A Timing-Aware Configurable Adder Based on Timing Detection for Low-Voltage Computing

Xuemei Fan , Tingting Zhang , *Graduate Student Member*, IEEE, Hao Liu ,
Shengli Lu , and Jie Han , *Senior Member, IEEE*

***Abstract*—Low-voltage computing effectively saves energy in circuit operations, but it suffers from an increasing propagation delay. Approximate computing can significantly reduce the propagation delay by using a simplified or improved circuit, albeit with an inevitable accuracy loss. To address these challenges, a timing-aware configurable adder (TACA) is proposed to achieve a good trade-off between energy efficiency and accuracy at low operating voltages. This design relies on the functions of timing-error detection and correction (TEDC) for the newly-proposed accuracy-configurable full adders (ACFAs). The ACFA operates in an exact mode and two approximate modes by using four transistors as power gating. The TEDC generates timing-error signals when the delay violates the timing constraint due to voltage overscaling. Then, an improved configuration scheme is developed to enable the ACFA to work in an approximate mode by allowing for error signals at runtime. This approximation shortens the carry propagation chain. Thus, the TACA is adapted to timing conditions at different supply voltages by reducing the propagation delay rather than the operation frequency.**

***Keywords*—Low-Voltage Operation, Timing Detection and Correction, Approximate Computing, Dynamic Accuracy-Configurable Adder.**

I. Introduction

Low-voltage operation provides an efficient way to reduce power in circuits and systems [1-6]. However, due to process, voltage, and temperature (PVT) variations, it is challenging to satisfy all timing constraints [1]. The propagation delay of a circuit drastically increases as the supply voltage scales down, leading to considerable timing errors. Conventional designs avoid these errors by conservatively reserving timing margins and reducing the operation frequency. However, the use of these timing margins wastes excessive energy and reduces the throughput, because a circuit does not always work in the worst case. Applicable to the voltage overscaling (VOS) technique, configurable approximate computing (AC) emerges as a potential approach to overcoming these limitations [2-7].

AC effectively reduces delay and power dissipation by using simplified or improved circuits. Although an accuracy loss is inevitable, it can be partly tolerated in some applications, such as neural networks (NNs) and image processing [8]. In order to adapt to various accuracy requirements or timing conditions, many accuracy-configurable adder (ACA) designs [9-20] have been researched. Most of the studies reduce the propagation delay at low voltages by directly discarding some least significant bits (LSBs) in the input data [5-6]. However, it leads to a considerable accuracy loss. As the supply voltage scales down, the increasing number of discarded LSBs significantly deteriorates the accuracy. To improve the accuracy, some ACA designs use computation-error detection and compensation (CEDC) units [9-11]. Although the CEDC allows the ACA to produce an exact result, it simultaneously increases the design complexity and power dissipation of the circuits. Moreover, circuits for realizing the dynamic configuration between exact and approximate operation modes incur extra hardware overhead and a considerable increase in delay.

This paper presents an energy-efficient timing-aware configurable adder (TACA) design for operations at a low supply voltage. The TACA consists of accuracy-configurable full adders (ACFAs) and timing-error detection and correction (TEDC) units. When the delay violates the timing constraints at low voltages, the ACFA will work in an approximate mode (AM), configured by a timing-error signal generated by the TEDC unit at runtime. This shortens the carry chain of the TACA to make it stably work at lower voltages without reducing the operation frequency or throughput. Otherwise, the ACFA works in the exact mode (EM). Moreover, an improved configuration scheme (ICS) is also developed to reduce the accuracy loss in the TACA.

The proposed TACA provides a superior trade-off between hardware performance and accuracy by switching between the EM and AM at different supply voltages. The main novelties of this work are summarized as follows:

- An ACFA is designed by adding four transistors in the traditional mirror adder. The operation is switched between an exact and two approximate modes. Compared with an exact full adder, the proposed ACFA reduces energy by 5.16% in the approximate mode.
- An ICS is developed to reduce the accuracy loss of multi-bit adders based on ACFAs. Compared with an exact adder, a 16-bit ACFA-based adder achieves up to 80% power saving and a reduction in the critical path delay by 83.02%, with an accuracy loss of only 11.96%. This accuracy is further verified by high output qualities of three image processing applications.

Manuscript received 23 October 2022; revised 2 January 2023; accepted 26 January 2023. This work was supported in part by the Fundamental Research Funds for the Central Universities under Project 3206002204C3 and in part by the Natural Sciences and Engineering Research Council (NSERC) of Canada under Project RES0048688. This article was recommended by Guest Editor C. N. M. Ho. (Corresponding authors: Hao Liu; Xuemei Fan.)

Xuemei Fan, Hao Liu, and Shengli Lu are with the National ASIC System Engineering Technology Research Center, Southeast University, Nanjing, Jiangsu 210096, China (e-mail: merry_fan@seu.edu.cn; nicky_lh@seu.edu.cn; lsl@seu.edu.cn).

T. Zhang and H. Jie are with the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB T6G 1H9, Canada (e-mail: ttzhang@ualberta.ca; jhan8@ualberta.ca).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/JETCAS.2023.3243135.

Digital Object Identifier 10.1109/JETCAS.2023.3243135

- A TACA is developed from the TEDC units and ACFAs with the ICS to realize runtime accuracy configuration at various supply voltages. It is further applied in a classification application using convolutional neural networks (CNNs). As the supply voltage scales down from 1.1 V to 0.5 V, the operation frequency and energy efficiency of the computing circuits in the CNN accelerator are significantly improved without decreasing the throughput.

II. Related Work

The use of VOS and configurable AC techniques for further improving energy efficiency has been considered in the literature [2-7, 23]. Afzali-Kusha et al used voltage islands to perform the VOS in an approximate coarse grain reconfigurable architecture [2-3]. Different blocks of an accuracy-configurable block-based carry look-ahead adder (AC-CLA) are supplied with different voltages by a power switch box to improve energy efficiency [3]. However, the AC-CLA suffers from a considerable extra hardware cost. The complexity of a design is effectively reduced by simply truncating the LSBs, resulting in a significant accuracy loss [4-6]. Moreover, many extra razor flip-flops [21] with a large circuit cost are used to monitor timing conditions [4-5]. Hence, the tradeoff between accuracy and energy efficiency needs to be effectively improved. Although razor flip-flops [21] are still used to monitor the timing condition, Toshinori et al. improved the accuracy and shortened the propagation delay by using the carry-maskable adders [14] at low voltages [23]. Yin et al. further improved the accuracy by grouping the input data and customizing multi-bit adders, where the length of the critical path depends on the number of bits in each group [7].

The existing ACA designs can be categorized into four groups depending on the configuration methods:

i) Truncation-based: This method directly ignores the operations of some LSBs in an adder to simplify the circuit. The designs in [4-6] configure the approximation degree of adders by simply discarding different numbers of LSBs. It brings a significant reduction in delay and power dissipation. However, the accuracy loss will deteriorate as the number of LSBs increases.

ii) CEDC-based: The CEDC-based ACAs use different numbers of CEDC units to correct errors, thus producing results with various accuracies. However, it used a large number of sub-adders and comparison circuits to realize different levels of approximation [9]. Moreover, extra clock cycles are required to correct errors, leading to a large delay. Although more flexible designs [10,11] were proposed, the problem of high hardware costs resulting from the use of redundant sub-adders and EDC circuits cannot be thoroughly eliminated.

iii) Carry-prediction-based: This kind of ACAs uses the predicted carry signals as the carry-in of the more significant full adders to shorten the carry chain [12]. For example, Xu et al. added some carry selection circuits and sub-adders to configure the accuracy [13]. By omitting the error-correction circuits, the carry-prediction-based method outperforms the CEDC-based approach [10]. However, these designs still suffer from considerable power overhead caused by the sub-adders, carry prediction and carry selection circuits.

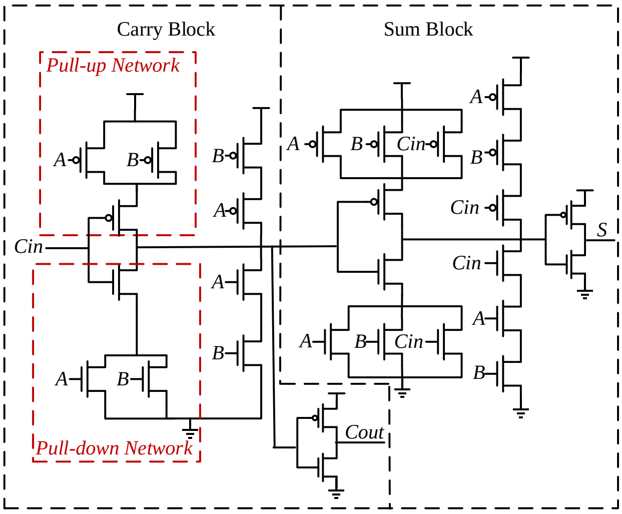


Fig. 1 The structure of the MA [22].

