

Communication

# Performance Comparison of Carry-Lookahead and Carry-Select Adders Based on Accurate and Approximate Additions

Padmanabhan Balasubramanian <sup>1,\*</sup>  and Nikos Mastorakis <sup>2</sup>

<sup>1</sup> School of Computer Science and Engineering, Nanyang Technological University, 50 Nanyang Avenue, Singapore 639798, Singapore

<sup>2</sup> Department of Industrial Engineering, Technical University of Sofia, boulevard Sveti Kliment Ohridski 8, Sofia 1000, Bulgaria; mastor@tu-sofia.bg

\* Correspondence: balasubramanian@ntu.edu.sg; Tel.: +65-6790-4745

Received: 12 November 2018; Accepted: 29 November 2018; Published: 2 December 2018



**Abstract:** Addition is a fundamental operation in microprocessing and digital signal processing hardware, which is physically realized using an adder. The carry-lookahead adder (CLA) and the carry-select adder (CSLA) are two popular high-speed, low-power adder architectures. The speed performance of a CLA architecture can be improved by adopting a hybrid CLA architecture which employs a small-size ripple-carry adder (RCA) to replace a sub-CLA in the least significant bit positions. On the other hand, the power dissipation of a CSLA employing full adders and 2:1 multiplexers can be reduced by utilizing binary-to-excess-1 code (BEC) converters. In the literature, the designs of many CLAs and CSLAs were described separately. It would be useful to have a direct comparison of their performances based on the design metrics. Hence, we implemented homogeneous and hybrid CLAs, and CSLAs with and without the BEC converters by considering 32-bit accurate and approximate additions to facilitate a comparison. For the gate-level implementations, we considered a 32/28 nm complementary metal-oxide-semiconductor (CMOS) process targeting a typical-case process–voltage–temperature (PVT) specification. The results show that the hybrid CLA/RCA architecture is preferable among the CLA and CSLA architectures from the speed and power perspectives to perform accurate and approximate additions.

**Keywords:** arithmetic circuits; ripple-carry adder; carry-lookahead adder; carry-select adder; digital design; standard cells; CMOS

## 1. Introduction

Addition is pervasive in microprocessing and digital signal processing hardware, and addition is performed using an adder. For practical applications, the adder should feature high speed and low power. In this context, the carry-lookahead and carry-select adders are two popular high-speed, low-power adder architectures [1]. Two variants of the carry-lookahead adder (CLA) are common, namely the recursive CLA (RCLA) [2] and the block CLA (BCLA) [3]. The speed performance of these CLA architectures can be improved by adopting a hybrid CLA architecture involving a small-size ripple-carry adder (RCA) in the least significant adder bit positions as a replacement for one or more sub-CLAs [4]. Moreover, the improvement in the speed performance can be gained along with a reduction in the power dissipation in the case of the hybrid CLAs. Hence, the RCLA/RCA and the BCLA/RCA are also referred to as high-speed, low-power hybrid CLA architectures [5].

Two variants of the carry-select adder (CSLA) [6] are common—one architecture involving full adders and 2:1 multiplexers (MUXes), and the other architecture involving full adders, 2:1 MUXes, and binary-to-excess-1 code (BEC) converters [7,8]. In the literature, the designs of homogeneous and hybrid CLAs at the gate level and transistor level are discussed. Similarly, the designs of CSLAs with and without BEC converters are described. However, the designs of various CLAs and CSLAs were considered separately. A direct comparison of the performances of CLAs and CSLAs based on different addition bit-widths was recently performed in Reference [9], but the CLA synthesized was not delay-optimal; rather, it was optimized for area. In fact, the CLA designs reported in Reference [9] consistently exhibit more critical path delays than even the simple RCA for different addition bit-widths. Here, our focus is on realizing high-speed, low-power designs of CLAs and CSLAs at the gate level, followed by a comparison of their performances in terms of the design metrics. Such a comparison could be useful for determining which of these two architectures would be potentially a better choice for implementing high-speed and low-power addition, and whether a homogeneous or a hybrid version of that adder architecture would be preferable. Also, a high-speed and low-power adder that is custom synthesized could be included as a user-defined library component, which can subsequently be utilized during the automated synthesis of high-speed and low-power computer arithmetic within an application-specific integrated circuit (ASIC) design environment.

This communication discusses the physical realization of homogeneous and hybrid CLAs and CSLAs with and without BEC converters by considering example 32-bit additions, which correspond to accurate and approximate computation. To perform approximate addition, the lower-part-OR approximate adder (LOA) of Reference [10] is considered. The diverse CLAs and CSLAs were physically implemented using a 32/28 nm bulk complementary metal-oxide-semiconductor (CMOS) process [11], targeting a typical-case process–voltage–temperature (PVT) specification with a supply voltage of 1.05 V and an operating temperature of 25 °C to perform the simulations.

