

A power and area efficient approximate carry skip adder for error resilient applications

Sujit Kumar PATEL*, Bharat GARG, Shireesh Kumar RAI

Electronics and Communication Engineering, Thapar Institute of Engineering and Technology,
Patiala, Punjab, India

Received: 11.07.2019

Accepted/Published Online: 11.09.2019

Final Version: 27.01.2020

Abstract: The compute-intensive multimedia applications on portable devices require power and area efficient arithmetic units. The adder is a prime building block of these arithmetic units and limits the overall performance. Therefore, this paper analyzes the logic operations of the state-of-the-art adders and presents a novel low complexity adder segment with new carry prediction logic by removing the redundant logic and sharing the common operations. Further, a new power and area efficient approximate carry skip (PAEA-CSK) adder is proposed using the novel adder segment. The effectiveness of the proposed PAEA-CSK adder is evaluated and compared over the existing adders by implementing them in VHDL and synthesizing using the Synopsys Design Compiler with the 65nm TSMC CMOS Library. The synthesis result shows that the proposed PAEA-CSK adder requires 27.28% and 18.03% less area and power, respectively, over the existing carry skip-based approximate adder with the same accuracy. Further, the Sobel edge detector (SED) embedded with the proposed adder improves PSNR by a minimum of 16.94 dB over the SED embedded with a non-zeroing bit-truncation adder.

Key words: Approximate computing, carry skip adder, low-power design, error resilient applications

1. Introduction

The exponential growth of multimedia applications on modern portable devices demands highly power efficient and low complexity processing units. The conventional design techniques exhibit trade-off between area, power, and delay, where improving one metric degrades the other. Therefore, these design techniques fail to improve all the parameters simultaneously. Several applications, such as multimedia and big data analysis, exhibit error resiliency in abundance and do not require 100% accurate results [1]. Relaxing the computational accuracy with acceptable quality may significantly reduce the area/power consumption. Therefore, approximate computing has emerged as a new design paradigm for these applications and showed remarkable improvement in performance. Accuracy is considered a new trade-off parameter in these applications to achieve improved design metrics. In various signal processing systems, the arithmetic unit consumes significant amounts of area/power and determines the performance of the design. The adder is not only the key arithmetic unit in different signal processing systems; it is also used to implement other units such as multiplier, divider, incrementer, and decrementer. Therefore, the design of a power and area efficient adder is the critical requirement to achieve efficient signal processing systems.

Several design techniques have been reported to achieve high performance adder architectures. These techniques can be classified based on computational accuracy, namely accurate, approximate, and accuracy-

*Correspondence: sujit.patel@thapar.edu

configurable design. The accurate design techniques exploit resource sharing and/or redundant logic elimination, whereas approximate design techniques identify and compute the sum of nonsignificant parts using approximate computing [2]. The accuracy-configurable designs are nothing but approximate designs with error detection and correction (EDC) logic to achieve the desired accuracy–performance trade-off. In accurate adder designs, the ripple carry adder (RCA) is the simplest design but exhibits large carry propagation delay. Therefore, several high speed adders are presented in the literature [3]. For example, the carry look-ahead (CLA) adder computes carry signals in advance to reduce delay but shows large area overhead. Further, the carry select adder (CSLA) [4, 5] and carry skip adder (CSKA) [6] provide reduced delay with additional area and power consumption. On the other hand, to achieve an area and power efficient design, approximate adders are implemented either using a simplified full adder [7] or by designing approximate sum logic for few least significant bits (LSBs) [8, 9]. The approximate sum logic is designed such that overall error is minimized. Further, a generic approximate carry skip adder (ACSKA) in which previous v -segments are used to predict the carry-in for the current segment is presented in [10]. The consideration of more segments for carry prediction improves the accuracy at the cost of large delay and area. Although the approximate adders provide improved design metrics with acceptable amount of error, they require large redesign efforts to achieve an approximate design with different accuracy–performance trade-off.

There are several applications that require different accuracy under different conditions; for example, a security camera requires clearer images once the object is detected [11]. For these applications, the accuracy of the adder should be reconfigurable during run-time to achieve maximum power benefits with acceptable quality. An accuracy configurable adder (ACA) [11] computes the approximate sum using subadders and provides accurate results by adding the error value to the next pipelined stages. Therefore, it provides trade-off between computational latency and accuracy. A reconfigurable approximate CLA (RAP-CLA) adder [12] considers fixed w terms (window) of carry expressions (conventional CLA) for carry generation during approximate mode, while conventional CLAs carry expressions in accurate mode. Although the RAP-CLA shows high performance, it requires large power and area overhead due to the use of multiplexers for switching between accurate and approximate mode. Finally, a scalable nonzeroing bit-truncation (NzBt) adder is presented in [13] that can switch between accurate and approximate mode using the control signal. In approximate mode, the NzBt fixes certain LSB of input operands to complementary values, which leads to output sum bits with constant logic ‘1’.

Among the recent approximate adders, the approximate carry skip (CSK) adder provides better trade-off between area, delay, and power performance. However, the existing CSK-based approximate adder exhibits redundant logics in its design. Therefore, it could be possible to develop an area and power efficient CSK-based approximate adder by identifying and eliminating redundant logics. The key contributions are as follows:

- An analysis of state-of-the-art CSK adders on the basis of logic complexity is presented.
- A novel low-complexity approximate adder segment with new carry prediction logic is proposed by removing the redundant logic operations.
- A new power and area efficient approximate carry skip (PAEA-CSK) adder is presented using the proposed adder segments.

The rest of the paper is organized as follows: section 2 presents related work to achieve low power approximate/accuracy-configurable adders. Section 3 presents a comprehensive analysis of existing adders based on the logic complexity. The low complexity CSK adder segment is presented in section 4, while the

proposed area and power efficient adder is presented in section 5. Section 6 presents the simulation environment and comparative analysis of the proposed adder over the existing. Finally, section 7 concludes the paper.

