	24.75	0.8473
TGA2	22.26	0.7479	25.39	0.8647
LOA	31.83	0.9547	33.11	0.9666
AFA1	25.60	0.8526	28.03	0.9200
AFA2	24.83	0.8536	27.76	0.9153
AFA3	25.79	0.8825	28.44	0.9268
AFA4	27.72	0.9160	31.17	0.9558
AFA5	29.83	0.9371	31.46	0.9611
AFA6	33.21 [†]	0.9687 [†]	36.81 [†]	0.9847 [†]

otherwise the error value compared to that of a precise output becomes significant. For instance, for the same reason AMA3 and NFAx are not efficient in terms of PSNR for image processing applications. Also, it is recommended that for variation effects, transmission gates and XOR or XNOR gates not be used when possible, as they are sensitive to variability. For example, VAFA and TGA2 are the most sensitive to variability. Consequently, the trade-off between performance, accuracy, and variability is the best for AFA4 and AFA6 compared to the existing approximate full adders.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This research was in part supported by a grant from the Institute for Research in Fundamental Sciences (IPM) (Grant No.CS1398-4-14).

References

- [1] M.S. Ansari, B.F. Cockburn, J. Han, Low-power approximate logarithmic squaring circuit design for dsp applications, *IEEE Transactions on Emerging Topics in Computing* doi:10.1109/TETC.2020.2989699.
- [2] H. Esmaeilzadeh, A. Sampson, L. Ceze, D. Burger, Architecture support for disciplined approximate programming, in: *ACM SIGPLAN Notices*, Vol. 47, ACM, 2012, pp. 301–312. doi:10.1145/2150976.2151008.
- [3] S. Ullah, S. Rehman, M. Shafique, A. Kumar, High-performance accurate and approximate multipliers for fpga-based hardware accelerators, *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* doi:10.1109/TCAD.2021.3056337.
- [4] M.S. Kim, A.A.D.B. Garcia, H. Kim, N. Bagherzadeh, The effects of approximate multiplication on convolutional neural networks, *IEEE Transactions on Emerging Topics in Computing* doi:10.1109/TETC.2021.3050989.
- [5] Kamal M, Ghasemazar A, Afzali-Kusha A, Pedram M. Improving efficiency of extensible processors by using approximate custom instructions. In: 2014 Design, Automation & Test in Europe Conference & Exhibition (DATE). IEEE; 2014. p. 1–4. <https://doi.org/10.7873/DATE.2014.238>.
- [6] Samadi M, Lee J, Jamshidi DA, Hormati A, Mahlke S. Sage: Self-tuning approximation for graphics engines, in. In: *Proceedings of the 46th Annual IEEE/ACM International Symposium on Microarchitecture*; 2013. p. 13–24. <https://doi.org/10.1145/2540708.2540711>.
- [7] Garg B, Patel SK. Reconfigurable carry look-ahead adder trading accuracy for energy efficiency. *Journal of Signal Processing Systems* 2021;93(1):99–111. <https://doi.org/10.1007/s11265-020-01542-1>.
- [8] Adl SMT, Mirzaei M, Mohammadi S. Elastic buffer evaluation for link pipelining under process variation. *IET Circuits, Devices & Systems* 2018;12(5):645–54. <https://doi.org/10.1049/iet-cds.2017.0394>.
- [9] Mirzaei M, Mosaffa M, Mohammadi S. Variation-aware approaches with power improvement in digital circuits. *Integration, the VLSI Journal* 2015;48:83–100. <https://doi.org/10.1016/j.vlsi.2014.07.001>.
- [10] S.V.R. Chittamuru, I.G. Thakkar, S. Pasricha, S.S. Vatsavai, V. Bhat, Exploiting process variations to secure photonic noc architectures from snooping attacks, *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* doi:10.1109/TCAD.2020.3014184.
- [11] Chatterjee A, Musavvir S, Kim RG, Doppa JR, Pande PP. Power management of monolithic 3d manycore chips with inter-tier process variations. *ACM Journal on Emerging Technologies in Computing Systems (JETC)* 2021;17(2):1–19. <https://doi.org/10.1145/3430765>.