Yang et al. used a 3-input NAND gate to realize configuration by a mask signal without redundant carry-prediction and CEDC circuits [14]. However, it can only configure the accuracy between the exact adder and the lower-part-OR adder [20, 24], which introduces a considerable loss in accuracy.

iv) Power-gating-based: This class introduces the transistor-level power gating to perform the configuration [15-20]. It uses a few transistors as power gating, instead of complex gate-level circuits for high energy efficiency. For example, Tsai et al. added the switching transistors and output isolation elements into a radix-4 adder to enable this adder to work in exact and approximate modes [15]. Similarly, the adders in [16-17] realize reconfiguration between two modes, by adding transistors as switches into a mirror adder (MA). Compared with the design in [16], the light-weight configurable approximate adder (LCAA) [17] obtains a higher accuracy in the approximate mode because it only modifies the operation of the carry signal. A dual-mode full adder (DMFA) [18] introduces a transistor as power gating to directly connect the output signals with input signals to reduce the delay. However, two additional multiplexers are required to perform configurations. Chandan et al. generated approximate sum signals but did not simplify the generation of the carry signal, thus only obtaining a limited benefit for reducing the delay [19]. The runtime accuracy reconfigurable adder (RARA) [20] introduces four transistors as switches to simplify the operation of the MA. However, the RARA reduces the delay and improves the energy efficiency only when the logic value of the carry-in signal is 0.

III. Proposed Accuracy-configurable Full Adder

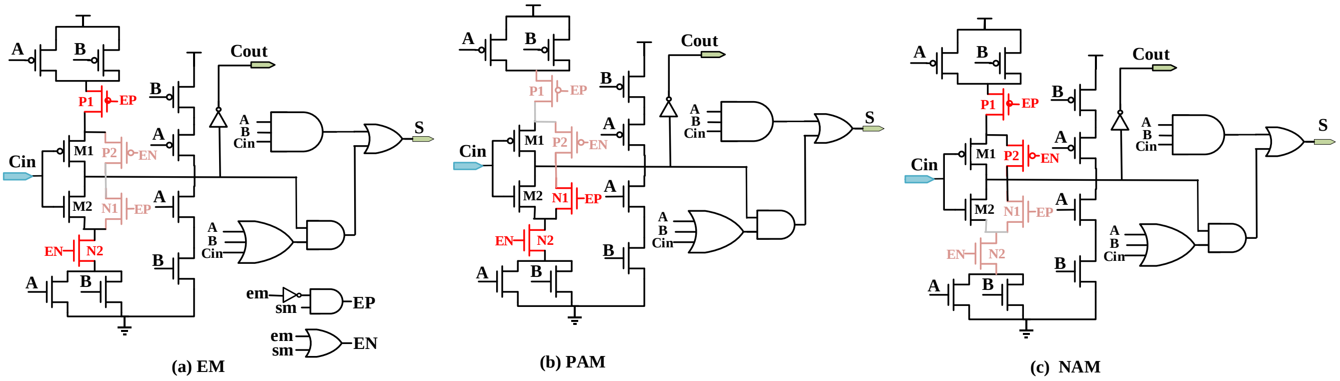
A. Preliminaries

A mirror full adder (MA) [22] consists of a sum block and a carry block with a pull-up network (PUN) and a pull-down network (PDN), as shown in Fig.1. Logic expressions of the output signals denoted by $Cout$ and S are given by:

$$Cout = (A + B) \cdot Cin + A \cdot B\,, \tag{1}$$

$$S = A \cdot B \cdot Cin + (A + B + Cin) \cdot \overline{Cout}\,, \tag{2}$$

where Cin, A and B represent the input carry-in signal and two input data signals, respectively. As indicated in (2), the



(a) EM (b) PAM (c) NAM

Fig. 2 The structures of the proposed ACFA working in three different modes.

generation of S depends on the propagation of $\overline{Cout}$. Thus, the propagation delay of the circuit path increases as the carry chain lengthens.

The commonly used *n*-bit ripple carry adder (RCA) is constructed by cascading *n* full adders in series [6]. The carry-out signal of the lower significant full adder is propagated and serves as the carry-in signal of the next full adder with higher significance. As a result, the propagation delay of an *n*-bit RCA is in $O(n)$, which would proportionally deteriorate as the length of the carry chain increases [1].

TABLE I THE TRUTH TABLE OF THE PROPOSED ACFA

Inputs			Accurate (*em* =1) (*sm* =0)		Approximate (*em* =0)			
A	B	Cin	S	Cout	Positive (*sm* =1)		Negative (*sm* =0)	
					S	Cout	S	Cout
0	0	0	0	0	0			
0	1	0	1	0	0×	1×	1	0
1	0	0	1	0	0×	1×	1	0
1	1	0	0	1	0	1	0	1
0	0	1	1	0	1	0	1	0
0	1	1	0	1	0	1	1×	0×
1	0	1	0	1	0	1	1×	0×
1	1	1	1	1	1			

B. *The proposed ACFA design*

The proposed ACFA adds four transistors into the carry block of an MA. As shown in Fig. 2, two transistors in the proposed ACFA, denoted by P1 and N2, are added in series with M1 and M2, respectively. The other two, denoted by N1 and P2, are added in parallel with M2 and M1. As switches, these four transistors cut off the carry chain in the approximate mode. Thus, the proposed ACFA can operate in three modes: the exact mode (EM), the positive approximate mode (PAM) and the negative approximate mode (NAM), which are configured by the mode-selection signals, denoted by *em* and *sm*.

Fig. 2 presents the operations of the ACFA in different modes. The signals denoted by *EN* and *EP* are used to enable or disenable these four transistors, the logic values of them are given by:

$$\begin{cases} EP = \overline{em}\cdot sm \\ EN = em + sm \end{cases}. \tag{3}$$

Then, the three operation modes are configured by *em* and *sm* in three cases, as follows:

(i) EM: When *em* is set to logic ‘1’, the logic of *EN* and *EP* will become ‘1’ and ‘0’, respectively, as indicated in (3). P1 and N2 all turn on, like short circuits. P2 and N1 turn off, like open circuits. In this way, the ACFA works in the EM, as shown in Fig. 2 (a).

(ii) PAM: If *em* and *sm* are respectively set to ‘0’ and ‘1’, both *EN* and *EP* will be set as ‘1’. As shown in Fig. 2 (b), both P1 and P2 turn off as open circuits to shut down the PDN unit. Then, the carry out signal is computed by a logic OR of *A* and *B*, thus generating the approximate sum. As shown in Table I, the ACFA generates two positive errors in this AM when *A*, *B* and *Cin* are ‘010’ and ‘100’.

(iii) NAM: When *em* and *sm* are both set to ‘0’, *EN* and *EP* will be set to ‘0’. Then, as presented in Fig. 2 (c), P1 and P2 will turn on and N1 and N2 will turn off. Due to the shut-down of the PUN unit, the behavior of the carry block is reconfigured to an AND operation of *A* and *B*. As indicated in Table I, it generates two negative errors in this AM, when *A*, *B* and *Cin* are ‘011’ or ‘101’.

Therefore, the logic functions of the ACFA working in three operation modes are expressed by:

$$Cout = (A+B)\cdot(Cin+EN)+A\cdot B\cdot\overline{EN}$$
$$=\begin{cases}(A+B)\cdot Cin + A\cdot B, & EN=0, EP=1\\ A+B, & EN=1, EP=1\\ A\cdot B, & EN=0, EP=0\end{cases}, \tag{4}$$

$$S=\begin{cases} A\oplus B\oplus Cin, & EN=0, EP=1\\ A\cdot B\cdot Cin+(A+B+Cin)\cdot\overline{A+B}, & EN=1, EP=1\\ A\cdot B\cdot Cin+(A+B+Cin)\cdot\overline{A\cdot B}, & EN=0, EP=0\end{cases}. \tag{5}$$

The configuration logic of three operation modes can be expressed by:

$$\begin{cases} EM = EP\cdot\overline{EN} = em\cdot\overline{sm}\\ PAM = EP\cdot EN = \overline{em}\cdot sm\\ NAM = \overline{EP}\cdot\overline{EN} = \overline{em}\cdot\overline{sm}\end{cases}. \tag{6}$$

When operating in the AMs, i.e., both the PAM and NAM, the ACFA produces an inexact result in only two of the eight possible input cases, as shown in Table I. Thus, the probability of producing approximate results is only 25%. The generation of *Cout* is no longer dependent on *Cin*. Therefore, the propagation delay from *Cin* to *Cout* is eliminated, thus cutting off the carry chain and reducing the delay. Moreover, as shown in Fig. 2 (b) and (c), the shut-down of the PUN and PDN units, which are labeled in a light color, contributes to a significant reduction in energy consumption. In the NAM, *S* and *Cout* stay in logic '1' and '0', respectively, whereas it is the opposite case for the PAM. Thus, the power consumption is further reduced because of the decreased internal switch activities. Furthermore, the erroneous results will be offset by the positive and negative errors in the accumulative computation.