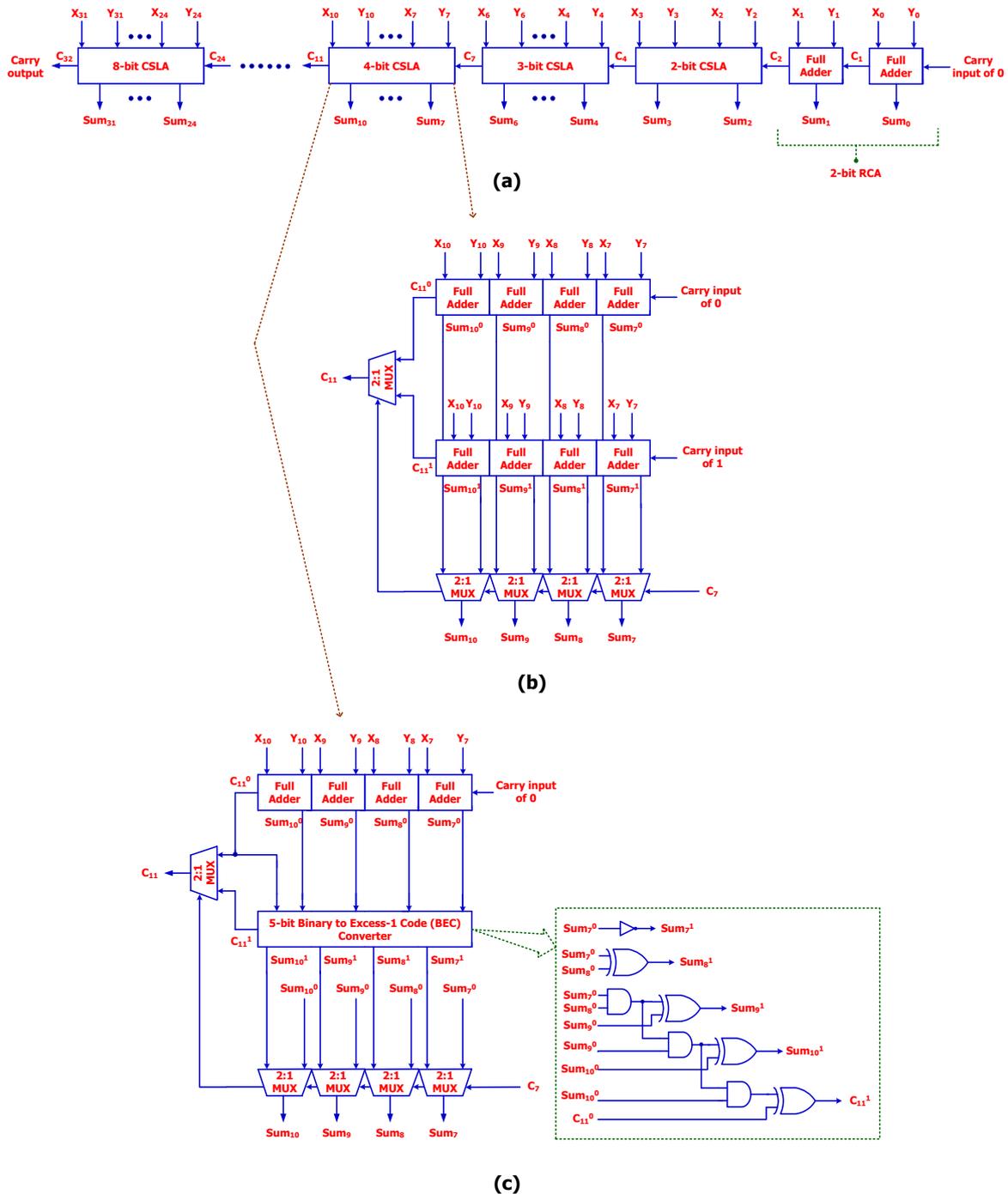
The rest of this communication is organized into four sections. Section 2 discusses the CSLA architectures. Section 3 discusses the homogeneous and hybrid CLA architectures. Section 4 presents the simulation results obtained for the CLAs and CSLAs corresponding to accurate and approximate additions. This is followed by the conclusions given in Section 5.

## 2. CSLA Architectures

The CSLA architecture involves partitioning the augend and addend input bits into equally or unequally sized groups, and segmenting the entire addition into many sub-additions, which can be performed in parallel. If the augend and addend input bits of a CSLA are partitioned into equally sized groups, it is called “uniform CSLA”, and, if the augend and addend input bits of a CSLA are partitioned into unequally sized groups, it is called “non-uniform CSLA”. It was noted in References [7,8] that a non-uniform CSLA is preferable for achieving high speed and low power.

Two types of CSLA architectures are common, the first involving an RCA in the least significant adder bit positions and using dual RCAs of appropriate size as dictated by the input partitions, with one set of RCAs having a fixed carry input of 0 and another set having a fixed carry input of 1. The outputs of the dual RCAs are given to 2:1 MUXes with the carry output of the preceding input partition serving as the select input for the MUXes belonging to the current input partition—this architecture shall be referred to as the “CSLA\_NOBEC”, implying no use of BEC converters. The second architecture uses an RCA for the least significant adder bit positions, using as many RCAs of appropriate size as dictated by the input partitions with a fixed carry input of 0 assigned. The outputs of these RCAs are given to BEC converters [7], which increment the outputs of the RCAs by binary 1—this architecture shall be referred to as the “CSLA\_BEC”, signifying the use of BEC converters. The selection of either the outputs of the RCAs having a fixed carry input of 0 or the outputs of the BEC converters to produce the required sum in a CSLA\_BEC is performed using the MUXes. For the MUXes belonging to an input partition, the carry outputs from a preceding input partition serves as their common selection input.

The block schematic of an optimum 32-bit non-uniform CSLA is portrayed in Figure 1a. The gate-level realizations of example 4-bit CSLAs with one involving full adders and 2:1 MUXes (CSLA\_NOBEC), and the other involving full adders, 2:1 MUXes, and BEC converters (CSLA\_BEC) are shown in Figure 1b,c respectively. The internal gate-level detail of an example 5-bit BEC converter is also shown within the green dotted rectangle in Figure 1c.



**Figure 1.** (a) Non-uniform 32-bit carry-select adder (CSLA) based on an optimum 8-7-6-4-3-2 input partition; (b) internal detail of a 4-bit (sub-)CSLA comprising full adders and 2:1 multiplexers (MUXes); (c) internal detail of a 4-bit (sub-)CSLA comprising full adders, 2:1 MUXes, and a 5-bit binary-to-excess-1 code (BEC) converter.