- [12] Y. Zhu, G.L. Zhang, T. Wang, B. Li, Y. Shi, T.-Y. Ho, U. Schlichtmann, Statistical training for neuromorphic computing using memristor-based crossbars considering process variations and noise, in: 2020 Design, Automation & Test in Europe Conference & Exhibition (DATE), IEEE, 2020, pp. 1590–1593. doi:10.23919/DATE48585.2020.9116244.
- [13] G.P. Srinivasa, S. Haseley, G. Challen, M. Hempstead, Quantifying process variations and its impacts on smartphones, in: 2019 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS), IEEE, 2019, pp. 117–126. doi:10.1109/ISPASS.2019.00019.
- [14] Sanabria-Borbón AC, Soto-Aguilar S, Estrada-López JJ, Allaire D, Sánchez-Sinencio E. Gaussian-process-based surrogate for optimization-aided and process-variations-aware analog circuit design. *Electronics* 2020;9(4):685. <https://doi.org/10.3390/electronics9040685>.
- [15] Li D, Minoia G, Repossi M, Baldi D, Temporiti E, Mazzanti A, Svelto F. A low-noise design technique for high-speed cmos optical receivers. *IEEE J. Solid-State Circuits* 2014;49(6):1437–47. <https://doi.org/10.1109/JSSC.2014.2322868>.
- [16] A. Awny, R. Nagulapalli, D. Micusik, J. Hoffmann, G. Fischer, D. Kissinger, A.C. Ulusoy, 23.5 a dual 64gbaud 10kω5% thd linear differential transimpedance amplifier with automatic gain control in 0.13 μm bimos technology for optical fiber coherent receivers, in: 2016 IEEE International Solid-State Circuits Conference (ISSCC), IEEE, 2016, pp. 406–407. doi:10.1109/ISSCC.2016.7418079.
- [17] R. Nagulapalli, K. Hayatleh, S. Barker, S. Zourob, N. Yassine, S. Sridevi, A pvt insensitive programmable amplifier for biomedical applications, in: 2017 International conference on Microelectronic Devices, Circuits and Systems (ICMDCS), IEEE, 2017, pp. 1–5. doi:10.1109/ICMDCS.2017.8211724.
- [18] Mirzaei M, Mosaffa M, Mohammadi S, Trajkovic J. Power and variability improvement of an asynchronous router using stacking and dual-vth approaches. In: *Digital System Design (DSD)*, 2013 Euromicro Conference on. IEEE; 2013. p. 327–34. <https://doi.org/10.1109/DSD.2013.41>.
- [19] Yang T, Ukezono T, Sato T. A low-power configurable adder for approximate applications. In: 2018 19th International Symposium on Quality Electronic Design (ISQED). IEEE; 2018. p. 347–52. <https://doi.org/10.1109/ISQED.2018.8357311>.
- [20] Mahdiani HR, Ahmadi A, Fakhraie SM, Lucas C. Bio-inspired imprecise computational blocks for efficient vlsi implementation of soft-computing applications. *IEEE Trans. Circuits Syst. I Regul. Pap.* 2010;57(4):850–62. <https://doi.org/10.1109/TCSI.2009.2027626>.
- [21] L.B. Soares, M.M.A. da Rosa, C.M. Diniz, E.A.C. da Costa, S. Bampi, Design methodology to explore hybrid approximate adders for energy-efficient image and video processing accelerators, *IEEE Transactions on Circuits and Systems I: Regular Papers* doi:10.1109/TCSI.2019.2892588.
- [22] Mazahir S, Ayub MK, Hasan O, Shafique M. Probabilistic error analysis of approximate adders and multipliers. In: *Approximate Circuits*. Springer; 2019. p. 99–120. https://doi.org/10.1007/978-3-319-99322-5_5.
- [23] Wu Y, Li Y, Ge X, Gao Y, Qian W. An efficient method for calculating the error statistics of block-based approximate adders. *IEEE Trans. Comput.* 2019;68(1): 21–38. <https://doi.org/10.1109/TC.2018.2859960>.
- [24] Akbari O, Kamal M, Afzali-Kusha A, Pedram M. Rap-cla: A reconfigurable approximate carry look-ahead adder. *IEEE Trans. Circuits Syst. II Express Briefs* 2018;65(8):1089–93. <https://doi.org/10.1109/TCSII.2016.2633307>.