IV. Proposed Timing-Aware Configurable Adder

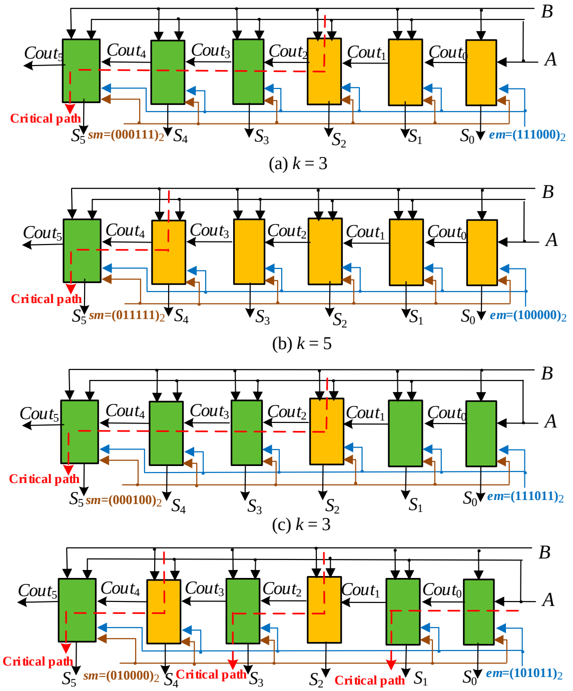
To detect the real-time timing condition and then reduce the propagation delay of addition circuits at a low voltage, TEDC units and the proposed ACFA are employed to build the TACA. An improved configuration scheme (ICS) is especially proposed to improve the accuracy of the ACFA-based adder. Thus, the proposed TACA can stably work at a lower voltage without reducing the operation frequency and obtain a superior tradeoff between hardware performance and accuracy.

A. Improved Configuration Scheme (ICS)

For an *n*-bit adder based on the RCA architecture and ACFAs, a common configuration scheme is generally performed by splitting this adder into an (*n*-*k*)-bit exact part and a *k*-bit approximate part. Fig. 3(a) illustrates the implementation of a 6-bit ACFA-based adder with *k* equal to 3. The ACFAs for computing the three most significant bits (MSBs) labeled in green, work in the EM, whereas those for computing three LSBs labeled in orange, are configured to work in the AM. According to (1) and (2), the sum signals of the 4th-6th least significant full adders denoted by *S*[5:3] are calculated by *A*[5:3], *B*[5:3] and *Cin*[5:3]. *Cin*[5:3] depends on the carry out signal $Cout_2$ propagated from the 3rd adder, ordered from LSBs to MSBs. Because this ACFA works in an AM, $Cout_2$ is only produced based on A_2 and B_2 instead of $Cout_1$, thus cutting the original carry chain. The delay of the critical path is directly proportional to the number of operations on the critical path which is denoted by N_c. Then, the delay of the critical path is significantly reduced, since N_c is reduced from *n*=6 to *n*-*k*+1=4. However, the approximate results produced by the 1st-3rd ACFAs will cause an inevitable accuracy loss, which worsens as the value of *k* increases.

The ICS is proposed to reduce the number of approximate ACFAs to improve the accuracy. Simultaneously, the critical path delay is reduced to be as short as that using the common configuration scheme by specifically choosing the position of these approximate ACFAs. For example, for a 6-bit ACFA-based adder using the ICS, the binary values of *em and sm* are set as $(111011)_2$ and $(000100)_2$, respectively, as shown in Fig. 3 (c). Then N_c of this adder is reduced from *n=6* to *max* {*n*-*k*+1, *k*-1}= 4, which is similar to that in Fig. 3(a). However, it configures only the 3rd least significant ACFA to work in an AM instead of three LSBs. Thus, the accuracy of this adder is higher than that of the adder in Fig. 3(a).

Then, we consider the case where *k* is equal to 5. The 6-bit ACFA-based adder in Fig. 3 (b) configures five LSBs to work in an AM by using the common configuration method. By using the ICS, the adder in Fig. 3 (d) only introduces two approximate ACFAs. Because higher significant bit work in the AM compared with the adders in Fig. 3 (a) and (c), the accuracy of adders in Fig. 3 (d) is obviously worse. However, it is still higher than that of the adder in Fig. 3 (b), resulting from the reduction of the number of approximate ACFAs. Moreover, the critical path of the adder in Fig. 3 (d) is cut off by the 3rd and 5th ACFAs and equally divided into three groups from MSBs to LSBs. In this way, the use of the ICS reduces N_c of this adder from max {*n*-*k*+1, *k*-1}=4 to $\lfloor n/3 \rfloor$ =2, which is similar to using the common configuration scheme as shown in Fig. 3 (b). Thus, the ICS provides a superior tradeoff between accuracy and delay. $\lfloor x \rfloor$ denotes the greatest integer operation which indicates an integral part of the real number *x*.



(d) *k* = 5

Fig. 3 The illustrations of 6-bit ACFA-based adders with different values of *k* using (a) (b) the common configuration scheme and (c) (d) the proposed ICS

Based on the proposed ICS, the critical path of the ACFA-based adder is split into *m*+1 groups, where *m* represents the number of the ACFAs working in the AM. If configuring the LSB of each group to work in the AM, the delay of this adder will be shortened to the minimum. In this way, N_c of this adder can be configured at the range from *n*+1-*m* to $\left\lfloor \frac{n}{m+1} \right\rfloor$. Thus, the critical path delay reduces, as *m* increases, whereas the accuracy of this adder decreases.

When *m* is equal to 1, assume that the *n*-bit adder using the ICS only configures the *k*th ACFA to work in the AM, N_c of this adder will be reduced to *max*{*n*-*k*+1, *k*-1}. However, the N_c of the *n*-bit adder using the common configuration scheme is fixedly reduced to *n*-*k*+1 when this adder works in the AM. When *k* is larger than $\lfloor 0.5n \rfloor + 1$, *max*{*n*-*k*+1, *k*-1} is always equal to *k*-1 and the difference between *k*-1 and *n*-*k*+1 increases as *k* increases. This leads to a larger delay of the critical path by using the ICS, compared with that by using the common configuration scheme. Thus, the value of *m* needs to be increased to further shorten the critical path, when

k is larger than $\lfloor 0.5n \rfloor + 1$. Denote the index of approximate sum signals of the adder using the ICS by S_i. To improve the tradeoff between accuracy and critical path delay, the values of m and i are given by:

$$m = \begin{cases} 1 & , k \leq \lfloor 0.5n \rfloor + 1 \\ \left\lfloor \dfrac{n}{n-k+1} \right\rfloor - 1, & k > \lfloor 0.5n \rfloor + 1 \end{cases}, \tag{7}$$

$$\begin{cases} i = \begin{cases} k-1 & , k \leq \lfloor 0.5n \rfloor + 1 \\ (x+1)k - x(n+1) - 2 \, , & k > \lfloor 0.5n \rfloor + 1 \end{cases} \\ i > 0, \; x = 0, 1, \ldots, \left\lfloor \dfrac{n}{n-k+1} \right\rfloor - 2 \end{cases}. \tag{8}$$

The configuration details with respect to m and Si are presented in TABLE II, when k is larger than $\lfloor 0.5n \rfloor + 1$. By using the proposed ICS, the critical path delay of an ACFA-based adder is similar to that using the common configuration scheme. However, m is always smaller than k, when k is larger than $\lfloor 0.5n \rfloor + 1$, as shown TABLE II. It indicates the fewer approximate ACFAs and the higher accuracy of adders using the ICS, compared with using the common configuration scheme.

TABLE II THE CONFIGURATION DETAILS WITH RESPECT TO *m* AND *S*$_I$ WITH DIFFERENT VALUES OF *k*

n	k	m	the index of approximate sum signals
8	6	2	S_2, S_5
	7	3	S_2, S_4, S_6
	8	8	S_0, S_1, S_2, ... , S_7
16	10	2	S_2, S_9
	11	2	S_4, S_{10}
	12	3	S_1, S_6, S_{11}
	13	3	S_4, S_8, S_{12}
	14	5	S_1, S_4, S_7, S_{10}, S_{13}
	15	7	S_2, S_4 , S_6, S_8, S_{10}, S_{12}, S_{14}
	16	16	S_0, S_1, S_2, ... , S_{16}

B. The Proposed Multi-bit TACA

Fig. 4 presents the structure of an n-bit TACA constructed by n ACFAs and one TEDC unit (meaning the number of the ACFAs may work in the AM, denoted by m equal to 1), which is developed based on the ICS. The core idea of the TACA is that the timing errors are predicted in advance, by detecting the late-arriving output of the monitored full adder at the middle point of the critical path [25]. The TEDC unit is used to detect the late-arriving output, then generates a timing-error signal. n-1 ACFAs labeled in light yellow always work in the EM, as shown in Fig. 4. One ACFA labeled in orange will be configured to work in an AM depending on the timing-error signal, thus cutting off the carry chain.