### 3. Homogeneous CLA and Hybrid CLA/RCA Architectures

#### 3.1. RCLA and RCLA/RCA Architectures

The RCLA architecture is based on a physical realization of the recursive carry-lookahead equations [1–3]. The generalized logic expressions of the propagate and generate functions, the lookahead carry output, and the sum output are given by Equations (1) to (4). In Equations (1) to (4),  $I$  represents an arbitrary adder bit position,  $X$  and  $Y$  represent the adder input bits,  $P$  and  $G$  represent the propagate and generate functions,  $C$  represents the carry signal, and  $SUM$  represents the sum output. The propagate function is derived by performing an exclusive-OR (XOR) of the corresponding augend and addend input bits. The generate function is derived from a logical conjunction of the corresponding augend and addend input bits. The lookahead carry output ( $C_{I+1}$ ) corresponding to a sub-CLA is derived based on Equation (3), involving the propagate and generate functions and the carry input ( $C_I$ ). The sum output bit is produced by an XOR of the corresponding propagate function and the carry input bit.

$$P_I = X_I \oplus Y_I \quad (1)$$

$$G_I = X_I Y_I \quad (2)$$

$$C_{I+1} = G_I + P_I G_{I-1} + \dots + P_I P_{I-1} \dots P_0 C_I \quad (3)$$

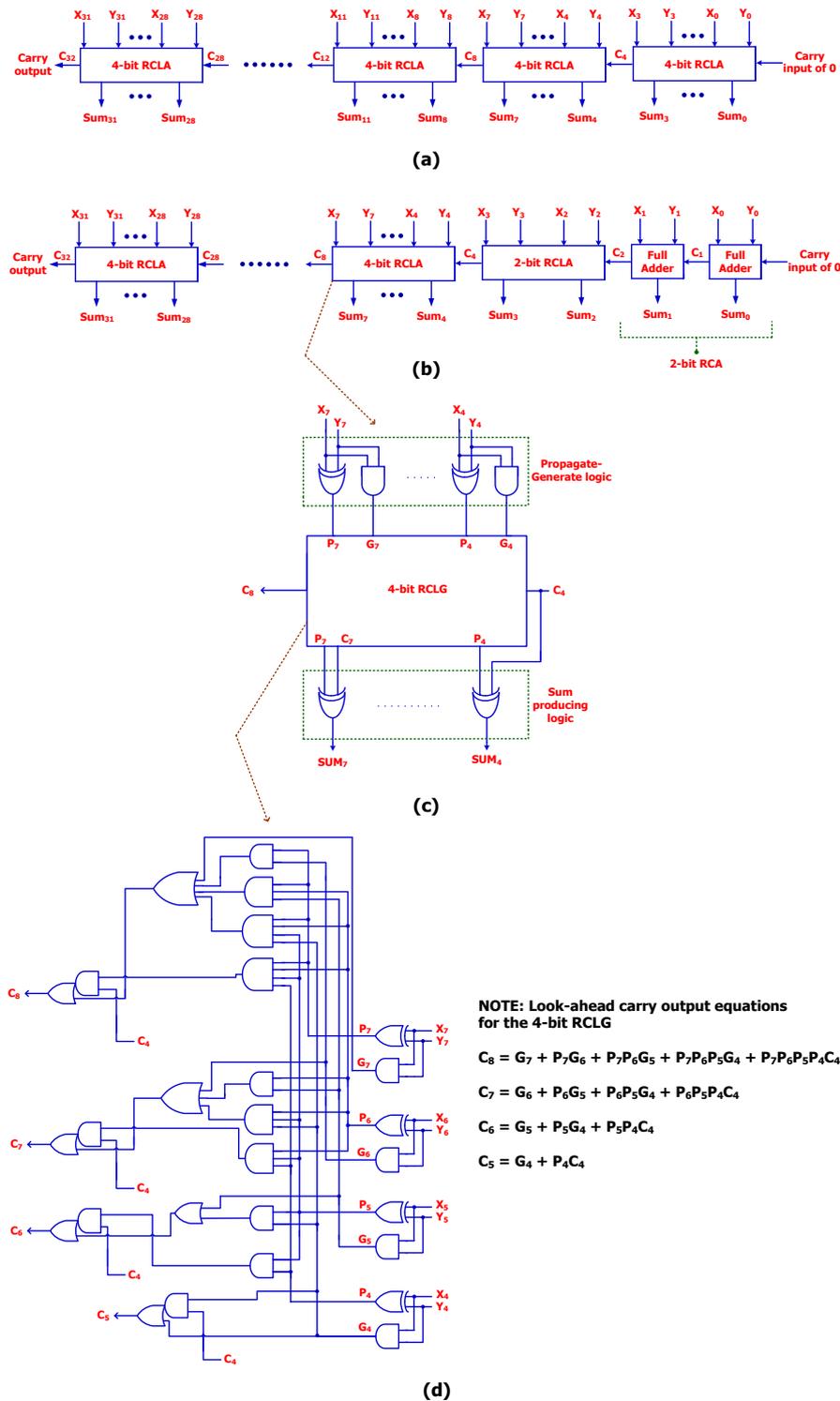
$$SUM_I = P_I \oplus C_I \quad (4)$$

Equation (3) is inherently recursive in nature since the lookahead carry output corresponding to any bit position can be derived based on a knowledge of the given carry input. An RCLA basically consists of the propagate–generate logic which encompasses Equations (1) and (2), the recursive carry-lookahead generator (RCLG) which encompasses Equation (3), and the sum-producing logic which encompasses Equation (4). An RCLA is usually constructed by cascading many small-sized (sub-)RCLAs. For example, a 32-bit RCLA may be realized by cascading eight 4-bit (sub-)RCLAs. Different realizations of the RCLA are possible, as discussed in References [4,5], and the physical realization of a high-speed and low-power RCLA is of interest [5].

A homogeneous 32-bit RCLA, comprising eight delay-optimized 4-bit RCLAs [5], is shown in Figure 2a. An optimum hybrid 32-bit RCLA incorporating a 2-bit RCLA and two full adders in the least significant bit positions is shown in Figure 2b. The internal details of an example 4-bit RCLA is depicted in Figure 2c, which consists of the propagate–generate logic, a 4-bit RCLG, and the sum-producing logic. The gate-level realization of a delay-optimized 4-bit RCLG is portrayed in Figure 2d [5]. It may be seen from Figure 2d that, based on the carry input  $C_4$ , and according to Equation (3), four lookahead carry output signals are generated, namely  $C_5$ ,  $C_6$ ,  $C_7$ , and  $C_8$ . Of these,  $C_4$ ,  $C_5$ ,  $C_6$ , and  $C_7$  are subjected to XOR with the corresponding propagate functions, namely  $P_4$ ,  $P_5$ ,  $P_6$ , and  $P_7$ , to produce the respective sum output bits, i.e.,  $SUM_4$  to  $SUM_7$ .  $C_8$  is the lone lookahead carry output signal that is passed on to the successive 4-bit sub-RCLA to serve as its carry input. In an  $M$ -bit (sub-)RCLA,  $M$  propagate and generate functions will be generated,  $M$  lookahead carry signals will be produced, and  $(M-1)$  lookahead carry signals will be utilized internally in the  $M$ -bit (sub-)RCLA to produce the  $M$  sum output bits. Only the most significant lookahead carry signal will be passed on to the successive sub-RCLA to serve as its carry input.