2. Related work

A low complexity arithmetic unit is the essential requirement of all modern devices to achieve efficient signal processing. Since the adder is the basic building block of any arithmetic unit, several accuracy configurable adder architectures are reported in the literature. Based on the computational accuracy, our review is broadly divided into three categories, namely accurate, approximate, and accuracy-configurable adders, and is presented in the following subsections.

2.1. Accurate adder architectures

Among the several accurate adder architectures, RCA is the simplest and shows linear area and delay characteristic with bit-width. The worst case delay occurs when carry propagates from LSB to most significant bit (MSB), making this design the slowest one. Various high speed adders, namely CLA, CSKA, and CSLA, are presented to reduce the delay. The CLA precomputes carry-in signals for each intermediate stage of addition and thus eliminates carry propagation. The combinational logic for generating carry-in is bulky for more than four bits. The CSKA reduces the carry propagation delay by implementing architecture such that input carry will be either killed in the segment or skipped when the propagate condition is asserted. In the CSLA, partial sum is computed under the assumption of carry-in as logic '0' and '1'. The carry-in signal is used to select the correct sum and output carry instead of computing it, thus significantly reducing the delay of the adder. Although several redundant logic elimination techniques are presented to reduce the area of these adders [4-6], the area overhead of these high speed adders is very large, making these designs area/power inefficient. The following subsection presents approximate adders that leverage relaxation in computational accuracy to achieve power/area efficient designs.

2.2. Approximate adder architectures

Most of the approximate adders are implemented by either using an approximate full adder (FA) or employing specific logic to generate an approximate sum for a few LSBs. In [7], five different approximate FAs are presented by selectively eliminating some transistors from accurate mirror adders. In [14], XOR/XNOR-based approximate FAs with reduced node capacitances and power consumption are presented. These FAs are then used at some LSBs to design a large bit-width approximate adder. On the other hand, Zhu et al. [8] presented an error tolerant adder (ETA) where inputs are divided into two parts: an upper part containing a few MSBs and a lower part containing the remaining LSBs. The conventional accurate adder is used to compute the sum of MSBs to reduce the amount of error, while simplified logic is used to compute the sum of LSBs. It reduces the power and delay significantly at the cost of large error when small inputs are added. In [9], some OR gates compute an approximate sum for the lower part to reduce the delay at the cost of large error. To reduce the amount of error, a lower part constant-OR adder (LOCA) is presented in [15]. The LOCA systematically computes an approximate sum by employing constant logic '1' for a few LSBs, OR gates for the next few MSBs, and an accurate logic for the remaining MSBs. Recently, a reverse carry propagate FA is presented in [16], where the significance of the carry-in is more than that of the carry-out. A process tolerant adder is presented in [17] that provides low power and high performance even under high process variation. Along with these techniques, various segment-based approximate adders are also presented. These adders compute an approximate sum for each segment using subadders and consider carry-in for each subadder to a few previous segment(s). ETA-II,

ETA-IIM [18], and carry skip approximate adders [10, 19] are based on segmentation that truncates carry propagation. Further, the probabilistic error analysis of these segment-based adders is presented in [20, 21]. To increase the applicability of approximate designs, various accuracy configurable architectures are also presented, which are reviewed in the next subsection.

2.3. Accuracy configurable architectures

In the direction of variable accuracy design, most of the architectures employ an approximate adder to compute the sum and with EDC logic to achieve golden output whenever required. The generic accuracy configurable adder (GeAr) [22] changes the inputs to the subadders and reevaluates the partial sums to achieve output with higher accuracy. Further, five variable latency speculative adder (VLA) architectures based on a high performance parallel prefix adder are presented in [23]. The VLA reduces large errors using simple EDC logic, which occurred in 2's complement addition. A new carry maskable FA (CMFA) is presented in [24], which is then utilized at a few LSBs to design a carry maskable adder (CMA). Although the CMA reduces dynamic power consumption over the conventional FA, it delivers very high error rate. An iterative accuracy programmable adder is proposed in [25] that can reconfigure the probability of getting correct output. A high speed CSLA architecture that provides improved power-delay product (PDP) using a Manchester carry chain is presented in [26].

Recently, a simple accuracy reconfigurable adder (SARA) was presented in [27] that divides the operands into multiple smaller-sized segments and computes the approximate sum using predicted or accurate carry-in. In the SARA, the predicted carry-in is generated by the previous one bit and therefore provides a large error rate when operated in approximate mode while exhibiting poor performance over the RCA when operated in accurate mode. Finally, a quality scalable non-zeroing bit-truncation (NzBt)-based adder is presented in [13], where additional control logic is added at a few LSBs to force input operands into complementary values. Although the NzBt adder reduces power consumption when operated in approximate mode, it requires a significantly large area and consumes more power over the exact RCA when operated in accurate mode. Recently, a RAP-CLA [12] was presented in which the longest carry propagation path is truncated when operated in approximate mode. However, the complexity of the RAP-CLA is large, which makes it power and area inefficient.

From the literature review, it is observed that RCA-based approximate adders show high power and area efficiency at the cost of large delay. On the other hand, CLA-based approximate adders provide higher performance with small area/power penalty. The approximate adders based on the CSK technique provide moderate design parameters between RCA and CLA. Therefore, to achieve high performance design, in the present paper CSK-based adders are reviewed in detail and novel architecture is presented.

3. Analysis of state-of-the-art carry skip adders

From the literature review, it can be seen that the adders presented in [6] and [10] are the best available CSK scheme-based accurate and approximate adders, respectively. Detailed analysis of the logic operations involved in these adders is required for the identification of redundant logics to reduce the implementation complexity. These adders are built by cascading the small-size adders (called segments) of identical or variable sizes. Therefore, an m -bit adder segment is considered for the analysis. This section presents an analysis of the accurate adder segment followed by the approximate adder segment.