Based on a 50% duty-cycle clock premise, one TEDC unit is inserted between the (t+1)th ACFA at the half point of the critical path and the D-flip-flop for storing the results of this ACFA. To reduce the power dissipation, a previous timing error tolerant flip-flop (ETFF) [26] with only nine transistors is employed to implement the TEDC unit by replacing the original D-flip-flop. The TEDC is activated on the falling edge of the signal which is inverted to the clock and denoted by *NCLK*.

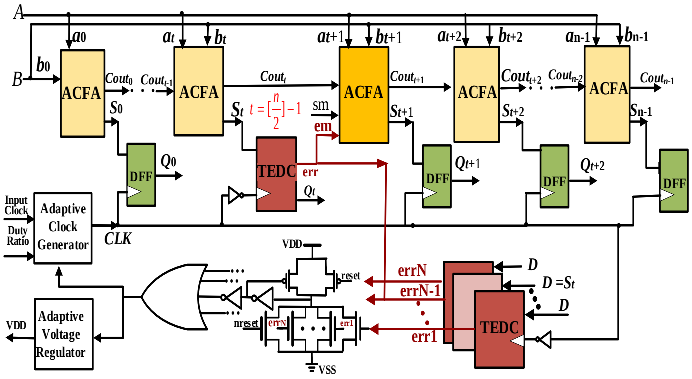
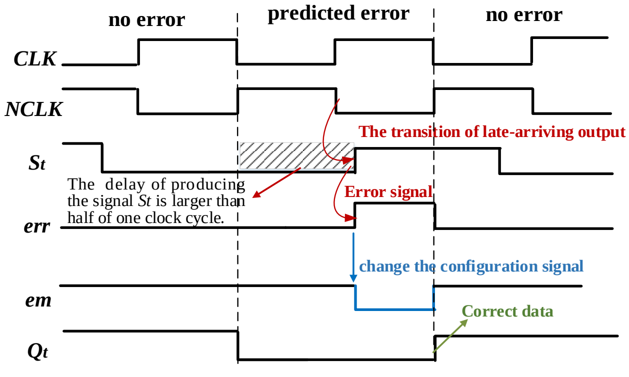


Fig. 4 The structure of the proposed n-bit TACA with m equal to 1



no error
predicted error
no error
CLK
NCLK
St
err
em
Qt
The transition of late-arriving output
The delay of producing the signal St is larger than half of one clock cycle.
Error signal
change the configuration signal
Correct data

Fig. 5 The timing diagram of the proposed TACA, where the configuration signal denoted by *em* is changed and it configures the monitored ACFA to work in an AM to cut off the critical path, thus avoiding the timing violation.

When the supply voltage scales down, the delay of producing the output signal *St* of the (t+1)th ACFA will become larger than half of the clock cycle. It indicates that the critical path delay of the TACA will larger than the clock cycle. Then the delay violates the timing constraint and the transition of *Si* will occur after the falling edge of *NCLK*. In this case, the TEDC unit detects the late transition of *Si* during a negative clock phase and generates a timing error signal. The (t+2)th ACFA is configured to work in the AM to cut off the critical path of the TACA. Therefore, the TACA can work at a lower voltage and maintain a high operation frequency without timing violations.

In ideal conditions, this half-path insertion of the TEDC unit is based on a 50% duty-cycle clock premise. If the duty cycle changes, the insertion points will be transferred to the falling edge of *NCLK*. The middle ACFA of the TACA is not always at the half point of the critical path, especially when n is odd. When the monitored ACFA is set in front of an exact middle point, it will cause a false timing-error prediction. On the other hand, if the position of the monitored ACFA is set behind the middle point, it will increase the timing constraint and cause slight energy dissipation. The first- half part and the second-half part of these ACFAs almost have identical logic-depth and are located near each other so that the different impact of PVT variations under the VOS on the delay between these two parts can be ignored.

Thus, to avoid the potential and false timing-error prediction and PVT variations, the monitored ACFA needs to be carefully chosen to be after and as close as possible to the exact middle point. Thus, the value of t is given by:

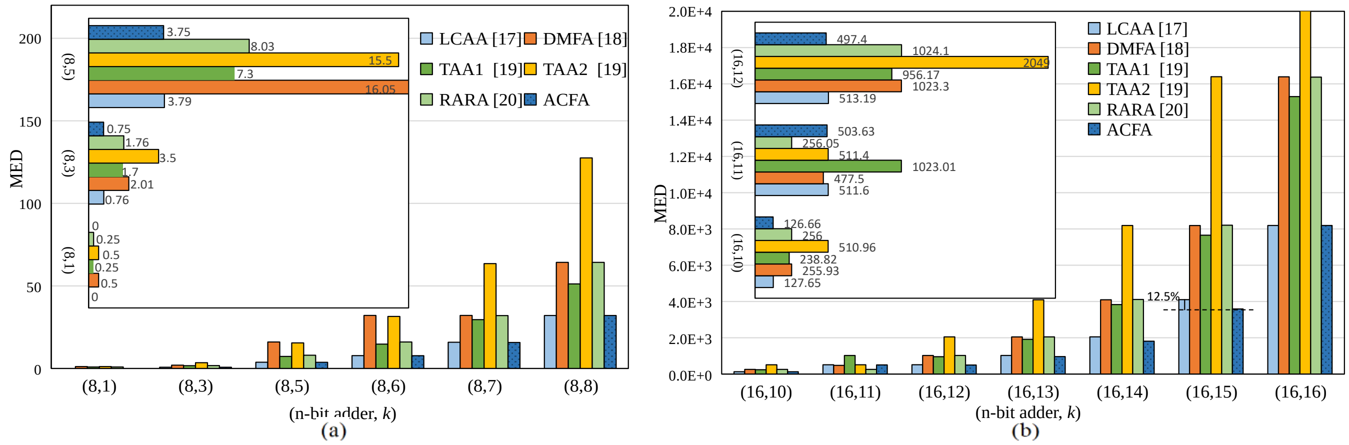


Fig. 6 The performance in MED of (a) 8-bit and (b) 16-bit proposed ACFA-based adders and other ACAs with different values of *k*

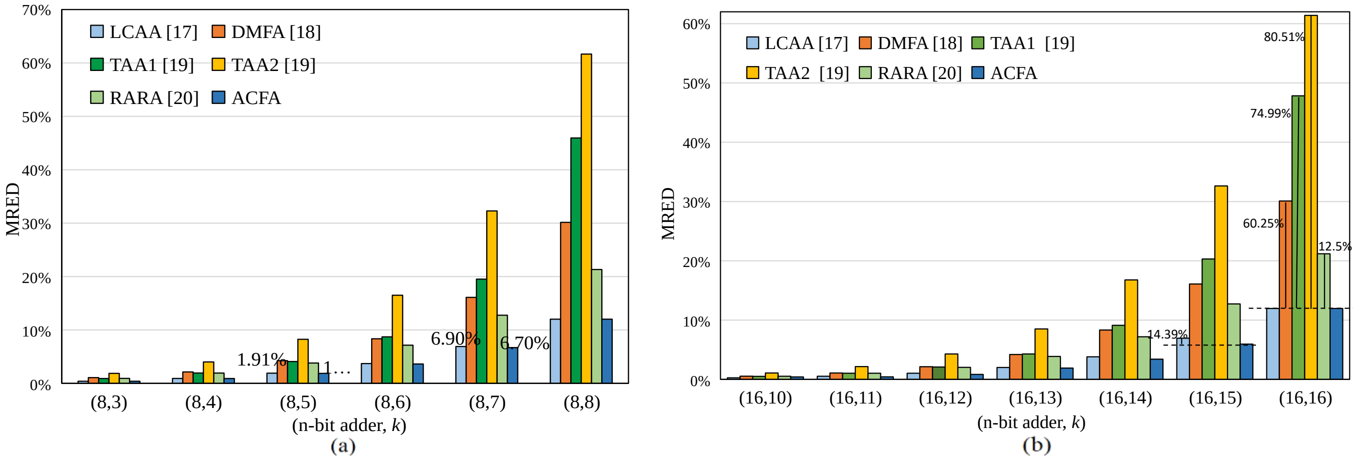


Fig. 7 The performance in MRED of (a) 8-bit and (b) 16-bit the proposed ACFA-based adders and other ACAs with different values of *k*

$$t = \left\lfloor \frac{n}{2} \right\rfloor - 1 \qquad (9)$$

When *m* is equal to 1 and the $(t+2)^{th}$ full adder is configured to work in an AM, it will minimize the critical path delay based on the ICS.