It may be noticed from Figure 2d that the maximum data path delay (also called the critical path delay) is encountered in producing  $C_8$ , which is given by the sum of the propagation delays of a two-input XOR gate, a four-input AND gate, a four-input OR gate, and the final AO21 complex gate. The least significant 4-bit (sub-)RCLA present in an  $N$ -bit RCLA would encounter this critical path delay. However, the subsequent 4-bit (sub-)RCLAs would encounter the least possible data path delay, which is the propagation delay of just one AO21 complex gate. Hence, it may be beneficial to replace the least significant  $M$ -bit (sub-)RCLA in an  $N$ -bit RCLA with a reduced (sub-)RCLA and any full adders. Given this, Figure 2b shows the replacement of a 4-bit (sub-)RCLA by a 2-bit (sub-)RCLA and two full

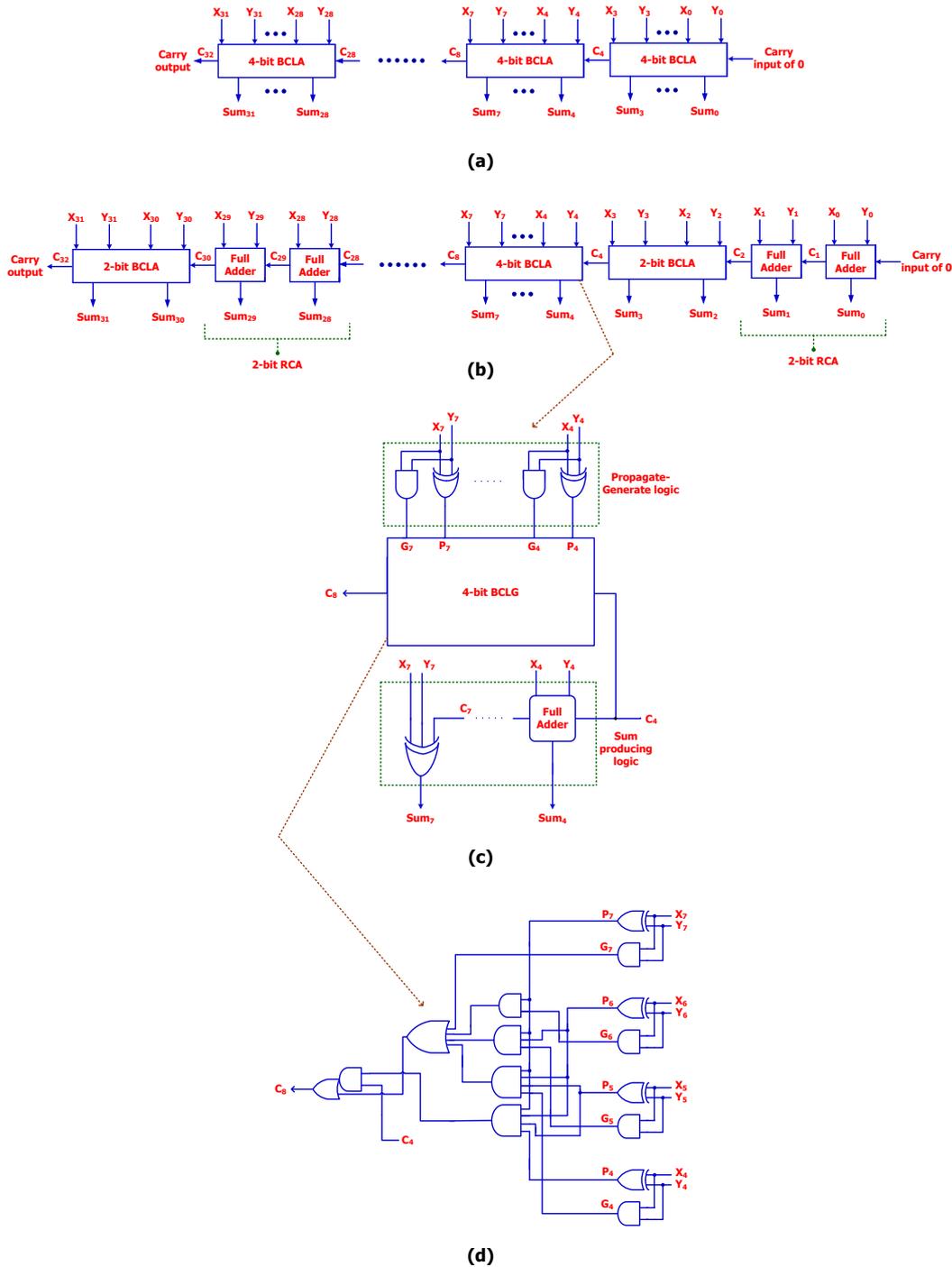
adders, thereby giving rise to a hybrid RCLA/RCA architecture. The hybrid RCLA/RCA architecture would help reduce the critical path delay, the silicon area, and the average power dissipation of the homogeneous RCLA [5].



**Figure 2.** (a) A 32-bit homogeneous recursive carry-lookahead adder (RCLA); (b) an optimum 32-bit hybrid RCLA/ripple-carry adder (RCA); (c) internal detail of a 4-bit (sub-)RCLA; (d) gate-level detail of a delay-optimized 4-bit recursive carry-lookahead generator (RCLG) which is custom-realized, and the corresponding look-ahead carry output equations.