3.1. Analysis of the accurate adder segment

An m -bit j^{th} adder segment of the CSK adder presented in [6] is shown in Figure 1. It consists of an m -bit RCA, an incrementation block, and an output carry generation logic. The RCA computes the intermediate sum ($s^0 = \{s_{m-1}^0, s_{m-2}^0, \dots, s_0^0\}$) and carry-out (c_{out}^j) signals using m -bit inputs (x and y). An incrementation block computes the final sum bits ($s = \{s_{m-1}, s_{m-2}, \dots, s_0\}$) from the intermediate sum signals and input carry c_{in}^j . The final output carry signal (c_{in}^{j+1}) is generated in complement form using AND and AOI gates.

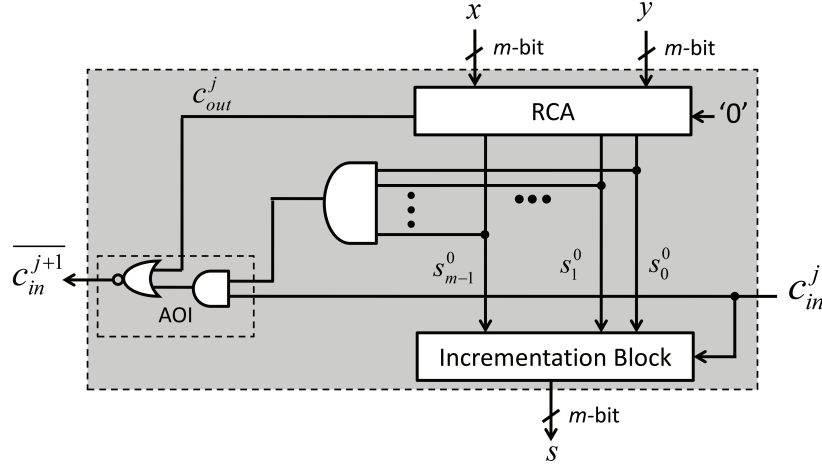


Figure 1. The j^{th} segment of the accurate adder [6].

The logic operations performed in this adder segment are as follows:

Logic operations performed in RCA:

$$p_i = x_i \oplus y_i; \quad g_i = x_i \cdot y_i \quad (1a)$$

$$c_i = g_i + p_i \cdot c_{i-1}; \quad c_{-1} = 0 \quad (1b)$$

$$c_{out}^j = c_{m-1} \quad (1c)$$

$$s_i^0 = p_i \oplus c_{i-1} \quad (1d)$$

Output carry generation operation:

$$\overline{c_{in}^{j+1}} = c_{out}^j + c_{in}^j \cdot \prod_{i=0}^{m-1} s_i^0 \quad (1e)$$

Logic operations performed in the incrementation block:

$$z_i = s_i^0 \cdot z_{i-1}; \quad z_{-1} = c_{in}^j \quad (1f)$$

$$s_i = s_i^0 \oplus z_{i-1}, \quad (1g)$$

where $0 \leq i \leq m - 1$. From above equations, we observed that:

- i The computation of the i^{th} sum bit requires three XOR operations (first computes propagate (p_i), second computes intermediate sum (s_i^0), and the third computes final sum (s_i)).

- ii Similarly, the output carry requires two XOR operations (first computes propagate (p_i), whereas the second computes intermediate sum (s_i^0)) and AND-OR-Invert operation.

If we use a propagate signal to generate sum and carry bits in place of the intermediate sum, it will save m -XOR operations in the m -bit adder segment. Therefore, the existing adder [6] exhibits redundant operations that can be eliminated to achieve an adder segment with low-logic complexity. The next subsection presents an analysis of the existing approximate adder segment.

3.2. Analysis of the approximate adder segment

An m -bit j^{th} adder segment [10] is shown in Figure 2. It consists of a subcarry generator (SCG), multiplexer selection logic (MSL), multiplexer (MUX), and m -bit subadder (SA). The SCG produces output carry (c_{out}^j) and a group propagate (P^j) signal. The MSL generates select signals for the MUX to select the desired carry out using group propagate signals ($P^j, P^{j-1} \dots$) generated by SCGs. The subadder produces sum bits using m -bit inputs (x and y) and carry in (c_{in}^j) that comes from the previous segment. The logic operations performed in the j^{th} SCG and subadder are given by (2).

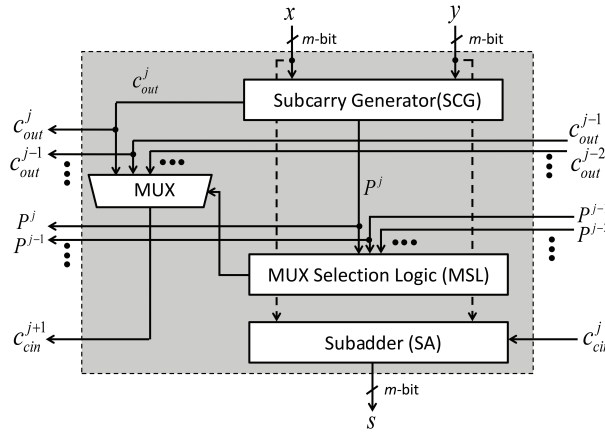


Figure 2. The j^{th} segment of the approximate adder presented in [10].

Logic operations performed in SCG:

$$p_i = x_i \oplus y_i; \quad g_i = x_i \cdot y_i \tag{2a}$$

$$P^j = p_{i-1} \cdot p_{i-2} \dots p_0 \tag{2b}$$

$$c_{out}^j = g_{i-1} + p_{i-1} \cdot g_{i-2} + \dots + p_{i-1} \dots p_1 \cdot g_0 \tag{2c}$$

Logic operations performed in SA:

$$p_i = x_i \oplus y_i; \quad g_i = x_i \cdot y_i \tag{2d}$$

$$c_i = g_i + p_i \cdot c_{i-1}; \quad c_{-1} = c_{in}^j \tag{2e}$$

$$s_i = p_i \oplus c_{i-1}, \tag{2f}$$

where $0 \leq i \leq m - 1$. It can be observed that (2a) and (2d) are identical. Therefore, one set of propagate and generate operations can be removed and the other set can be shared between SCG and SA, which saves

m -XOR and m -AND operations. The MSL circuit and carry selection MUX are different for the different values of v (representing number of carry-out signals generated by SCGs used for carry prediction). Although the microarchitecture of the MSL and MUX is not given in [10], the implementation corresponds to the given functionality as shown in Figure 3 for $v = 2, 3$, and 4. From Figure 3, it can be seen that the logic circuit of MSL is different for different value of v , i.e. there is no regularity. If the MSL and carry selection MUX are designed with the regular structure, the resultant design would eliminate the irregularity and would provide a low complexity design.