If the duty-cycle clock is further modulated to drive more TEDC units to be inserted in the TACA, more ACFAs can be configured to work in the AM. This allows the proposed TACA to work in the AM with various degrees of approximation and different critical paths. Moreover, multiple TEDCs are also inserted in different paths in a circuit. Then, the potential timing violations of the critical path can be predicted and avoided through the dynamic configuration. To reduce the propagation delay of these error signals, dynamic OR gates are performed to cluster them from different paths. If the number of timing error signals is larger than a threshold, it indicates excessive timing violations in a circuit. Then, the adaptive voltage regulator and adaptive clock generator will reduce the operation frequency or increase the supply voltage to mitigate the problem of timing violation.

V. Performance Evaluation

In this section, the accuracy and hardware performance of the proposed ACFA and multi-bit ACFA-based adder are compared with other ACA designs [17-20]. The energy efficiency of the proposed TACA is also evaluated at the supply voltage ranging from 1.1 V to 0.5 V.

A. Experiment setup

The proposed ACFA-based adder is implemented by using the improved configuration scheme and the previous ACA designs [17-20] use the common configuration scheme. To fairly compare with other ACAs, the approximate ACFAs in the ACFA-based adder all work in the NAM and these adders are all based on an RCA structure. The accuracies of 8-bit and 16-bit ACFA-based adders and previous designs are evaluated by MATLAB using Monte Carlo simulation. One million uniformly distributed random data are considered as input samples.

The proposed ACFA and other configurable full adders are implemented by using the Synopsys HSPICE tool and the layouts of these full adders are generated by the Cadence Virtuoso. The hardware performances in the critical path delay and the average power of adders are simulated by the Synopsys PTPX. The hardware performances are all simulated by using the SMIC 40nm process at the temperature of 25 °C, the operation frequency of 100 MHz and the supply voltages ranging from 1.1 V to 0.5 V. Moreover, buffers are added before input signals and an output load of FO4, the value of which amounts to 2.8-3.2 fF in the SMIC COMS 40nm process is used at the output side. The power consumptions are also considered for power and delay evaluation, because these adders will be utilized to build a larger system with other components [27].

B. Accuracy

An *n*-bit ACA can generate accurate and approximate results, denoted by *R* and *R'*, respectively. A basic error

(a)

metric to characterize the errors of various designs is the error distance (ED), given by $ED = |R' - R|$. The relative error distance (RED) calculated by $RED = |ED/R|$, indicates the relative difference with respect to the accurate result. The mean ED (MED) and mean RED (MRED) [8] are also used to evaluate the accuracy of approximate designs. A smaller MED or MRED indicates a higher accuracy. These two error metrics are given by:

$$MED = \sum_{i=1}^{N} ED_i \cdot P(ED_i)\,, \tag{10}$$

$$MRED = \sum_{i=1}^{N} RED_i \cdot P(RED_i)\,, \tag{11}$$

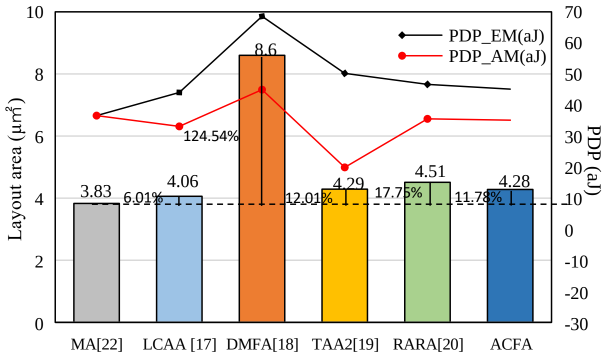


Fig. 8 Layout areas and PDPs of the MA and other configurable full adders

where N is the total number of test samples and the ED and RED of the i^{th} test sample are denoted by ED_i and RED_i, respectively. $P(EDi)$ and $P(MREDi)$ respectively represent the probabilities that ED_i and RED_i occur.

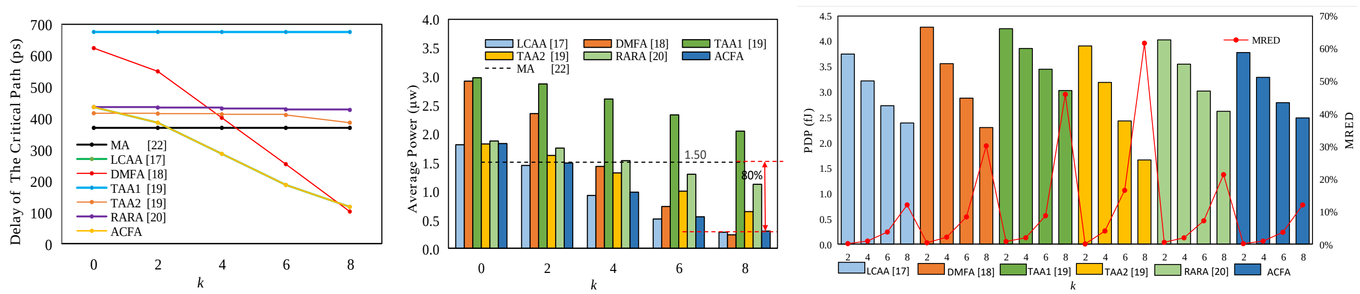
The performance in the MED and MRED of 8-bit and 16-bit ACFA-based adders and other designs [16-20] with various values of k are shown in Fig. 6 and Fig. 7, respectively. Simulation results show that the computing accuracy of our proposed ACFA-based adder and the LCAA-based adder [17] are higher than other designs [18-20]. Because the error rates for both ACFA and LCAA are only 25% and the difference between exact and approximate results is only 1 when errors occur. However, when k is in the range of from $\lfloor 0.5n \rfloor + 1$ to n, the ICS configures fewer full adders to work in an AM, compared with the common configuration scheme. The MED and MRED of the ACFA-based adder are always smaller than that of the LCAA-based adder when k is larger than $\lfloor 0.5n \rfloor + 1$, leading to considerable improvement in accuracy.

As shown in Fig. 6 and Fig. 7, the 16-bit ACFA-based adder obtains the reduction of up to 12.5% MED and 14.39% MRED compared with the LCAA-based adder. The TAA2-based adder has up to 74.99% larger accuracy loss than that of the LCAA-based adder, as the LSBs of the sum in a TAA2 are always locked at ‘0’. The DMFA-based adder has a similar MED to the RARA-based adder, which is slightly larger than that of TAA1-based adder. However, compared with the DMFA-based adder, the TAA1-based and TAA2-based adders have larger MRED, which indicates that they have a higher possibility of producing large error distance.

C. Hardware Performance

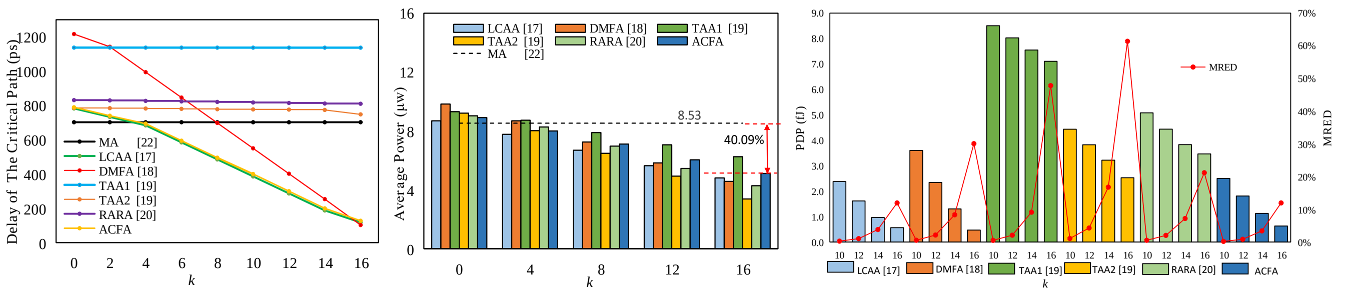
The propagation delay of the critical path, the power dissipation and area cost are three basic metrics for measuring the hardware performance. The power and delay product (PDP) is also considered as a compound metric indicating the energy consumption.