3.2. BCLA and BCLA/RCA Architectures

The BCLA [3], also called the section-carry-based carry-lookahead adder (SCBCLA) [4,5], is another type of CLA, which also utilizes the recursive carry-lookahead equation for synthesis. Just like the RCLA, an  $N$ -bit BCLA is constructed using many small  $M$ -bit (sub-)BCLAs. Figure 3a shows a homogeneous 32-bit BCLA constructed using eight delay-optimized 4-bit (sub-)BCLAs, and Figure 3b shows a hybrid 32-bit BCLA/RCA.



**Figure 3.** (a) Homogeneous 32-bit block carry-lookahead adder (BCLA); (b) optimum hybrid 32-bit BCLA/RCA; (c) internal detail of a 4-bit (sub-)BCLA; (d) gate-level detail of a delay-optimized 4-bit block carry-lookahead generator (BCLG), which is custom-realized.

However, the BCLA is different from the RCLA. An  $M$ -bit BCLA (also called the sub-BCLA) receives a lookahead carry input from a preceding (sub-)BCLA and produces one lookahead carry output for the successive (sub-)BCLA. Recall that, in contrast, an  $M$ -bit RCLA produces  $M$  lookahead carry outputs. Figure 3c shows an example  $M$ -bit (sub-)BCLA, assuming  $M = 4$ .

An  $M$ -bit BCLA consists of the propagate–generate logic, an  $M$ -bit block carry-lookahead generator (BCLG), and the sum-producing logic, as shown in Figure 3c. The carry input received by the  $M$ -bit (sub-)BCLA is used to produce one lookahead carry output. The carry input is simultaneously processed by a cascade of  $(M-3)$  full adders and one three-input XOR gate, which resembles a sub-RCA, to produce the corresponding sum output bits. The gate-level detail of a delay-optimized 4-bit BCLG is shown in Figure 3d. Basically, the logic corresponding to  $C_8$  is extracted from Figure 2d, and the rest of the lookahead carry output logic is discarded, resulting in Figure 3d. The logic expression for  $C_8$ , corresponding to Figure 3d, is the same as that given in Figure 2d.

In a homogeneous  $N$ -bit BCLA, constructed by cascading  $N/4$  4-bit (sub-)BCLAs where  $N$  modulo 4 is equal to 0, the critical path traversed in the least significant 4-bit (sub-)BCLA would correspond to the lookahead carry output, which would comprise a two-input XOR gate, a four-input AND gate, a four-input OR gate, and the final AO21 complex gate. To reduce the critical path delay, one option would be to replace the least significant 4-bit (sub-)BCLA with a smaller size (sub-)BCLA and any full adders. Figure 2b shows the resultant optimum hybrid BCLA/RCA architecture for 32-bit addition. It may be seen from Figure 2b that the most significant 4-bit (sub-)BCLA is also replaced with a 2-bit (sub-)BCLA and two full adders just like what was done for the least significant 4-bit (sub-)BCLA. This is because the critical path of the most significant 4-bit (sub-)BCLA would have three full adders and one three-input XOR gate. On the contrary, in Figure 2b, the critical path would encounter one AO21 gate and two full adders, which helps slightly reduce the critical path delay. Just like the hybrid RCLA/RCA architecture, the hybrid BCLA/RCA architecture would help reduce the critical path delay, the silicon area, and the average power dissipation of the homogeneous BCLA architecture.

#### 4. Results and Discussion

A semi-custom ASIC-style standard cell-based physical implementation of homogeneous and hybrid 32-bit CLAs and CSLAs was considered to compare their performances in terms of the design metrics. All the adders were implemented using a high- $V_t$  32/28 nm CMOS process technology [11]. The full adder and 2:1 MUX present in the digital cell library [11] were utilized while realizing the CLAs and CSLAs. The critical path delay, silicon area, and average power dissipation of the adders were estimated, and the simulation environment corresponds to a typical-case PVT specification of the standard digital cell library with a recommended supply voltage of 1.05 V and an operating junction temperature of 25 °C. To estimate the average power dissipation, about 1000 random input vectors were identically supplied to all the adders at time intervals of 5 ns (200 MHz) using the same test bench. The switching activity captured during the functional simulations was subsequently utilized to estimate the average power dissipation. The critical path delays and area occupancies were also estimated. Default wire load, i.e., the maximum wire load selection group “predcaps”, was automatically included while estimating the design metrics. Synopsys electronic design automation (EDA) tools, namely Design Vision and VCS, were used to implement and simulate the designs, and PrimeTime was used to estimate the design metrics. The time-based power analysis mode of PrimeTime was invoked to accurately estimate the average power dissipation.

The power–delay product (PDP) and the energy–delay product (EDP) are well-known parameters to quantify the low-power design efficiency of a digital circuit or system [12]. Given this, the PDP and the EDP of the adders were calculated and normalized. The normalization was performed in this manner; among the calculated PDP and EDP values, the highest values of PDP and EDP were considered as the baseline values, and these values were respectively used to divide the actual PDP and EDP values of all the adders. Thus, the least fractional value of the PDP and EDP parameters

corresponding to an adder would imply that it is the best among the lot. Since power, delay, and energy are desirable to be minimized, the lowest values of PDP and EDP would indicate the best design.