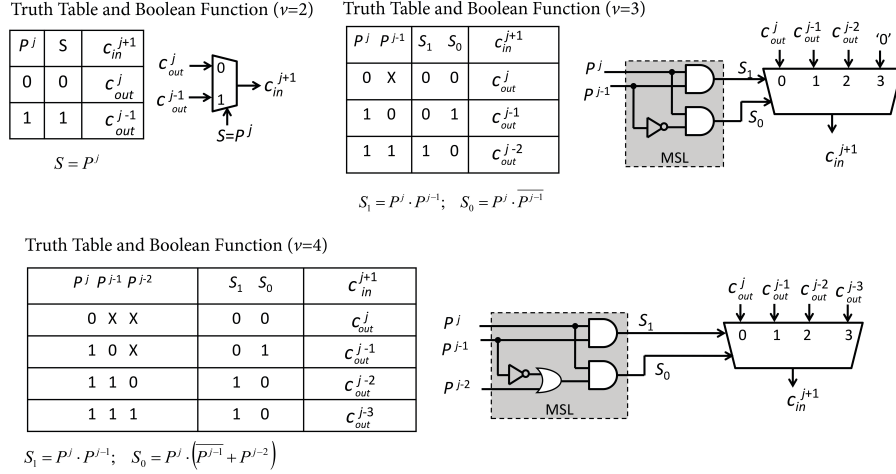


Figure 3. MSL circuit and carry selection multiplexer of the approximate adder segment [10] with a truth table for $v = 2, 3$, and 4.

4. Proposed low-complexity approximate adder segment

The proposed low-complexity approximate carry-skip adder segment shown in Figure 4 is derived on the basis of the analysis presented in the previous section. It consists of a subcarry generation cum propagate-generate (PG) unit, final sum-generation (FSG), unit and a new carry prediction logic (CPL) called New-CPL. The SCG cum PG unit computes the propagate ($p = \{p_{m-1}, p_{m-2}, \dots, p_0\}$) and generate ($g = \{g_{m-1}, g_{m-2}, \dots, g_0\}$), group-propagate (P^j), and carry-out (C_{out}^j) signals from the m -bit inputs (x and y). The final sum ($s = \{s_{m-1}, s_{m-2}, \dots, s_0\}$) is produced by the FSG unit using p , g , and C_{in}^j signals. The SCG cum PG and FSG units perform the following logic operations:

Operations performed in SCG cum PG unit:

$$p_i = x_i \oplus y_i; \quad g_i = x_i \cdot y_i \tag{3a}$$

$$P^j = \prod_{i=0}^{m-1} p_i \tag{3b}$$

$$C_{out}^j = g_{i-1} + p_{i-1}g_{i-2} + \dots + p_{i-1} \dots p_1g_0 \tag{3c}$$

Operations performed in FSG unit:

$$c_i = g_i + p_i \cdot c_{i-1}; \quad c_{-1} = C_{in}^j \tag{3d}$$

$$s_i = p_i \oplus c_{i-1}, \tag{3e}$$

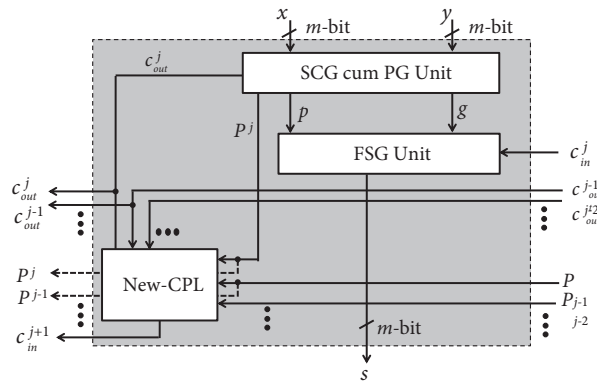


Figure 4. Proposed approximate adder segment with the new carry prediction logic.

where $0 \leq i \leq m - 1$. The operation of New-CPL is described in the truth tables shown in Figure 5 for different values of v . The logic circuits are developed for New-CPLs according to their operations (given in truth table) using 2-to-1 multiplexer(s) as shown in Figure 5 for $v = 2, 3$, and 4. Further, it is observed that if group propagate output bit (P^j) of the SCG cum PG unit is set to logic ‘1’, its carry-out bit (c_{out}^j) will always be logic ‘0’, i.e. if $P^j = 1$ then $c_{out}^j = 0$. Therefore, the design of CPL by exploiting this property requires only one AND and one OR gates in place of a 2-to-1 multiplexer, which provides the saving of one AND and one NOT gates. It is clear that the proposed New-CPLs exhibit low-logic complexity with a regular structure over the carry prediction logic of [10]. The logic diagram of the proposed adder segment is shown in Figure 6.