Comparisons of the layout areas and the PDPs of the MA [22], the proposed ACFA and previous full adders [17-20] are presented in Fig. 8. The layout areas and the PDP in the EM of the ACFA and other full adders are all larger than the MA [22], due to the extra transistors added as power gating. Similar to the LCAA [17], the proposed ACFA has a smaller area, among these configurable designs, which is only 11.78% larger than that of the MA. Although The TAA2 [19] has the



(a) The delay of critical paths (b) Average power (c) PDP vs MRED

Fig. 9 The performance of 8-bit DACA with different values of k



(a) The delay of critical paths (b) Average power (c) PDP vs MRED

Fig. 10 The performance of 16-bit DACA with different values of k

TABLE III THE PERFORMANCE IN DELAY AND PDP OF AN EXACT ADDER AND THE TACA WORKING IN THE EM AND AM

Voltage (V)	Exact adder		TACA-EM		TACA-AM	
	Delay (ps)	PDP (fJ)	Delay (ps)	PDP (fJ)	Delay (ps)	PDP (fJ)
1.1	369.43	4.070	435.77	4.550	236.98	2.975
1.0	387.90	2.768	457.56	3.094	248.83	2.023
0.9	429.29	0.810	506.38	0.905	275.38	0.592
0.8	463.62	0.204	546.87	0.228	297.40	0.149
0.7	487.32	0.041	574.83	0.046	312.60	0.030
0.6	657.61	0.028	775.70	0.032	421.84	0.021
0.5	745.47	0.012	879.34	0.014	478.20	0.009

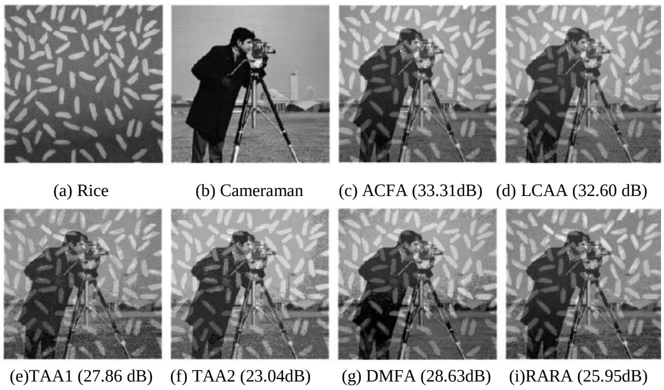
largest reduction of PDP in the AM compared with that of the MA, the accuracy of the TAA2 is comparatively lowest, as discussed in Section V. B. When working in the EM, the proposed ACFA only has a slightly larger PDP than the LCAA and the MA. When working in the AM, the proposed ACFA has a 5.16% energy saving compared with the MA.

The hardware performance of the proposed ACFA-based adder and other ACAs are simulated to comprehensively analyze the trade-off between hardware performance and accuracy. The critical path delay, average power and PDP of 8-bit and 16-bit adders with different values of *k* are presented in Fig. 9 and Fig. 10, respectively. For the EM where k is equal to 0, it is inevitable to introduce extra hardware for realizing accuracy configuration. Among these configurable designs, the proposed ACFA-based adders have lower extra costs of circuits, which are slightly larger than the LCAA-based adders [17]. The proposed ACFA-based adder only has an increase of 2.01% in power and 17.95% in critical path delay compared with the accurate adder, while the DMFA-based adder has the increase of 14.98% in power and 68.89% in delay.

As *k* increases, both the power and delay of these adders will proportionally decrease because of the simplification of circuits and the shortened critical paths. The ACFA-based adder provides a significant reduction in delay, resulting from cutting off the carry chains of approximate ACFAs. Moreover, the sum and carry blocks of *m* full adders are shut off or simplified, leading to the reduction of 80% and 40.09% in average power compared with 8-bit and 16-bit exact adders, respectively. Thus, compared with the exact adders, up to 80.05% and 89.43% of energy in PDP are saved for 8-bit and 16-bit ACFA-based adders, respectively.

Comparisons of the MRED and PDP of these adders are presented in Fig. 9 (c) and Fig. 10 (c). Generally, both 8-bit and 16-bit ACFA-based adders show strong competitiveness with a remarkable trade-off between PDP and accuracy (in the MRED). Although these 8-bit adders with the same *k* have the similar performance in PDP, the proposed ACFA-based adder has the smallest MRED, resulting from using the ICS. Among these16-bit adder, the ACFA-based adder and the LCAA-based adder show a better performance in reducing energy consumption. The TAA2-based adder [19] achieves the largest power saving when *k* is larger than 8. However, it cannot reduce much in the PDP, since the delay of this adder is not shortened, as shown in Fig. 10 (c).

The performance in critical path delay and PDP of 8-bit TACA working in the EM and an AM with *m* equal to 1 and 8-bit exact adder are presented in TABLE III. As the supply voltage scales down from 1.1 V to 0.5 V, the energy consumptions of both the TACA and exact adder reduce, although the critical delay increases. The PDP of the TACA working in the EM is larger than that of the extra adder, resulting from the energy of the extra transistors and the TEDC units. However, the PDP of the TACA working in the AM is significantly smaller than that of the extra adder, benefiting from cutting off the carry chain and simplifying the operation of this adder in the AM. When the voltage reduces from 1.1 V to 0.5 V, the TACA working in the AM reduces average power by 99.83% and increases the critical path delay by only 29.44%, resulting from cutting off the carry chain. Thus, it obtains a reduction in PDP of up to 99.77% at the voltage of 0.5 V, compared with the exact adder working at the voltage of 1.1 V.


(a) Rice (b) Cameraman (c) ACFA (33.31dB) (d) LCAA (32.60 dB)

(e)TAA1 (27.86 dB) (f) TAA2 (23.04dB) (g) DMFA (28.63dB) (i)RARA (25.95dB)

Fig. 11 Results of image addition: Cameraman + Rice

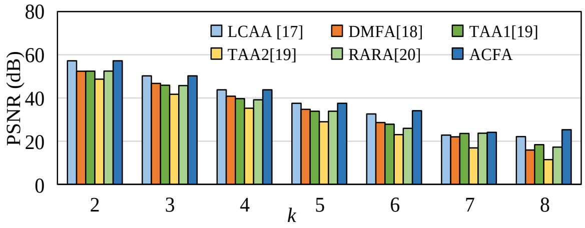


Fig. 12 PSNR results of image addition by using 8-bit adders with different values of *k*

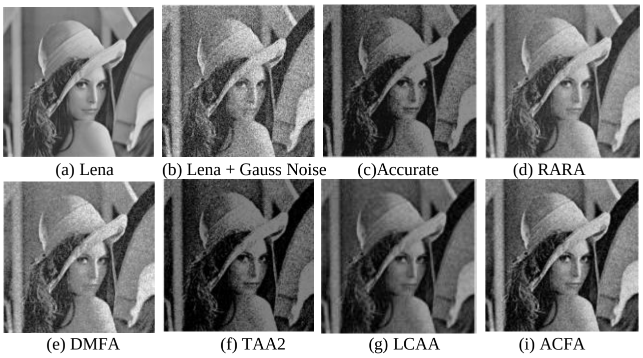
VI. APPLICATION

A. *Image processing*

To evaluate the improvement in accuracy of the proposed design using the ICS, 8-bit and 16-bit ACFA-based adders are applied in image processing applications: addition, filtering and compression. The output quality of image processing is usually proportional to the value of the peak signal-to-noise ratio (PSNR). The structural similarity (SSIM) [26] correlated with the quality perception of the human visual system is also considered as an alternative method to evaluate output qualities of image processing.

1) *Image Addition*

As a basic operation, image addition is widely used to perform many tasks, for example imaging masking and enhancing [29]. The image addition is performed usually by adding two pixels at the same position in two original images and then generating a new pixel. The classic images: Rice and Cameraman with the same size of 512×512, are considered


(a) Lena (b) Lena + Gauss Noise (c)Accurate (d) RARA

(e) DMFA (f) TAA2 (g) LCAA (i) ACFA

Fig. 13 Results of image filtering

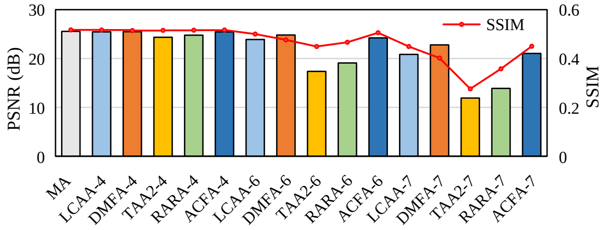


Fig. 14 Output qualities of image filtering by using various 8-bit adders with different values of *k*

as a case study, in which 512^2 addition operations are required.

Compared with using exact adders, the differences between the images processed by the proposed ACFA-based adder and the adder in [17] are nearly indistinguishable visually. However, the images processed by other designs [18-20] produce varying degrees of visible distortions, as shown in Fig. 11. Fig. 12 gives the PSNR results of using 8-bit exact adders and ACAs with different values of *k* (ranging from 2 to 8). The proposed ACFA-based adder and the LCAA-based adder [17] always produce the highest PSNR, with the same values of *k* as other ACAs [18-20]. The TAA2-based adder results in the lowest PSNR and worse output quality because of its considerably large MED and MRED. The image addition by using the ACFA-based adders has the same PSNR as that using the LCAA-based adder when the value of *k* is smaller than 6; otherwise, the proposed adder has a higher PSNR.