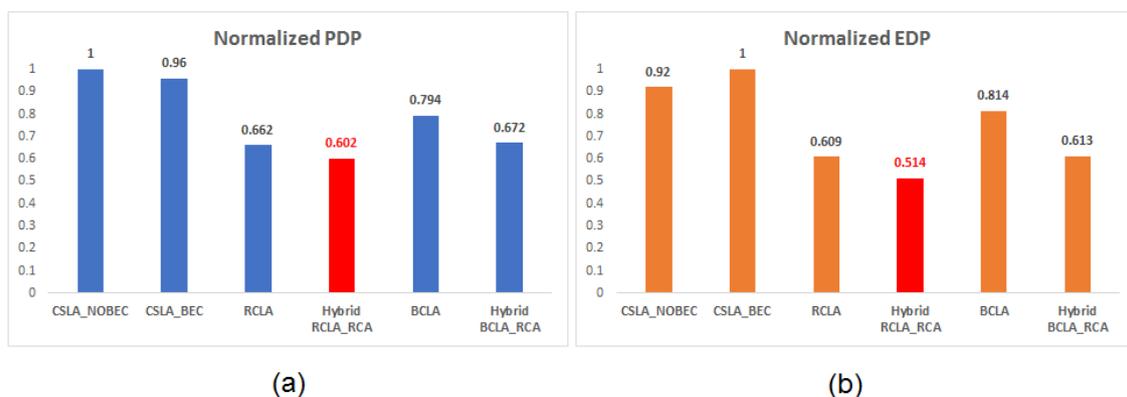
#### 4.1. Results for Accurate Addition

Accurate 32-bit CSLAs pertaining to CSLA\_NOBEC and CSLA\_BEC architectures discussed in Section 2 were physically realized using the gates of the 32/28 nm standard digital cell library [11]. Accurate 32-bit homogeneous RCLA and BCLA, and hybrid RCLA/RCA and BCLA/RCA, which are discussed in Section 3, were also physically realized using the same digital cell library. The design metrics estimated for the accurate 32-bit adders are given in Table 1. The normalized PDP and EDP plots of the adders are shown in Figure 4a,b, respectively. The red bars in Figure 4 highlight the best among the CLAs and CSLAs considered for physical implementation, which corresponds to the hybrid RCLA/RCA architecture.

From Table 1, in terms of delay and power, the hybrid RCLA/RCA architecture is preferable to the rest, and this enables it to have the lowest values of PDP and EDP compared to the other CLAs and CSLAs. The hybrid 32-bit RCLA/RCA reports a 9.1% reduction in the PDP and a 15.6% reduction in the EDP compared to its closest counterpart, namely the homogeneous 32-bit RCLA. Moreover, the former requires 6% less silicon than the latter. The 32-bit BCLA was found to occupy less area than the 32-bit RCLA. This is because the 4-bit BCLA shown in Figure 3c requires 22.6% less silicon than the 4-bit RCLA shown in Figure 2c for the physical realization. In terms of the area occupancy, CSLA\_NOBEC was found to be the best among the CLAs and CSLAs considered. Nevertheless, the PDP and EDP of the hybrid RCLA/RCA are significantly lower than the corresponding parameters of the CSLA\_NOBEC by 39.8% and 44.1%, respectively.

**Table 1.** Design metrics corresponding to accurate 32-bit addition. CSLA\_NOBEC—carry-select adder with no binary-to-excess-1 code converter; CSLA\_BEC— carry-select adder with binary-to-excess-1 code converter; RCLA—recursive carry-lookahead adder; RCA—ripple-carry adder; BCLA—block carry-lookahead adder.

Type of Adder	Delay (ns)	Area ( $\mu\text{m}^2$ )	Power ( $\mu\text{W}$ )
CSLA_NOBEC	1.13	418.32	61.51
CSLA_BEC	1.28	459.49	52.12
RCLA	1.13	646.54	40.70
Hybrid RCLA/RCA	1.05	607.91	39.82
BCLA	1.26	500.16	43.80
Hybrid BCLA/RCA	1.12	457.97	41.72



**Figure 4.** Normalized (a) power–delay product (PDP) plots, and (b) energy–delay product (EDP) plots of accurate 32-bit adders.

#### 4.2. Results for Approximate Addition

For performing approximate addition, the LOA presented in [10] was utilized, as its efficacy was verified based on neural network and fuzzy applications. Moreover, the LOA was found to offer the optimum cost–error trade-off in the stochastic regime [13]. The LOA basically bi-partitions the input bits and gives them over for processing to the most significant accurate adder part and the least significant approximate adder part. Here, for example, we considered an equal bi-partition of the input bits to realize the LOA i.e., 16 bits were allotted to the accurate adder part, and 16 bits were allotted to the approximate adder part. The approximate adder part consists of a series of two-input OR gates, each of which performs a logical disjunction of the corresponding augend and addend input bits. The most significant bit pair of the approximate adder part is subjected to AND and its output is given as the carry input for the accurate adder part. The accurate adder part may be realized using any high-speed adder. In this communication, the accurate adder part was realized using the CLA and CSLA architectures discussed in the previous sections. The resulting LOA structures are shown in Figure 5a–e.

Figure 5a shows the LOA which consists of a 16-bit non-uniform CSLA for the accurate adder part. An optimum 5-4-3-2-2 input partition [8] was considered to realize the non-uniform CSLA for the accurate adder part. The 16-bit CSLA can incorporate either the CSLA\_NOBEC architecture or the CSLA\_BEC architecture. Figure 5b shows the homogeneous RCLA used for the accurate adder part which comprises four 4-bit (sub-)RCLAs. Figure 5c shows the use of the hybrid RCLA/RCA for the accurate adder part with a 2-bit (sub-)RCLA and two full adders used in the least significant nibble position of the accurate adder part, and three 4-bit (sub-)RCLAs used for the more significant bit positions.