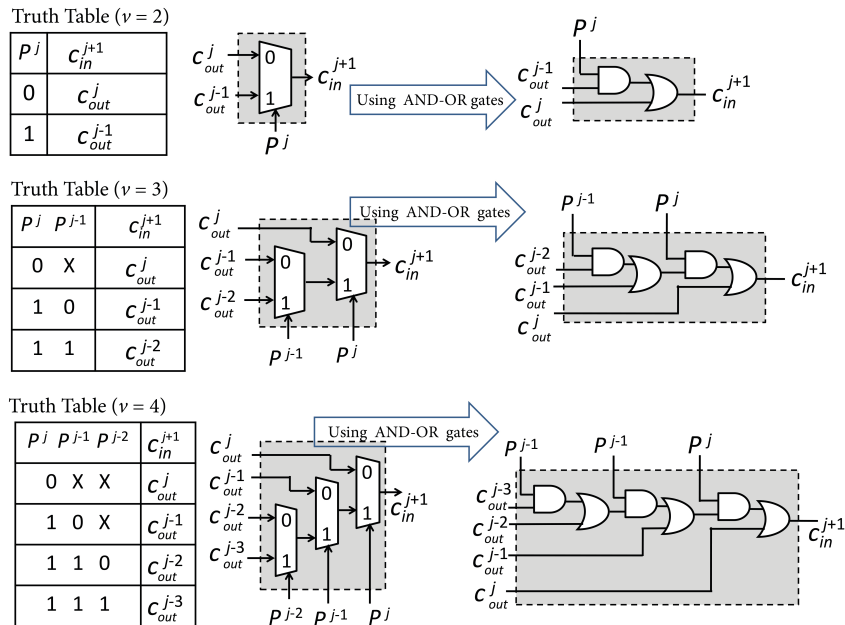


Figure 5. Truth tables and logic circuits of New-CPL for $v = 2, 3$, and 4.

This circuit is derived using Boolean expressions given by (3) and AND-OR implementation of the proposed New-CPL. It can be observed from Figure 6 that the proposed adder segment has low-logic complexity and exhibits structural regularity. Although in Figure 6 the New-CPL is implemented for $v = 4$, it can be extended

to any value of v by addition/deletion of AND-OR logic. The logic complexity analysis of the proposed adder segment is presented in the next subsection.

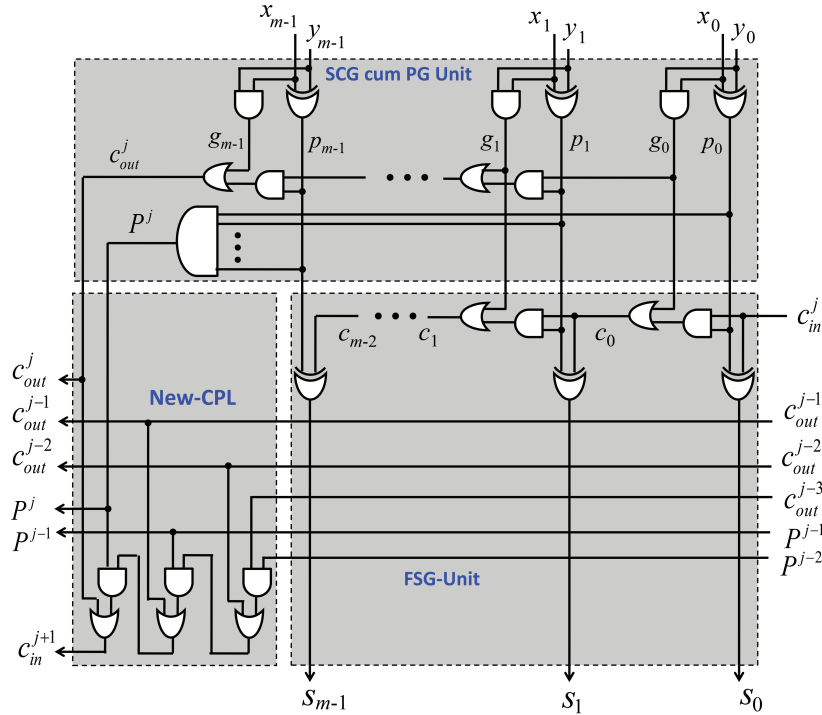


Figure 6. Logic diagram of the proposed j^{th} approximate adder segment with New-CPL ($v = 4$).

4.1. Theoretical analysis of the logic complexity

For the theoretical logic complexity analysis, we have taken area of 2-input AND, 2-input OR and NOT, 2-input XOR, and 3-input AOI gates from the TCBN65GPLUS TSMC 65 nm core library databook [28]. The areas of AND, OR, NOT, XOR, and AOI are 6, 6, 2, 12, and 6, respectively, in terms of transistor count. Based on the transistor count, the area estimation relation is given by (4).

$$Area = 6N_a + 6N_o + 2N_i + 12N_{xor} + 6N_{aoi}, \quad (4)$$

where N_a , N_o , N_i , N_{xor} , and N_{aoi} represent the gate counts of AND, OR, NOT, XOR, and AOI gates, respectively. Using (4), the total area of each design can be calculated by using the total gate count in the design. The generalized gate counts of m -bit proposed and existing adder segments are given in Table 1. Using the gate count given in Table 1 and (4), we have calculated the area of the proposed and existing adder segments for 8, 16, and 32 bits, and the calculated values are presented in Table 2 for comparison. Table 2 shows that the adder segments proposed in [10] and [6] require 30.15% and 8.7% more area over the proposed adder segment. Consequently, it is expected that the approximate adder built from the proposed adder segment will occupy less area and consume less power over the existing approximate adder of [10].

5. Proposed PAEA-CSK adder

The generalized architecture of the proposed PAEA-CSK adder shown in Figure 7 is derived from the proposed adder segment (shown in Figure 4). It receives n -bit inputs (A and B) and computes the n -bit sum S and carry

Table 1. General expressions for gate counts in m -bit CSK adder segments.

Gates	Area (in terms of gate count)		
	ACC-AS [6]	APP-AS [10]	Proposed
AND	$4m - 3$	$5m - 3 + \alpha_a$	$4m + v - 5$
OR	$m - 1$	$2m - 2 + \alpha_o$	$2m + v - 3$
NOT	--	α_i	--
XOR	$3m - 1$	$3m - 1$	$2m$
AOI	1	--	--

ACC-AS: Accurate adder segment, APP-AS: Approximate adder segment, v : number of carry-out signals (with carry-in = '0') generated from the segments (e.g., $v = 1$ indicates current segment only while $v = 2$ indicates current and previous one segment) used for carry prediction, $(\alpha_a, \alpha_o, \alpha_i)$: Number of AND, OR and NOT gates respectively in MSL and MUX for given value of v .

Table 2. Theoretical comparison of the area for CSK adder segments.