2) *Image Filtering*

To eliminate Gaussian noise, arithmetic mean filtering is usually applied in image smoothing [30]. Let a classic image: Lena as a base with the size of 512×512, as shown in Fig. 13 (a). A Gaussian noise with a mean of 0 and a standard deviation of 0.02 is added into this base. The base image denoted by a is filtered by an $m \times m$ mean filter. Assume $b(x, y)$ represents the pixel at the *x* row and the *y* column of the new image *b*, it is computed by m^2 addition operations, as:

$$b(x, y) = \frac{1}{m \cdot m} \sum_{j=-1}^{1} \sum_{i=-1}^{1} a(x+1, y+1) \qquad (12)$$

A mean filter with *m* equal to 3 is applied for filtering the introduced noise by using 8-bit exact adders, the ACFA-based adders and other ACAs [17-20].

When the value of *k* is smaller than 6, these adders have almost similar PSNR and SSIM values, as shown in Fig. 14. Otherwise, among these adders, the DMFA-based adder [18] has the highest PSNR, because of the even distribution of its errors. However, the ACFA-based adder still has the highest SSIM when *k* is larger than 4, because of its smaller MRED among these adders. The ACFA-based adder and the LCAA-based adder always produce similar output qualities of image filtering.

3) *Image Compression*

Image compression is usually realized by two processes, the discrete cosine transform (DCT) and inverse DCT (IDCT) [31]. A 2-D original image is divided into some *m* × *m* blocks, When these blocks in the original image are compressed in parallel, the speed for image compression will increase as *m* decreases. Let *C* represents the coefficient matrix and *I* represent the pixel matrix of one block. In matrix notation, a 2-D DCT and IDCT transformations are respectively expressed as:

$$O = C \cdot I \cdot C^T , \qquad (13)$$

$$I^{'} = C^T \cdot O^{'} \cdot C , \qquad (14)$$

where *O* represents the resulting matrix of DCT transformation, and *O*' contains the quantified and encoded DCT outputs. *I*' is the compressed pixel matrix of an input block. As a result, $4m^2(m\text{-}1)$ addition operations are required to compute a set of DCT and IDCT transformations.

Using 16-bit exact adders to compress the 256×256 image Lena is considered as a baseline. In this base case, *m* and the compression ratio are set as 4 and 37.5%, respectively. Fig. 15 provides the compressed images by using 16-bit ACFA-based adders and other ACAs [17-20] with the value of *k* set as 10. The output quality increases when *m* decreases from 4 to 2, as shown in Fig. 15 (f) and (g). The women in the images compressed by the ACFA-based adder and the proposed LCAA-based adder [17] can still be clearly identified. However, the output qualities of the images compressed by using the ACAs [18-20] have a significant degradation. That indicates the superior accuracy of the ACFA-based adder and the LCAA-based adder. Moreover, benefiting from the ICS, the image compressed by using the proposed ACFA-based adder is slightly clearer than that using the LCAA-based adder.

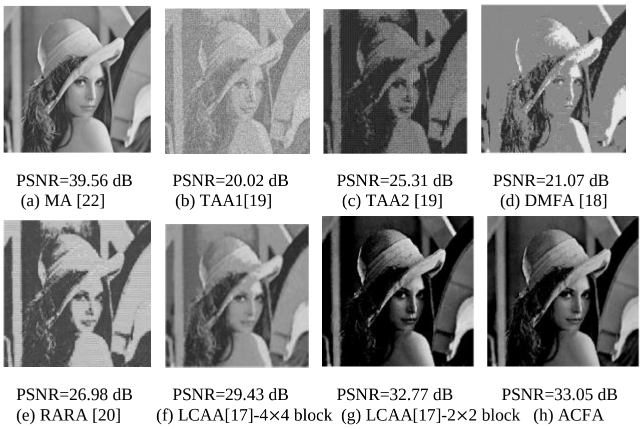

PSNR=39.56 dB (a) MA [22]
PSNR=20.02 dB (b) TAA1[19]
PSNR=25.31 dB (c) TAA2 [19]
PSNR=21.07 dB (d) DMFA [18]
PSNR=26.98 dB (e) RARA [20]
PSNR=29.43 dB (f) LCAA[17]-4×4 block
PSNR=32.77 dB (g) LCAA[17]-2×2 block
PSNR=33.05 dB (h) ACFA

Fig. 15 Results of image compression by using different 16-bit adders with the value of *k* equal to 10

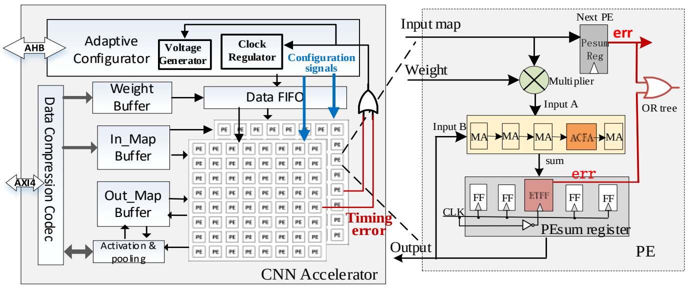


Fig. 16 The structure of the CNN accelerator

B. *CNN-based Classification*

A convolutional neural network (CNN) based on the classic LeNet-5 model [32] is built for digit classifications. The weight and bias parameters of this CNN are trained in Python by using 10000 images in the MNIST dataset [33], then obtaining 98.73% classification accuracy by using exact adders. The input and weight sizes and the numbers of addition operations on each layer of this CNN are shown in TABLE IV.

The structure of the CNN accelerator using the proposed TACAs is presented in Fig. 16. This accelerator circuit consists of two 8 × 8 processing element (PE) arrays, external and internal memory units that include the input map buffers, output map buffers and weight buffers, advanced high performance bus (AHB) and an adaptive configuration unit. Each PE block is composed of the input and output registers, a multiplier and a 16-bit adder constructed by using the proposed TACA. The output register, namely, the PEsum register, is built by cascading some traditional flip-flops (FFs).

A TEDC unit based on the ETFF [28] is inserted in the circuit of data paths by replacing an original FF of the PEsum register. As the supply voltage is overscaled or the operation frequency is aggressively increased, the propagation delay will significantly increase. The TACA will immediately predict the timing violations that might occur and configure the ACFA to work in an AM. This effectively shortens the delay and leads this CNN accelerator to run at a lower voltage without reducing the throughput of the system.

The hardware implementation of the CNN accelerator circuit is performed by using the SMIC 40 nm process. The hardware prototype of this accelerator is implemented in RTL Verilog and synthesized using the Synopsys Design Compiler. The energy efficiency of this accelerator is significantly improved by using the TACAs, although a slight accuracy loss of 3.07% to 10.6% inevitably incurs at the supply voltages ranging from 1.1 V to 0.5 V. Up to 49.37% reduction of the PDP is obtained and the operation frequency of up to 800 MHz is realized at the supply voltage of 1.1 V. As the supply voltage reduces to 0.5 V, the propagation delay of the critical path increases by 28.73%, compared with that at the voltage of 1.1 V. However, the average power reduces by 98.99%, leading to the reduction in PDP of 99.71%. Compared with using exact adders, the critical path delay of computing circuits in the CNN accelerator reduces by 27% at a voltage of 0.5 V, due to the cut-off carry chain in the TACA. Thus, the highest operation frequency is increased to 995.92 MHz, without decreasing the throughput at the low voltage of 0.5 V.

VII. CONCLUSION

In this paper, a transistor-level accuracy-configurable full adder (ACFA) is proposed by adding four transistors as the power gating into the mirror full adder. The operation of the ACFA is dynamically switched among three modes at runtime. A timing-aware configurable adder (TACA) is further constructed based on the ACFAs and timing-error detection and correction circuits. Moreover, an improved configuration scheme is developed to efficiently reduce the critical path delay and accuracy loss of multi-bit ACFA-based adders. The voltage overscaling technique is introduced to reduce the energy of the TACA. Simulation results indicate that the proposed TACA achieves a significant saving in energy and a remarkable trade-off between hardware performance and accuracy. The use of the proposed designs contributes to the high output qualities obtained in three image processing applications and the high energy efficiency obtained in a CNN accelerator circuit. The trade-off between energy efficiency and accuracy can be improved in future work by investigating the flexibility and further configuration of the ACFA-based adder.

TABLE IV THE STRUCTURE OF LENET-5 AND THE NUMBER OF ADDITION OPERATIONS IN EACH LAYER

layer	Input size	Weight size	# of addition
1	32×32×1	5×5×6	(5×5-1)×28×28×6
2	14×14×6	5×5×6×16	(5×5-1)× 5 ×10×10×16
3	5×5×16	5× 5 ×16×120	(5×5-1)×15×120
4	1×1×120	120×84	119×84
5	1×1×84	84×10	83×10

REFERENCES

[1] Schlachter J., Camus V. and Enz C., "Near/Sub-threshold circuits and approximate computing: the perfect combination for ultra-low-power systems", IEEE Computer Society Annual Symposium on VLSI, 2015, pp. 476-480.