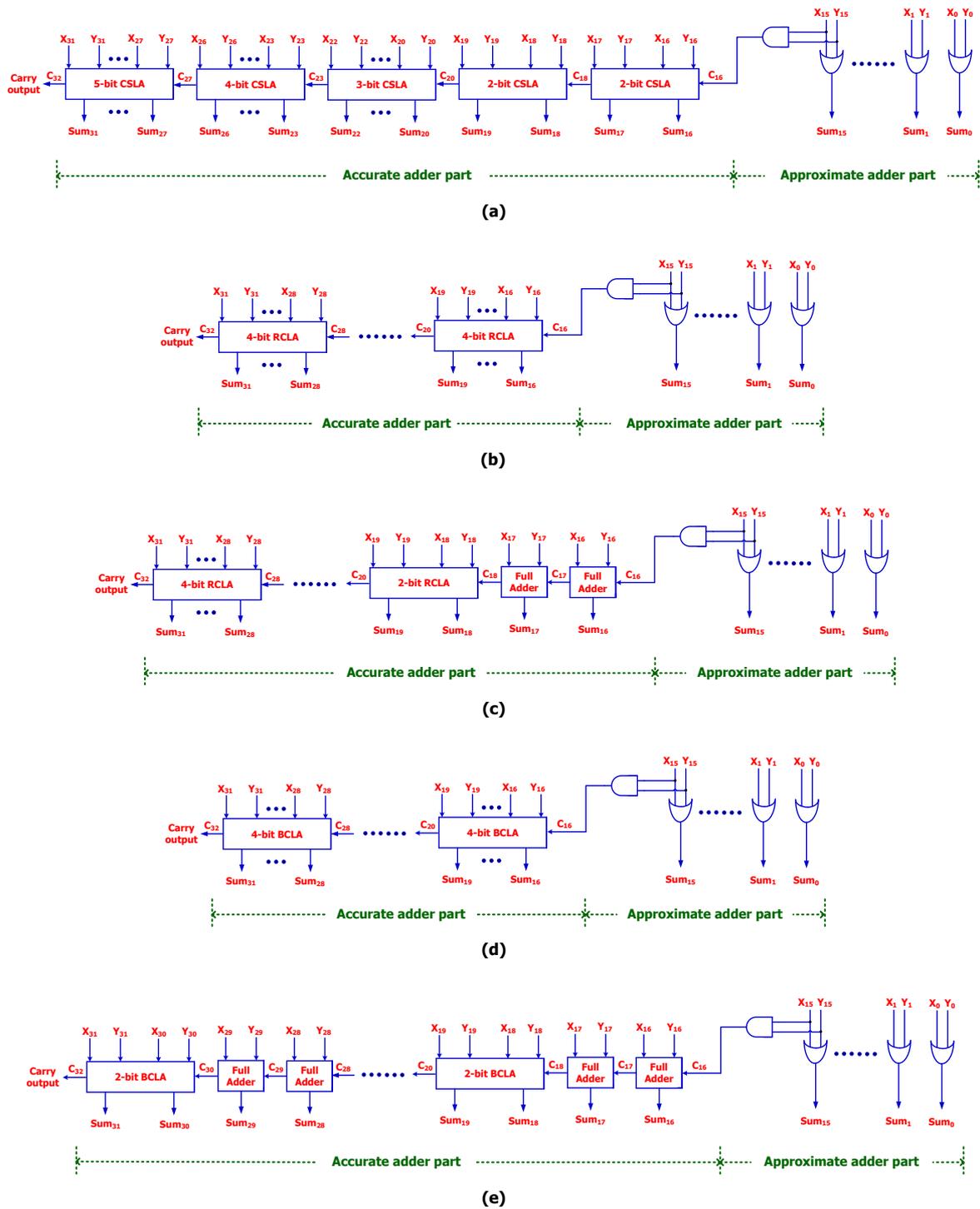
Figure 5d shows the homogeneous BCLA used to realize the 16-bit accurate adder part which was composed of four 4-bit (sub-)BCLAs, and Figure 5e shows the hybrid BCLA/RCA used to realize the accurate adder part. In the case of Figure 5e, the combination of a 2-bit (sub-) BCLA and two full adders was used for the most significant and least significant nibble positions of the accurate adder part, and 4-bit (sub-)BCLAs were used for the two intermediate nibble positions.

The design parameters such as critical path delay, silicon area, and average power dissipation estimated for the approximate 32-bit adders (LOAs) are given in Table 2. The PDP and EDP values were also calculated for the LOAs and normalized according to the same procedure discussed earlier. The normalized PDP and EDP plots are shown in Figure 6a,b respectively.

From Table 2, with respect to delay and power, it is seen that the LOA having the hybrid RCLA/RCA in the accurate adder part achieved better optimizations than the rest. Thus, the LOA incorporating the hybrid RCLA/RCA in the accurate adder part reported the lowest values of PDP and EDP compared to the LOAs which employed the other CLAs or CSLAs for the accurate adder part. The 32-bit LOA featuring the hybrid RCLA/RCA in the accurate adder part reported 11.3% and 17.1% reductions in the PDP and EDP, respectively, compared to its closest counterpart, namely the 32-bit LOA featuring a homogeneous RCLA for the accurate adder part. In addition, the former had a 10.8% lower silicon footprint than the latter.

In terms of the silicon area, the 32-bit LOA utilizing the hybrid BCLA/RCA for the accurate adder part was the best among the lot occupying 6.2% less area than its closest counterpart, namely the 32-bit LOA featuring the CSLA\_NOBEC architecture for the accurate adder part. This is mainly because only two 4-bit (sub-)BCLAs were used in the case of the former for the two intermediate nibble positions of the accurate adder part, and two 2-bit (sub-)BCLAs and four full adders were used for the remaining bit positions. Compared to a 4-bit (sub-)BCLA, the combination of a 2-bit (sub-)BCLA and two full adders consumes 33.7% less area. This translates into a lower area requirement for the 32-bit LOA comprising the 16-bit BCLA/RCA for the accurate adder part compared to the 32-bit LOA which consists of the 16-bit CSLA\_NOBEC for the accurate adder part. However, the LOA featuring a 16-bit RCLA/RCA for the accurate adder part achieved considerable reductions in the PDP and EDP values