Adder Segment	Bit-width (m)	Area (transistor count)	Excess Area (%)
ACC-AS[6]	8	498	7.79
	16	1026	8.91
	32	2082	9.46
APP-AS[10]	8	602	30.30
	16	1226	30.14
	32	2474	30.07
Proposed	8	462	--
	16	942	--
	32	1902	--

ACC-AS: Accurate adder segment, APP-AS: Approximate adder segment. For area estimation $v = 2$ is considered.

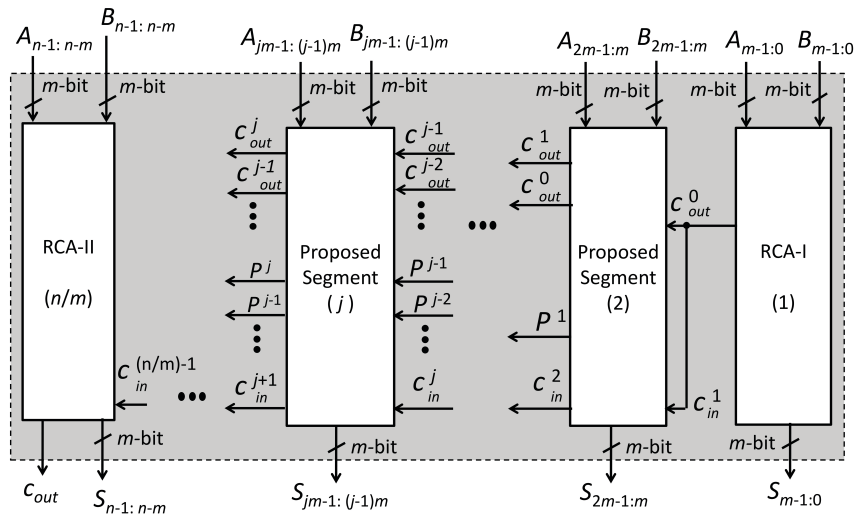


Figure 7. Generalized proposed n -bit approximate adder architecture.

(c_{out}) signal. In the proposed adder, conventional RCA subadder is used at the first and last stages to reduce the logic complexity. The RCA-I computes m -bit sum ($S_{m-1:0}$) and carry-out signal (c_{out}^0) using lower m -bits of the inputs, whereas the RCA-II computes m MSBs of sum and output carry using m MSBs of input operands and carry-in ($c_{in}^{\frac{n}{m}-1}$) signal received from the previous segment. On the other hand, an intermediate j^{th} segment receives m -bit inputs ($A_{jm-1:(j-1)m}^j$ and $B_{jm-1:(j-1)m}^j$), carry-in (c_{in}^j), carry outputs ($c_{out}^{j-1}, c_{out}^{j-2}, \dots$), and group propagate signals ($P_{out}^{j-1}, P_{out}^{j-2}, \dots$) from previous segments, and generates the sum ($S_{jm-1:(j-1)m}^j$), carry output (c_{in}^{j+1}), group propagate (P^j), and output carry (c_{out}^j) signals, where $1 \leq j \leq \frac{n}{m}$. The synthesis results and quality analysis are presented in the next section to evaluate the efficacy of the proposed design.

6. Synthesis results and quality analysis

To evaluate their effectiveness, the proposed and existing designs are implemented in VHDL and synthesized by Synopsys Design Compiler (SDC) with the 65nm TSMC CMOS Library. Further, the quality analysis is done by implementing them in MATLAB and simulating with one million random input patterns. Finally, the quality analysis of the proposed adder in a real application is performed by implementing the Sobel edge detector (SED) embedded with the proposed and existing adders. These SEDs are simulated with the benchmark Lena image and error metrics are computed [29]. The following subsections first provide the synthesis results of adder segments followed by the synthesis results of approximate adder architectures and finally present the quality analysis of these adders as a standalone unit and in the application.

6.1. Synthesis results of adder segments

We have implemented the proposed and existing adder segments for different bit widths ($m = 8, 16$, and 32 with $v = 2$) and synthesized them to achieve design metrics. The area, delay, and power reported by the SDC for various adder segments are listed in Table 3. The synthesis result of Table 3 confirms the theoretical calculations given in Table 2. The proposed adder segment on average (for the different bit widths) consumes 20.97% and 13.14% less power over the adder segment of [6] and [10], respectively. The comparison of area-delay product (ADP) for the proposed and existing adder segments is shown in Figure 8.

It can be observed from Figure 8 that the proposed adder segment on average reduces ADP by 55.3% and 46.7% over the adder segment presented in [6] and [10], respectively. From the synthesis results, it is clear that the proposed design is efficient in terms of ADP and power. Therefore, it is expected that the proposed approximate adder (shown in Figure 7) will provide significant power and ADP saving over the existing approximate adders.

6.2. Synthesis results and quality metrics analysis of the proposed PAEA-CSK

We have implemented the 16-bit proposed (using the generalized architecture shown in Figure 7) and the existing approximate CSK adder (reported in [10]) for different set of m and v values (e.g., $m = 2, v = 2$; $m = 4, v = 2$; $m = 2, v = 3$), in VHDL. We have also implemented 16-bit RAP-CLA with window size equal to 4 [12] and NzBt adder [13] for comparative analysis. The area, delay, and power as tabulated in Table 4 are obtained from synthesis using SDC. From the synthesis results, it is clear that the proposed approximate adder on average (for different set of m and v values) reduces ADP and power by 33.36% and 18.03%, respectively, over the existing similar adder design [10]. The proposed 16-bit adder (with $m = 2, v = 3$) consumes 47.03% less power

Table 3. Comparison of ASIC synthesis results for CSK adder segments.

Adder Segment	Bit width (m)	Delay (ns)	Area (μm^2)	Power (μW)
ACC-AS [6]	8	0.37	360.7	10.97
	16	0.69	661.3	22.62
	32	1.35	1310.0	46.00
APP-AS [10]	8	0.35	326.1	10.06
	16	0.68	566.2	20.31
	32	1.29	1114.9	42.09
Proposed	8	0.33	206.6	8.84
	16	0.65	288.0	17.77
	32	1.26	588.9	35.87

ACC-AS: Accurate adder segment, APP-AS: Approximate adder segment. Power is estimated with normalized clock at 200 MHz.