[2] Afzali-Kusha M., Akbari O., Kamal M. and Pedram M., "Energy and reliability improvement of voltage-based, clustered, coarse-grain reconfigurable architectures by employing quality-aware mapping", IEEE J. Emerg. Sel. Top. Circuits Syst., 2018, vol. 8, no. 3, pp. 480-493.

[3] Afzali-Kusha H., Kamal M. and Pedram M., "Low-power accuracy-configurable carry look-ahead adder based on voltage overscaling technique", Proc. - Int. Symp. Qual. Electron. Des., 2020.

[4] De la Guia Solaz M. and Conway R., "Razor based programmable truncated multiply and accumulate, energy-reduction for efficient digital signal processing", IEEE Trans. Very Large Scale Integration Syst., 2015, vol. 23, no. 1, pp. 189-193.

[5] Moons B. and Verhelst M., "An energy-efficient precision-scalable ConvNet processor in 40-nm CMOS", IEEE J Solid State Circuits, 2017, vol. 52, no. 4, pp. 903-914.

[6] Frustaci F., Perri S., Corsonello P. et al., "Energy-quality scalable adders based on nonzeroing bit truncation", IEEE Trans. Very Large Scale Integration Syst., 2018, vol. 27, no. 4, pp. 964-968.

[7] Yin S., Ouyang P., Zheng S. et al., "A 141 uw, 2.46 pj/neuron binarized convolutional neural network based self-learning speech recognition processor in 28nm CMOS", IEEE Symp VLSI Circuits Dig Tech Pap, 2018, pp. 139-140.

[8] Jiang H., Santiago F. J. H., Mo H. et al., "Approximate arithmetic circuits: A survey, characterization and recent applications", Proc. IEEE, 2020, vol. 108, no. 12, pp. 2108-2135.

[9] Kahng A. B., Kang S., "Accuracy-configurable adder for approximate arithmetic designs", DAC, 2012, pp. 820-825.

[10] Shafique M., Ahmad W., Hafiz. R. et al., “A low latency generic accuracy configurable adder”, DAC, 2015, pp. 1–6.

[11] Benara V., Purini S., "Accurus: A fast convergence technique for accuracy configurable approximate adder circuits", Proc. IEEE Comput. Soc. Annu. Symp., on VLSI, 2016, pp. 577-582.

[12] Kanani A., Mehta J. et al., "ACA-CSU: A carry selection based accuracy configurable approximate adder design", Proc. IEEE Comput. Soc. Annu. Symp., on VLSI, 2020, pp. 434-439.

[13] Xu W., Sapatnekar S. S., Hu J., "A simple yet efficient accuracy-configurable adder design", IEEE Trans. Very Large Scale Integration Syst., 2018, vol. 26, no. 6, pp. 1112-1125.

[14] Yang T., Ukezono T. and Sato T., "A low-power configurable adder for approximate applications", Proc. - Int. Symp. Qual. Electron. Des., 2018, pp. 347-352.

[15] Tsai K. L., Chang Y. J. et al., "Accuracy-configurable radix-4 adder with a dynamic output modification scheme", IEEE Trans. Circuits Syst. I, Regular Papers, 2021, vol. 68, no. 8, pp. 3328-3336.

[16] Li H., Fan X. et al., "An efficient light-weight configurable approximate adder design", VLSI-SoC, 2021, pp. 1-6.

[17] Hassani M. M., Rezaalipour M. and Dehyadegari M., "A novel ultra low power accuracy configurable adder at transistor level", Int. Conf. Comput. Knowl. Eng., 2018, pp.165-170.

[18] Raha A., Jayakumar H. et al., "Input-based dynamic reconfiguration of approximate arithmetic units for video encoding", IEEE Trans. Very Large Scale Integration Syst., 2015, vol. 24, no. 3, pp. 846-857.

[19] Jha C. K., Nandi A., Mekie J., "Quality tunable approximate adder for low energy image processing applications", ICECS, 2019, pp, 642-645.

[20] Yin S., Liu Z. et al., "RARA: Dataflow based error compensation methods with runtime accuracy-reconfigurable adder", Proc. - Int. Symp. Qual. Electron. Des., 2020, pp. 60-66.

[21] Das, S., Roberts, D. et al. "A self-tuning DVS processor using delay-error detection and correction, " IEEE J. Solid-State Circuits 2006, 41, 792–804.7.

[22] Morinaka H., Makino H. et al., "A 64bit carry look-ahead CMOS adder using modified carry select", CICC, 1995, pp. 585-588.

[23] Toshinori S., Tomoaki U., "On Applications of Configurable Approximation to Irregular Voltage", NORCAS, 2019.

[24] Mahdiani, H. R., Ahmadi A., Fakhraie S. M., Lucas C., "Bio-Inspired imprecise computational blocks for efficient VLSI implementation of Soft-Computing applications", IEEE Trans. Circuits Syst. I, Regular papers, 2010, vol. 57, no. 4, pp. 850-862.

[25] Zhou, J., Liu, X., Lam, Y.H. et al. "HEPP: A New In-Situ Timing-Error Prediction and Prevention Technique for Variation-Tolerant Ultra-Low-Voltage Designs", A-SSCC, 2013; pp. 129–132.

[26] Fan X. M., Liu H., Li H., Lu S. L., Han J., "Design of light-weight timing error detection and correction circuits for energy-efficient near-threshold voltage operation", Electronics 2022, 11, 2879.

[27] Aguirre-Hernandez M. and Linares-Aranda M., "CMOS full-adders for energy-efficient arithmetic applications", IEEE Trans. Very Large Scale Integration Syst., 2011, vol. 19, no. 4, pp. 718-721.

[28] Wang Z., Bovik A. C, Sheikh H. R. et al. "Image quality assessment: from error visibility to structural similarity", IEEE Trans. Image Process., vol. 13, no. 4, pp. 600-612, 2004.

[29] Almurib H. A. F., Kumar T. N. and Lombardi F., "Inexact designs for approximate low power addition by cell replacement", DATE, 2016, pp. 660-665.

[30] Zhang P. and LI F., "A new adaptive weighted mean filter for removing salt-and-pepper noise", IEEE Signal Process. Lett., 2014, vol. 21, no. 10, pp. 1280–1283.

[31] Suzuki T., Ikehara M., "Integer DCT based on direct-lifting of DCT-IDCT for lossless-to-lossy image coding", IEEE Trans. Image Process., 2010, vol. 19, no. 11, pp. 2958–2965.

[32] Lecun Y., Bottou L. et al., "Gradient-based learning applied to document recognition", Proc. IEEE., 1998, vol. 86, no. 11, pp. 2278-2324.

[33] Lecun Y., Cortes C., Burges C., 2001, MNIST handwritten digit database., http://yann.lecun.com/exdb/mnist

Xuemei Fan received the M.S. degree in Software Engineering from Southeast University, Nanjing, China, in 2018. She is currently pursuing the Ph.D. degree in Microelectronics and Solid State Electronics in Southeast University. Her current research interests include approximate computing, low-power near-threshold integrated circuit and neural network accelerator design.

Tingting Zhang (Graduate Student Member, IEEE) received the B.Sc. and M.Sc. degrees in the College of Electronic and Information Engineering from the Nanjing University of Aeronautics and Astronautics (NUAA), Nanjing, China, in 2016 and 2019, respectively. She is working toward the Ph.D. degree in the Department of Electrical and Computer Engineering, University of Alberta, Alberta, Canada, since Sep. 2019. Her research interests include approximate computing, Ising computing, combinatorial optimization, and nanoelectronic circuits and systems.

Hao Liu received the B.S. degree from Xi'an Jiaotong University, Xi'an, China, in 1996 and the Ph.D. degree from Southeast University, Nanjing, China, in 2011. He is currently an Associate Professor with Southeast University. His research interests include wireless sensor networks, data processing machine (DPM), and energy efficient SoC system.

Shengli Lu received the B.S. degrees in information and physics from Nanjing University, Nanjing, China, in 1987 and Ph.D degree in Microelectronics in Southeast University, in 2007. His current research interests include very large-scale integration (VLSI), energy efficient computing, artificial intelligence algorithms and accelerators circuit design.

Jie Han received the B.Sc. degree in electronic engineering from Tsinghua University, Beijing, China, in 1999 and the Ph.D. degree from the Delft University of Technology, Netherlands, in 2004. He is currently a Professor and Director of Computer Engineering in the Department of Electrical and Computer Engineering at the University of Alberta, Edmonton, AB, Canada. His research interests include approximate computing, stochastic computing, reliability and fault tolerance, nanoelectronic circuits and systems, novel computational models for learning and biological applications.