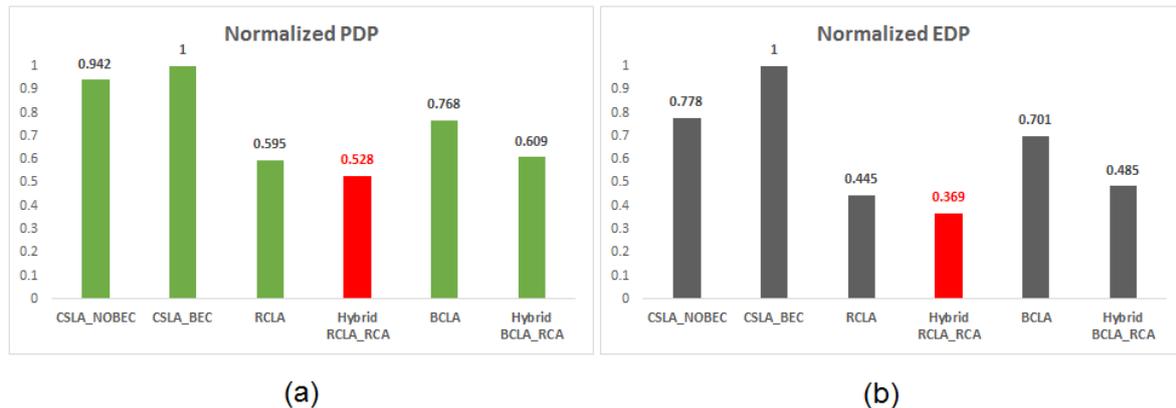
by 13.3% and 23.9%, compared to the corresponding design parameters of the LOA featuring a 16-bit BCLA/RCA for the accurate adder part.



**Figure 5.** Different 32-bit lower-part-OR approximate adders (LOAs) consisting of diverse 16-bit accurate adder parts and a common 16-bit approximate adder part.

**Table 2.** Design metrics corresponding to approximate 32-bit addition. LOA—lower-part-OR approximate adder.

Type of Accurate Adder Part Used in the LOA	Delay (ns)	Area ( $\mu\text{m}^2$ )	Power ( $\mu\text{W}$ )
CSLA_NOBEC	0.85	258.46	31.22
CSLA_BEC	1.03	279.05	27.34
RCLA	0.77	357.83	21.76
Hybrid RCLA/RCA	0.72	319.20	20.64
BCLA	0.94	284.64	23.01
Hybrid BCLA/RCA	0.82	242.45	20.91



**Figure 6.** Normalized (a) PDP plots, and (b) EDP plots of approximate 32-bit adders (LOAs)—the types of adders used for the 16-bit accurate adder part of LOAs are mentioned on the X-axis.

## 5. Conclusions

This communication discussed the implementation of high-speed, low-power CLAs and CSLAs and compared their performances by considering accurate and approximate 32-bit additions. The comparisons show that the hybrid RCLA/RCA architecture is most beneficial amongst the different CLA and CSLA architectures in terms of the delay and power dissipation. It was reported in Reference [9] that the CLA requires a substantially lower number of input patterns than the CSLA for the testing of stuck-at faults, which is another advantage of the CLA architecture. As a further work, the utility of the hybrid RCLA/RCA architecture can be studied at length by considering many digital signal processing operations which often involve additions and multiplications. Also, as an extension of this short communication, the family of high-speed parallel-prefix adders [14] can be considered for physical implementation, and an extensive comparison could be drawn toward the efficient realization of computer arithmetic.

**Author Contributions:** Conceptualization, P.B.; methodology, P.B. and N.M.; validation, P.B.; formal analysis, P.B. and N.M.; investigation, P.B.; resources, N.M.; data curation, P.B.; writing—original draft preparation, P.B.

**Funding:** This research received no external funding.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## References

1. Weste, N.H.E.; Eshraghian, K. *Principles of CMOS VLSI Design: A Systems Perspective*, 2nd ed.; Addison-Wesley Publishing Company: Reading, MA, USA, 1994; ISBN 978-0201533767.
2. Ercegovac, M.D.; Lang, T. *Digital Arithmetic*; Morgan Kaufmann Publishers: Burlington, MA, USA, 2003; ISBN 978-1558607989.
3. Omondi, A.R. *Computer Arithmetic Systems: Algorithms, Architecture and Implementations*; Prentice-Hall International (UK) Limited: Hertfordshire, UK, 1994; ISBN 978-0133343014.

4. Balasubramanian, P.; Mastorakis, N.E. ASIC-based implementation of synchronous section-carry based carry lookahead adders. In *Recent Advances in Circuits, Systems, Signal Processing and Communications*; Mladenov, V., Ed.; WSEAS Press: Athens, Greece, 2016; pp. 58–64, ISBN 978-1618043665.
5. Balasubramanian, P. Design of synchronous section-carry based carry lookahead adders with improved figure of merit. *WSEAS Trans. Circuits Syst.* **2016**, *15*, 155–164.
6. Bedrij, O.J. Carry-select adder. *IRE Trans. Electron. Comput* **1962**, *EC-11*, 340–346. [[CrossRef](#)]
7. Ramkumar, B.; Kittur, H.M. Low-power and area-efficient carry select adder. *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* **2012**, *20*, 371–375. [[CrossRef](#)]
8. Mohanty, B.K.; Patel, S.K. Area-delay-power efficient carry select adder. *IEEE Trans. Circuits Syst. II Exp. Briefs.* **2014**, *61*, 418–422. [[CrossRef](#)]
9. Saini, V.K.; Akhter, S.; Chauhan, T. Implementation, test pattern generation, and comparative analysis of different adder circuits. *VLSI Des.* **2016**, *2016*, 1260879. [[CrossRef](#)]
10. 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 Reg. Papers* **2010**, *57*, 850–862. [[CrossRef](#)]
11. Synopsys SAED\_EDK32/28\_CORE Databook. Revision 1.0.0. 2012. Available online: <https://www.synopsys.com/community/university-program/teaching-resources.html> (accessed on 10 December 2017).
12. Rabaey, J.M.; Chandrakasan, A.; Nikolic, B. *Digital Integrated Circuits: A Design Perspective*, 2nd ed.; Pearson Education: London, UK, 2003; ISBN 978-0130909961.
13. Najafi, A.; Weißbrich, M.; Vayá, G.P.; Garcia-Ortiz, A. A fair comparison of adders in stochastic regime. In Proceedings of the 27th International Symposium on Power and Timing Modeling, Optimization and Simulation, Thessaloniki, Greece, 25–27 September 2017.
14. Harris, D. A taxonomy of parallel prefix networks. In Proceedings of the Thirty-Seventh Asilomar Conference on Signals, Systems and Computers, Pacific Grove, CA, USA, 9–12 November 2003.



© 2018 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).