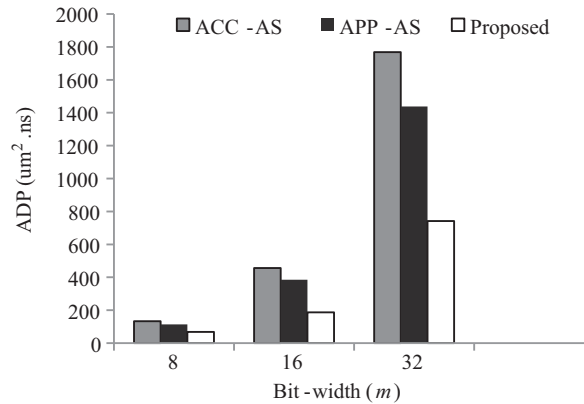


Figure 8. Comparison of area-delay products (ADPs) of adder segments for varying bit width.

and occupies 77.99% less area and also offers 79.82% less ADP over the RAP-CLA [12]. Further, the proposed 16-bit adder (with $m = 2, v = 3$) offers 60.0 % less delay and 60.48% less ADP over the NzBt adder [13].

For the assessment of the adder output quality, we have evaluated the mean error distance (MED), normalized error distance (NED), and error rate (ER) of the proposed and existing approximate adders, and they are summarized in Table 4. The quality result shows that the proposed adder, on average, for the different set of design parameters (m, v) provides 56.13%, 55.41%, and 67.43% less MED, NED, and ER, respectively, over the RAP-CLA [12]. Compared to NzBt [13], the proposed adder (with $m = 4$ and $v = 2$) involves 97.49% less NED and 96.98% less ER. Since the proposed architecture is implemented with reduced complexity without changing the functionality of [10], the quality metrics of the proposed adders are the same as those of [10]. Finally, for the quality analysis in the application, SEDs embedded with the proposed and existing adders are implemented and simulated with the Lena image. Since the quality metrics of the proposed and [10] are the same, only images processed by the SEDs with the proposed adders are shown in Figure 9. Further, the SED with the proposed adder provides a minimum of 16.94 dB higher PSNR over the SED embedded with the NzBt adder [13]. From Figure 9, it can be observed that the proposed adders provide acceptable image quality with higher power efficiency over the RAP-CLA [12].

Table 4. Comparison of ASIC synthesis results and quality metrics for 16-bit adders.

Adder	m, v	Design metrics				Quality metrics		
		Delay (ns)	Area (μm^2)	Power (μW)	ADP ($\mu m^2 \times ns$)	MED	NED	Error Rate (%)
NzBt [13]	--	0.55	451	14.53	248.0	85.13	0.334	99.99
RAP-CLA [12]	--	0.24	2024	27.48	485.8	2080	0.0308	20.99
ACSKA [10]	2, 2	0.20	520	17.63	104.0	2042	0.0307	13.69
	4, 2	0.30	596	20.60	178.9	555.21	0.0084	3.02
	2, 3	0.26	754	18.33	196.0	140.28	0.0021	3.8
Proposed	2, 2	0.19	449	15.40	85.3	2042	0.0307	13.69
	4, 2	0.28	433	16.30	121.4	555.21	0.0084	3.02
	2, 3	0.22	445	14.55	98.04	140.28	0.0021	3.8

ADP: area delay product, m : bit width of the segment, v : number of carry-out signals generated from the segments (e.g., $v = 1$ indicates current segment only while $v = 2$ indicates current and previous one segment) used for output carry prediction. Power is estimated with normalized clock at 200 MHz, MED: mean error distance, NED: Normalized error distance.

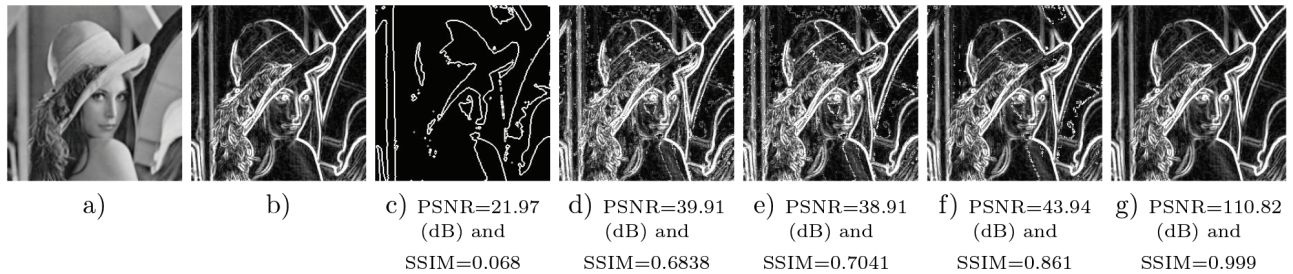


Figure 9. Lena image (256×256): a) original image, edge detected by Sobel edge detector embedded with b) accurate c) NzBt, d) RAP-CLA, and proposed adders with e) $m = 2, v = 2$ f) $m = 2, v = 3$, and g) $m = 4, v = 2$.

7. Conclusion

In this paper, various logic operations performed in the existing adder architectures are analyzed and redundant operations are identified. Further, a novel approximate adder segment architecture is presented by eliminating redundant logic and sharing common logics. The synthesis results show that the proposed adder segment provides on average 13.14% less power and 46.7% less ADP than the ACSKA segment. Using the proposed adder segment, a new approximate adder is proposed that reduces implementation complexity and power over the existing approximate adders. The 16-bit proposed approximate adder was found better in terms of power consumption and ADP than the best available approximate adder. Finally, the proposed approximate adder on average (for different set of m and v values) reduces 33.36% and 18.03% ADP and power, respectively, over the existing ACSKA.

References

- [1] Mittal S. A survey of techniques for approximate computing. ACM Computing Surveys (CSUR) 2016; 48 (4): 1-33.
- [2] Celia D, Vasudevan V, Chandrachoodan N. Optimizing power-accuracy trade-off in approximate adders. In: IEEE 2018 Design, Automation & Test in Europe Conference & Exhibition (DATE); Dresden, Germany; 2018. pp. 1488-1491.