- [25] Almurib HA, Kumar TN, Lombardi F. Approximate dct image compression using inexact computing. *IEEE Transactions on computers* 2018;67(2):149–59. <https://doi.org/10.1109/TC.2017.2731770>.
- [26] Dharmaraj C, Vasudevan V, Chandrachoodan N. Optimization of signal processing applications using parameterized error models for approximate adders. *ACM Transactions on Embedded Computing Systems (TECS)* 2021;20(2):1–25. <https://doi.org/10.1145/3430509>.
- [27] Rezaalipour M, Dehyadegari M. Linear-time error calculation for approximate adders. *Computers & Electrical Engineering* 2021;92:107139. <https://doi.org/10.1016/j.compeleceng.2021.107139>.
- [28] Fatemeh SE, Farahani SS, Reshadinezhad MR. Lahaf: Low-power, area-efficient, and high-performance approximate full adder based on static cmos. *Sustainable Computing: Informatics and Systems* 2021;30:100529. <https://doi.org/10.1016/j.suscom.2021.100529>.
- [29] Ahmadinejad M, Moaiyeri MH, Sabetzadeh F. Energy and area efficient imprecise compressors for approximate multiplication at nanoscale. *AEU-International Journal of Electronics and Communications* 2019;110:152859. <https://doi.org/10.1016/j.aeue.2019.152859>.
- [30] Jain R, Pandey N. Approximate karatsuba multiplier for error-resilient applications. *AEU-International Journal of Electronics and Communications* 2021; 153579. <https://doi.org/10.1016/j.aeue.2020.153579>.
- [31] Gupta V, Mohapatra D, Park SP, Raghunathan A, Roy K. Impact: imprecise adders for low-power approximate computing. In: *Proceedings of the 17th IEEE/ACM international symposium on Low-power electronics and design*. IEEE Press; 2011. p. 409–14.
- [32] Gupta V, Mohapatra D, Raghunathan A, Roy K. Low-power digital signal processing using approximate adders. *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.* 2013;32(1):124–37. <https://doi.org/10.1109/TCAD.2012.2217962>.
- [33] Z. Yang, A. Jain, J. Liang, J. Han, F. Lombardi, Approximate xor/xnor-based adders for inexact computing, in: *Nanotechnology (IEEE-NANO)*, 2013 13th IEEE Conference on, IEEE, 2013, pp. 690–693. doi:10.1109/NANO.2013.6720793.
- [34] H.A.F. Almurib, T.N. Kumar, F. Lombardi, Inexact designs for approximate low power addition by cell replacement, in: *Proceedings of the 2016 Conference on Design, Automation & Test in Europe, DATE '16*, EDA Consortium, San Jose, CA, USA, 2016, pp. 660–665. <http://dl.acm.org/citation.cfm?id=2971808.2971962>.
- [35] Weste NH, Harris D. *CMOS VLSI design: a circuits and systems perspective*. Pearson Education India; 2015.
- [36] Yang Z, Han J, Lombardi F. Transmission gate-based approximate adders for inexact computing. In: 2015 IEEE/ACM International Symposium on Nanoscale Architectures (NANOARCH). IEEE; 2015. p. 145–50. <https://doi.org/10.1109/NANOARCH.2015.7180603>.
- [37] S. Venkatachalam, S.-B. Ko, Design of power and area efficient approximate multipliers, *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 25 (5) (2017) 1782–1786. doi:10.1109/TVLSI.2016.2643639.
- [38] Waris H, Wang C, Liu W. High-performance approximate half and full adder cells using nand logic gate. *IEICE Electronics Express* 2019. <https://doi.org/10.1587/eleex.16.20190043>.
- [39] Mirzaei M, Mohammadi S. Process variation-aware approximate full adders for imprecision-tolerant applications. *Computers & Electrical Engineering* 2020;87: 106761. <https://doi.org/10.1016/j.compeleceng.2020.106761>.
- [40] Nanoscale integration and modeling (nimo) group, predictive technology model (ptm), note = <http://ptm.asu.edu/>, last accessed march 3, 2020.
- [41] *HSPICE. User Guide, Basic Simulation and Analysis*. Synopsys; 2013.
- [42] Wang Z, Bovik AC, Sheikh HR, Simoncelli EP, et al. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing* 2004; 13(4):600–12. <https://doi.org/10.1109/TIP.2003.819861>.