- [3] Jiang H, Han J, Lombardi F. A comparative review and evaluation of approximate adders. In: ACM Proceedings of the 25th edition on Great Lakes Symposium VLSI; Pittsburgh, PA, USA; 2015. pp. 343-348.
- [4] Ramkumar B, Kittur HM. Low-power and area-efficient carry select adder. *IEEE Transactions on Very Large Scale Integration Systems* 2012; 20 (2): 371-375.
- [5] Mohanty BK, Patel SK. Area-delay-power efficient carry-select adder. *IEEE Transactions on Circuits and Systems II: Express Briefs* 2014; 61 (6): 418-422.
- [6] Bahadori M, Kamal M, Afzali-Kusha A, Pedram M. High-speed and energy-efficient carry skip adder operating under a wide range of supply voltage levels. *IEEE Transactions on VLSI Systems* 2016; 24 (2): 421-433.
- [7] Gupta V, Mohapatra D, Raghunathan A, Roy K. Low-power digital signal processing using approximate adders. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 2013; 32 (1): 124-137.
- [8] Zhu N, Goh WL, Zhang W, Yeo KS, Kong ZH. Design of low-power high-speed truncation-error-tolerant adder and its application in digital signal processing. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 2010; 18 (8): 1225-1229.
- [9] Mahdiani HR, Ahmadi A, Fakhraie SM, Lucas C. Bio-inspired imprecise computational blocks for efficient VLSI implementation of soft-computing applications. *IEEE Transactions on Circuits and Systems I: Regular Papers* 2010; 57 (4): 850-862.
- [10] Kim Y, Zhang Y, Li P. Energy efficient approximate arithmetic for error resilient neuromorphic computing. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 2015; 23 (11): 2733-2737.
- [11] Kahng A, Kang S. Accuracy-configurable adder for approximate arithmetic designs. In: 49th ACM/EDAC/IEEE Design Automation Conference (DAC); San Francisco, CA, USA; 2012. pp. 820-825.
- [12] Akbari O, Kamal M, Afzali-Kusha A, Pedram M. RAP-CLA: a reconfigurable approximate carry look-ahead adder. *IEEE Transactions on Circuits and Systems II: Express Briefs* 2018; 65 (8): 1089-1093.
- [13] Frustaci F, Perri S, Corsonello P, Alioto M. Energy-quality scalable adders based on non-zeroing bit truncation. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 2019; 27 (4): 964-968.
- [14] Yang Z, Jain A, Liang J, Han J, Lombardi F. Approximate XOR/XNOR-based adders for inexact computing. In: 13th IEEE Conference on Nanotechnology; Beijing, China; 2013. pp. 690-693.
- [15] Dalloo A, Najafi A, Garcia-Ortiz A. Systematic design of an approximate adder: the optimized lower part constant-or adder. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 2018; 26 (8): 1595-1599.
- [16] Pashaeifar M, Kamal M, Afzali-Kusha A, Pedram A. Approximate reverse carry propagate adder for energy-efficient DSP applications. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 2018; 99 (11): 1-12.
- [17] Garg B, Sharma GK. A process-tolerant low-power adder architecture for image processing applications. *Turkish Journal of Electrical Engineering & Computer Sciences* 2019; 27 (3): 1839-1854.
- [18] Zhu N, Goh WL, Yeo KS. Ultra low-power high-speed flexible probabilistic adder for error-tolerant applications. In: IEEE 2011 International SoC Design Conference (ISOCC); Jeju, South Korea; 2011. pp. 393-396.
- [19] Kim Y, Zhang Y, Li P. An energy efficient approximate adder with carry skip for error resilient neuromorphic VLSI systems. In: ACM 2013 Proceedings of the International Conference on Computer-Aided Design (ICCAD); San Jose, CA, USA; 2013. pp. 130-137.
- [20] Dutt S, Dash S, Nandi S, Trivedi G. Analysis, modeling and optimization of equal segment based approximate adders. *IEEE Transactions on Computers* 2018; 68 (3): 314-330.
- [21] Celia D, Vasudevan V, Chandrachoodan N. Probabilistic error modeling for two-part segmented approximate adders. In: IEEE 2018 International Symposium on Circuits and Systems (ISCAS); Florence, Italy; 2018. pp. 1-5.
- [22] Shafique M, Ahmad W, Hafiz R, Henkel J. A low latency generic accuracy configurable adder. In: 52nd ACM/EDAC/IEEE Design Automation Conference (DAC); San Francisco, CA, USA; 2015. pp. 1-6.

- [23] Esposito D, De-Caro D, Strollo AGM. Variable latency speculative parallel prefix adders for unsigned and signed operands. *IEEE Transactions on Circuits and Systems I: Regular Papers* 2016; 63 (8): 1200-1209.
- [24] Yang T, Ukezono T, Sato T. A low-power configurable adder for approximate applications. In: *IEEE 19th International Symposium on Quality Electronic Design (ISQED)*; Santa Clara, CA, USA; 2018. pp. 347-352.
- [25] Garg B, Dutt S, Sharma G. Bit-width-aware constant-delay run-time accuracy programmable adder for error-resilient applications. *Microelectronics Journal* 2016; 50: 1-7.
- [26] Katreepalli R, Haniotakis T. High speed power efficient carry select adder design. In: *IEEE 2017 Computer Society Annual Symposium on VLSI (ISVLSI)*; Bochum, Germany; 2017. pp. 32-37.
- [27] Xu W, Sapatnekar SS, Hu J. A simple yet efficient accuracy-configurable adder design. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 2018; 26 (6): 1112-1125.
- [28] “dbtcbn65gplusbc0d88 TSMC 65nm CMOS library databook.”
- [29] Liang J, Han J, Lombardi F. New metrics for the reliability of approximate and probabilistic adders. *IEEE Transactions on Computers* 2012; 62 (9): 1760-1771.